

Machine Learning Platform for AI

Best Practices

Best Practices

Heart disease prediction

Overview

Heart disease is the biggest killer of humans. Heart disease causes 33% of deaths in the world. In China, hundreds and thousands of people die of heart disease every year. Data mining has become extremely important for heart disease prediction and treatment. It uses the relevant health exam indicators and analyzes their influences on heart disease. This document introduces how to use Alibaba Cloud Machine Learning Platform for AI to create a heart disease prediction model based on the data collected from heart disease patients.

Datasets

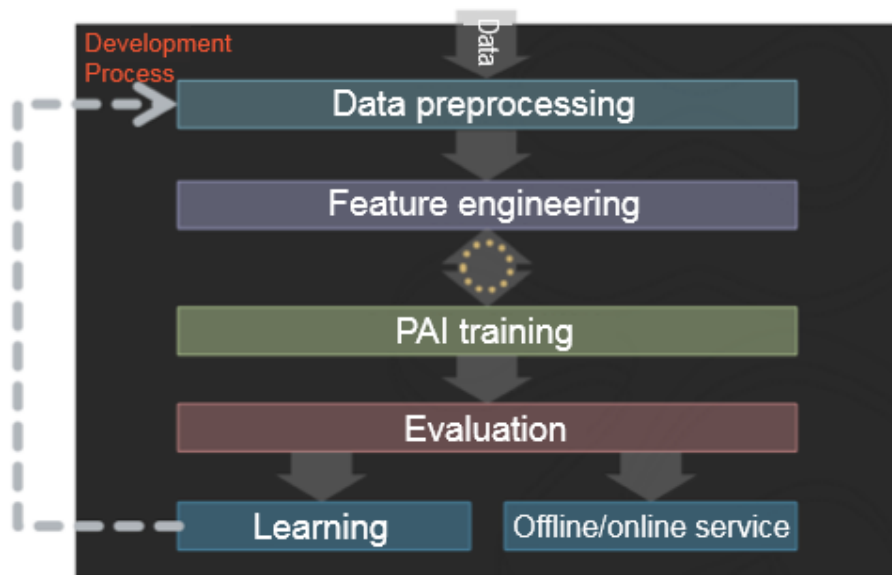
Data source UCI Heart Disease Dataset. This dataset is created based on 303 cases of heart disease in the United States. The attributes are as follows:

Name	Definition	Data Type	Description
age	Age	string	Age of a patient. The age attribute only uses numbers.
sex	Gender	string	Gender of a patient: female or male.
cp	Chest pain type	string	Chest pain types, including typical, atypical, non-anginal, and asymptomatic.
trestbps	Blood pressure	string	Blood pressure of a patient.
chol	cholesterol	string	Cholesterol of a patient.
fbs	Fasting blood sugar	string	True means that a patient's fasting blood sugar is

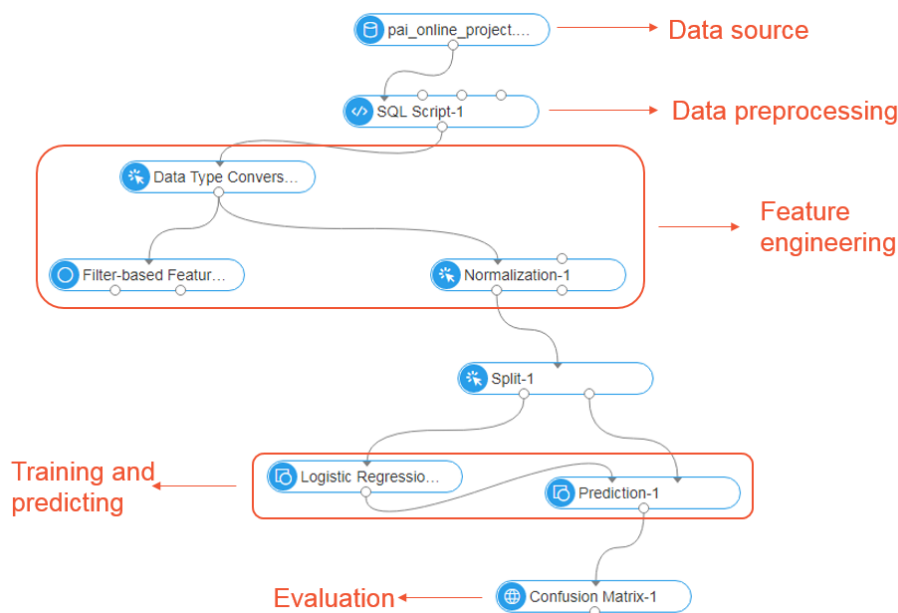
			greater than 120 mg/dl. False means that a patient's fasting blood sugar is equal to or less than 120 mg/dl.
restecg	Resting electrocardiographic result	string	The resting electrocardiographic results include normal, having ST-T wave abnormality, and showing probable or definite left ventricular hypertrophy.
thalach	Maximum heart rate achieved	string	Maximum heart rate of a patient.
exang	Exercise induced angina	string	True means that a patient has exercise induced angina. False means that a patient does not have exercise induced angina.
oldpeak	ST depression induced by exercise relative to rest	string	ST depression of a patient.
slop	Slope of the peak exercise ST segment	string	Slopes of the peak exercise ST segment, including down, flat, and up.
ca	Number of major vessels colored by flouroscopy	string	Number of major vessels colored by flouroscopy
thal	Defect type	string	defect types, including norm, fix, and rev.
status	Heart disease status	string	Health means that a patient does not have heart disease. Sick means that a patient has heart disease.

Data exploring procedure

The following figure shows the procedure of data mining:



The following figure shows the workflow of the project:



Data pre-processing

Data pre-processing, also known as data cleaning, is the process of analyzing and making changes to the source data, including irrelevant data removal, incomplete data fixing, and data type conversion. With 14 indicators and one goal field, this project focuses on predicting the presence or absence of heart disease in patients based on their health exam indicators. The project uses one of the generalized linear models: logistic regression. Additionally, the data type of all input indicators is double.

All input data:

Data exploration - pai_online_project.heart_disease_prediction - (Show top one hundred rows.)

Index	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop	ca	thal	status	style
1	63.0	male	an...	145.0	233.0	true	hyp	150.0	false	2.3	down	0.0	fix	buff	H
2	67.0	male	as...	160.0	286.0	false	hyp	108.0	true	1.5	flat	3.0	norm	sick	S2
3	67.0	male	as...	120.0	229.0	false	hyp	129.0	true	2.6	flat	2.0	rev	sick	S1
4	37.0	male	not...	130.0	250.0	false	norm	187.0	false	3.5	down	0.0	norm	buff	H
5	41.0	fem	ab...	130.0	204.0	false	hyp	172.0	false	1.4	up	0.0	norm	buff	H
6	56.0	male	ab...	120.0	236.0	false	norm	178.0	false	0.8	up	0.0	norm	buff	H
7	62.0	fem	as...	140.0	268.0	false	hyp	160.0	false	3.6	down	2.0	norm	sick	S3
8	57.0	fem	as...	120.0	354.0	false	norm	163.0	true	0.6	up	0.0	norm	buff	H
9	63.0	male	as...	130.0	254.0	false	hyp	147.0	true	1.4	flat	1.0	rev	sick	S2
10	53.0	male	as...	140.0	203.0	true	hyp	155.0	true	3.1	down	0.0	rev	sick	S1
11	57.0	male	as...	140.0	192.0	false	norm	148.0	false	0.4	flat	0.0	fix	buff	H

During data pre-processing, we must convert data of string and text types to numeric type based on the definition of the data.

Boolean data

For example, you can set the sex attribute to 0 to indicate female and set the attribute to 1 to indicate male.

Multivalued data

For example, you can use 0 through 3 to numerically rate the chest pain in ascending order for the cp attribute.

The data pre-processing is based on SQL scripts. Learn more, see the SQL script-1 component as follows:

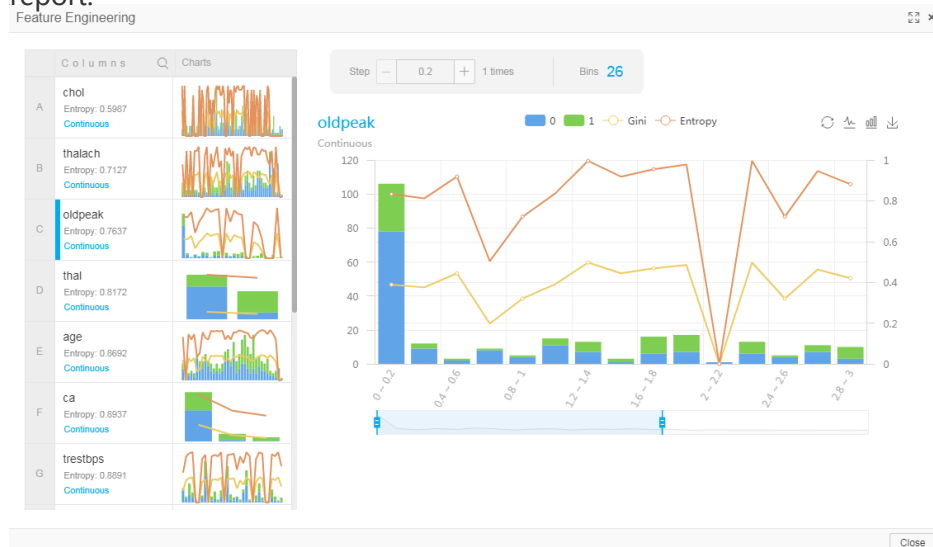
```
select age,
(case sex when 'male' then 1 else 0 end) as sex,
(case cp when 'angina' then 0 when 'notang' then 1 else 2 end) as cp,
trestbps,
chol,
(case fbs when 'true' then 1 else 0 end) as fbs,
(case restecg when 'norm' then 0 when 'abn' then 1 else 2 end) as restecg,
thalach,
(case exang when 'true' then 1 else 0 end) as exang,
oldpeak,
(case slop when 'up' then 0 when 'flat' then 1 else 2 end) as slop,
ca,
(case thal when 'norm' then 0 when 'fix' then 1 else 2 end) as thal,
(case status when 'sick' then 1 else 0 end) as ifHealth
from ${t1};
```

Feature engineering

Feature engineering includes feature derivation and scale change. This project uses the feature selection and data normalization components for feature engineering.

Filter-based feature selection

This component measures the influence of each indicator on the prediction results by using the entropy and Gini coefficient. You can view the final prediction results in the assessment report.



Data normalization

This project requires you to train your model by using dichotomous logistic regression. Therefore, you must avoid using different fundamental units for the indicators. Data normalization uses the following formula to ensure that all indicators use a value between 0 and 1: $\text{result} = (\text{val} - \text{min}) / (\text{max} - \text{min})$.

The following figure shows the results of data normalization:

Data exploration - pai_temp_121028_1317476_1 - (Show top one hundred rows.)

Index	sex	cp	tbs	restecg	exang	slo	thal	tfhealth	age	trestbps	chol	thalach	oldpeak	ca
1	1	0	1	1	0	1	0.5	0	0.70...	0.4811320...	0.244...	0.603053...	0.370967...	0
2	1	1	0	1	1	0.5	0	1	0.79...	0.6226415...	0.365...	0.282442...	0.241935...	1
3	1	1	0	1	1	0.5	1	1	0.79...	0.2452830...	0.235...	0.442748...	0.419354...	0.6666666666666666
4	1	0.5	0	0	0	1	0	0	0.16...	0.3396226...	0.283...	0.885496...	0.564516...	0
5	0	1	0	1	0	0	0	0	0.25	0.3396226...	0.178...	0.770992...	0.225806...	0
6	1	1	0	0	0	0	0	0	0.5625	0.2452830...	0.251...	0.816793...	0.129032...	0
7	0	1	0	1	0	1	0	1	0.6875	0.4339622...	0.324...	0.679389...	0.580645...	0.6666666666666666
8	0	1	0	0	1	0	0	0	0.58...	0.2452830...	0.520...	0.702290...	0.096774...	0
9	1	1	0	1	0	0.5	1	1	0.70...	0.3396226...	0.292...	0.580152...	0.225806...	0.3333333333333333

Model training and prediction

Supervised learning requires you to train your model to obtain the prediction results and compare the prediction results with the existing data. In this project, supervised learning is used to train the model to predict the presence or absence of heart disease in a group of patients.

Data split

Use the split component to split the data into the training dataset and predicting dataset at the ratio of 7:3. The training dataset is imported to the dichotomous logistic regression component for model training. The predicting dataset is imported to the prediction

component.

Dichotomous logistic regression

Logistic regression is a linear model. In this project, dichotomous logistic regression (determining the presence or absence of heart disease) is achieved by comparing the prediction results with a threshold. You can learn more about logistic regression from the Internet or relevant documentation. You can view the model that has already been trained by logistic regression on the Model page.

Logistic Regression Output

feature ▲	weight	
	1 ▲	0 ▲
sex	1.473569994686197	-
cp	2.730064736238172	-
fbs	-0.6007338270729394	-
restecg	0.8990240712157691	-
exang	0.9026382341453308	-
slop	1.041821068646534	-
thal	1.562393603912368	-
age	-0.4278050593226199	-

Prediction

The prediction component has two inputs: the model and the predicting dataset. The prediction results show the calculated data, the predicting data, and the probability of inconsistencies between the calculated data and predicting data.

Assessment

You can use the confusion matrix to assess the attributes of the model, such as the accuracy.

Confusion Matrix							
<div>Confusion Matrix Proportion Matrix Stats</div>							
Models ▲	true count ▲	False count ▲	Summary ▲	Accuracy ▲	Precision ▲	Recall Rate ▲	F1 ▲
0	47	11	58	82.418%	81.034%	90.385%	85.455%
1	28	5	33	82.418%	84.848%	71.795%	77.778%

Based on the accuracy of the prediction result, you can determine whether your model is well trained or not.

Conclusions

According to the workflow of data exploring, the following conclusions can be made:

Feature weight

- You can obtain the weight of each indicator in the prediction by using filter-based feature selection.

featname ▲	weight ▲
thalach	0.16569171224597157
oldpeak	0.14640697618779352
thal	0.13769166559906015
ca	0.11467097546217575
chol	0.10267709576600859
age	0.07876430484527841
trestbps	0.0772599125640569
slop	0.07702762609078306
restecg	0.015246832497405105
cp	0.0037507283721422424
exang	0
fbs	0
sex	0

- The maximum heart rate achieved (thalach) indicator has the greatest impact on heart disease prediction.
- The gender indicator does not have any impact on heart disease prediction.

Prediction results

Based on the 14 indicators, the model can predict heart disease with an accuracy of over 80%. This model can be used in heart disease prediction and treatment.

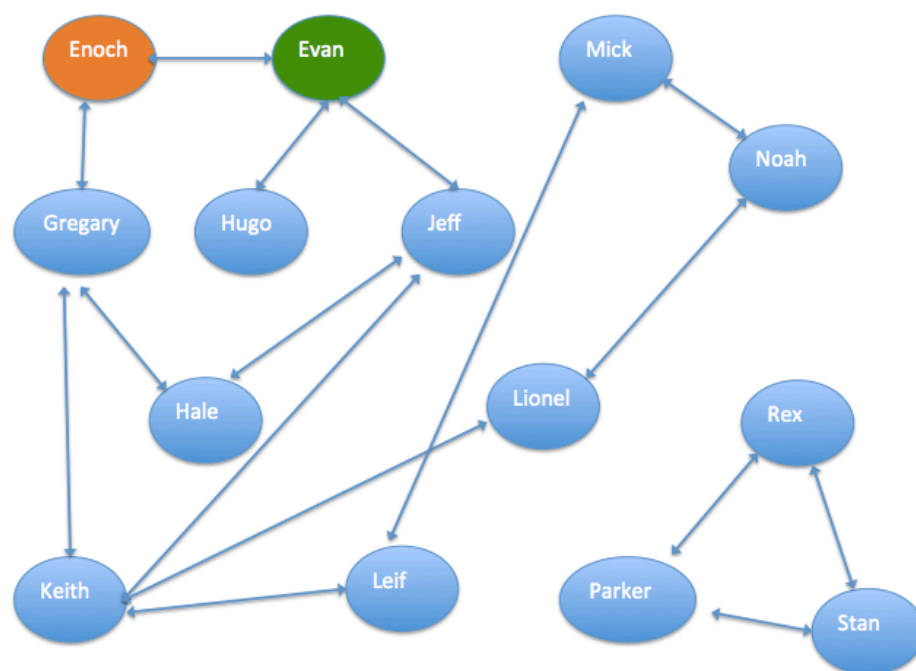
Financial risk management

Overview

This project is created by using Alibaba Cloud Network Chart. Network Chart is used to illustrate the interconnections among a set of entities, for example, the relationships among a group of people. Unlike hierarchical data, the relationships in Network Chart are represented by nodes and edges (links). The nodes are connected to each other through edges. Alibaba Cloud Machine Learning Platform For AI provides several Network Chart components, including K-Core, largest connected subgraph, and label propagation classification.

Scenario

The following figure shows the relationships among a group of people. The arrows in the figure represent the relationships between these people (for example, coworkers or relatives). Enoch is a trusted customer and Evan is a fraudulent customer. Based on this information and the relationship graph, Network Chart allows you to calculate the credit scores of the remaining people for financial risk management. By referencing the credit scores, you can make predictions about which of them may be fraudulent customers.



Datasets

Data source: the dataset in this project is provided by Alibaba Cloud Machine Learning Platform For AI. The dataset includes the following attributes:

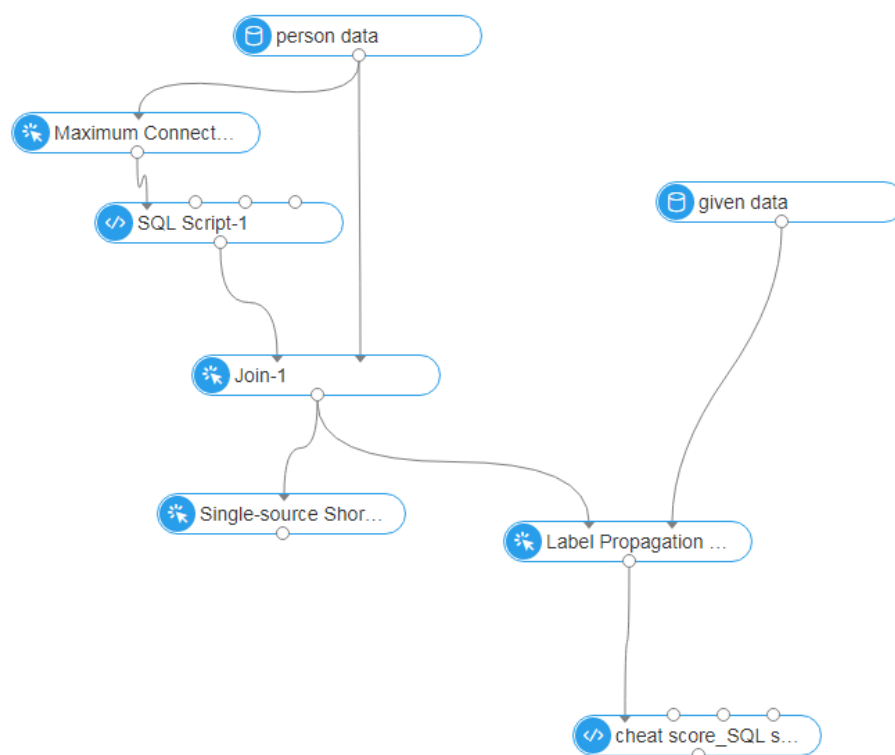
Name	Definition	Data Type	Description
start_point	Start node of an edge	string	Name of a person.
end_point	End node of an edge	string	Name of a person.
count	Relational closeness	double	The larger the value is, the closer relationship the two persons have.

The following figure shows the data entries:

start_point ▲	end_point ▲	count ▲
Enoch	Evan	10
Enoch	Gregary	2
Gregary	Hale	6
Evan	Hugo	2
Evan	Jeff	4
Gregary	Keith	7
Jeff	Keith	5
Hale	Jeff	11
Keith	Leif	3
Keith	Lionel	1
Leif	Mick	4

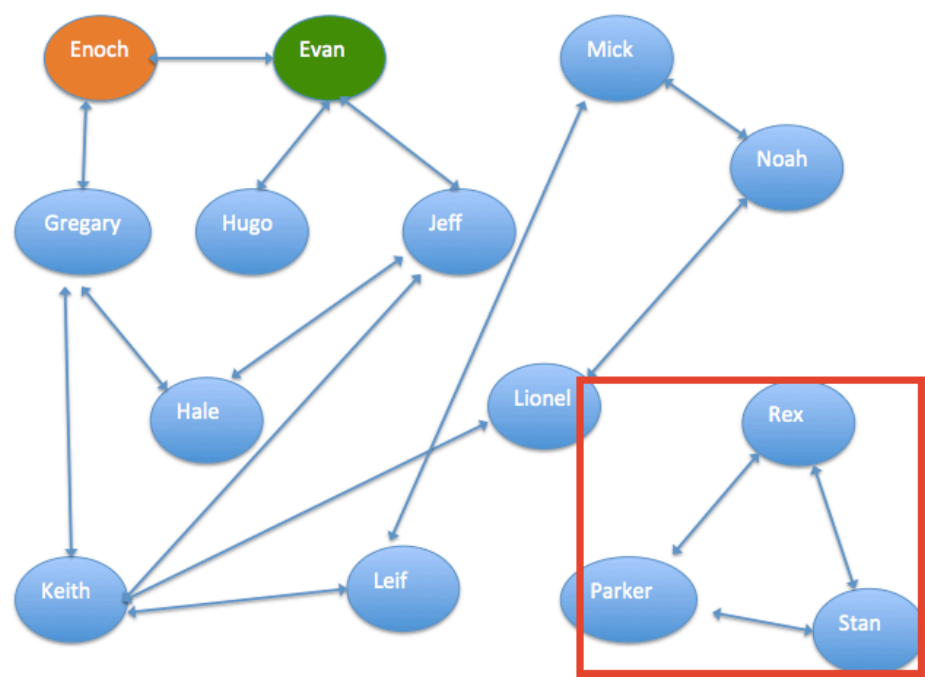
Data exploring procedure

The following figure shows the workflow of this project:



Largest connected subgraph

The largest connected subgraph allows you to find the cluster that contains the most interconnected entities. In this project, the largest connected subgraph divides the people into two groups and assigns each team a group ID (group_id). The group containing Parker, Rex, and Stan should be removed from the subgraph because the relationship between these people do not affect the prediction results. You can use the SQL script component and JOIN component to remove this group from the subgraph.



Single-source shortest path

The single-source shortest path allows you to measure the distance (number of nodes) that a start node must pass through to reach an end node.

The following figure shows the distances between Enoch and the others:

start_node ▲	dest_node ▲	distance ▲	distance_cnt ▲
Enoch	Hale	2	1
Enoch	Leif	3	1
Enoch	Hugo	2	1
Enoch	Keith	2	1
Enoch	Jeff	2	1
Enoch	Evan	1	1
Enoch	Lionel	3	1
Enoch	Mick	4	1
Enoch	Gregary	1	1
Enoch	Noah	4	1
Enoch	Enoch	0	0

Label propagation classification

Label propagation classification is a semi-supervised classification algorithm. It uses the existing label information of the nodes to predict the label information of the unlabeled nodes. Based on the correlations between the nodes, label propagation classification propagates each label to other nodes.

To use the label propagation classification component, make sure that you have a connected graph containing all entities and the data for labeling. In this project, the data for labeling is imported from the **Read Data Source** component. The weight column shows the probability of a person being a fraudulent customer.

point ▲	point_type ▲	weight ▲
Enoch	信用用户	1
Evan	欺诈用户	0.8

By SQL filtering, the final results show the probabilities of committing fraud for all people. The larger the value is, the larger probability a person may be fraudulent customer.

node ▲	tag ▲	weight ▼
Hugo	欺诈用户	1
Evan	欺诈用户	0.8
Noah	欺诈用户	0.42059743476528927
Jeff	欺诈用户	0.34784053907648443
Mick	欺诈用户	0.3113287445872401
Lionel	欺诈用户	0.2938277295951075
Leif	欺诈用户	0.24091136964145973
Keith	欺诈用户	0.2264783897173419

Product recommendation

Overview

The parable of beer and diapers is a classic case of data mining utilization. The diapers and beer are irrelevant. However, when the diapers and beer are put next to each other on shelves, both of their sales increase. The problem is how to find the hidden correlation between two irrelevant products. To resolve this problem, you can use collaborative filtering, which is one of the algorithms commonly used in data mining. This algorithm enables you to find the hidden correlation between different customers and products.

Collaborative filtering is a correlation rule-based algorithm. This project takes shopping behaviors as an example, including customers A and B and products X, Y, and Z. If both customers A and B have purchased products X and Y, collaborative filtering determines that customers A and B have similar interests in shopping. Collaborative filtering then recommends product Z to customer B because customer A has purchased product Z. In this case, collaborative filtering works based on customers' interests.

Scenario:

This project shows how to use the customer shopping behaviors recorded before July to find the correlations between products. We then use this information to recommend relevant products to customers. In addition, the project also makes an assessment of the recommendation results. For example, customer A purchased product X before July. Product X is strongly correlated with product Y. The system then recommends product Y to customer A after July and calculates the probability of customer A purchasing product Y.

Datasets

Data source: the two datasets are provided by the Tianchi challenges, including the shopping behaviors before July and the shopping behaviors after July.

The attributes are as follows:

Name	Definition	Data Type	Description
user_id	User ID	string	User ID of a customer.
item_id	Product ID	string	ID of a product.
active_type	Shopping behavior	string	A value of 0 indicates that the product page is viewed by the customer. A value of 1 indicates that the product is purchased. A value of 2 indicates that

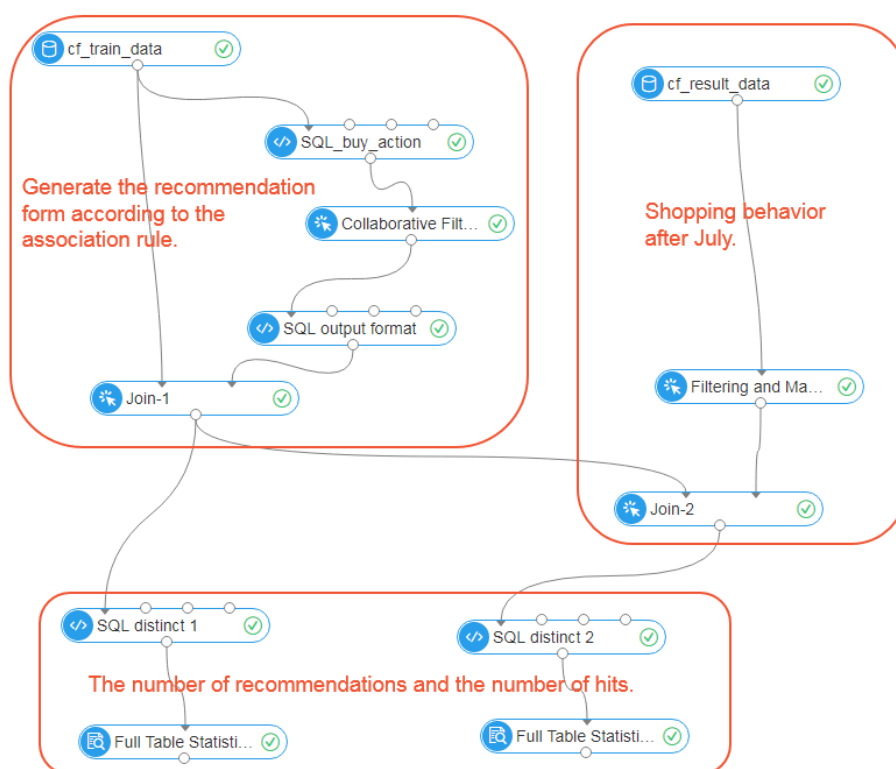
			the product is added to the customer' s favorites. A value of 3 indicates that the product is added to the customer' s shopping cart.
active_date	Purchased at	string	Time when the product is purchased.

The following figure shows the data entries:

10944750	8689	2	5月2日
10944750	25687	2	5月8日
10944750	7150	1	6月7日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日

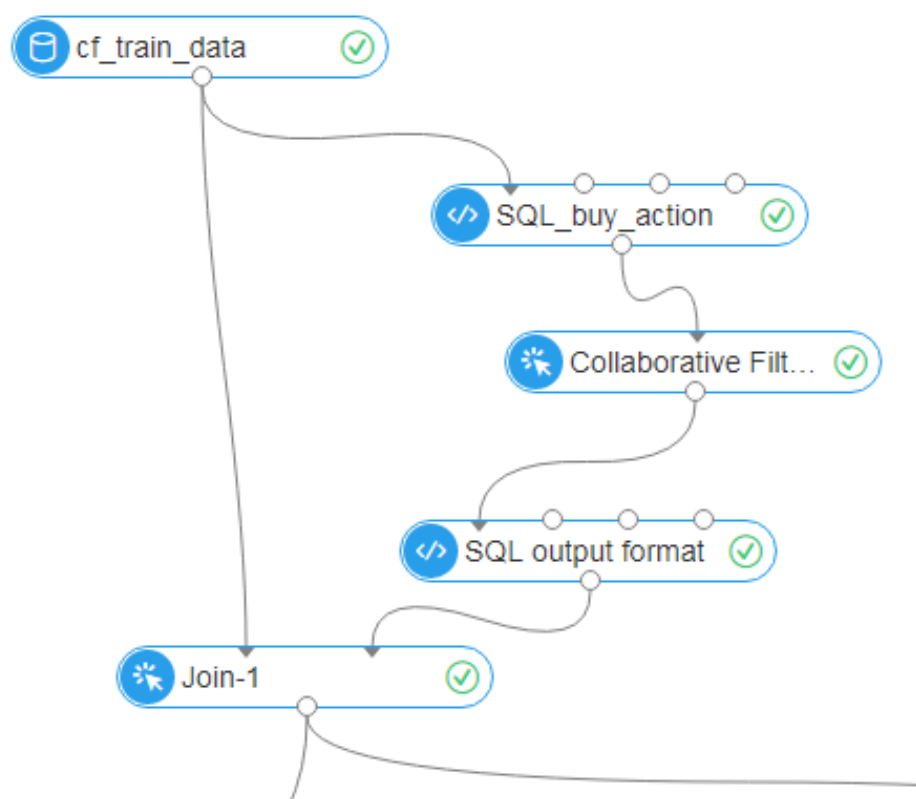
Data exploring procedure

The following figure shows the workflow of this project:



Collaborative filtering-based recommendation procedure

Load the dataset recorded before July, use SQL scripts to extract the shopping behaviors, and import the data to the collaborative filtering component. Set the **TopN** attribute to 1 for the collaborative filtering component. This allows the collaborative filtering component to find the most similar item for each input item and calculate its weight. Analyze the shopping behaviors and then make predictions about items that are most likely to be purchased by the same customer.



The following figure shows the relevant settings:

Column Settings

Parameter Settings

Similarity Type Optional.

wbcosine

Top N Optional (?)

1

Computation Method Optional (?)

Add

Min Item Quantity Optional (?)

2

Max Item Quantity Optional (?)

500

Smoothing Factor Optional (?)

0.5

Weighting Coefficient Optional (?)

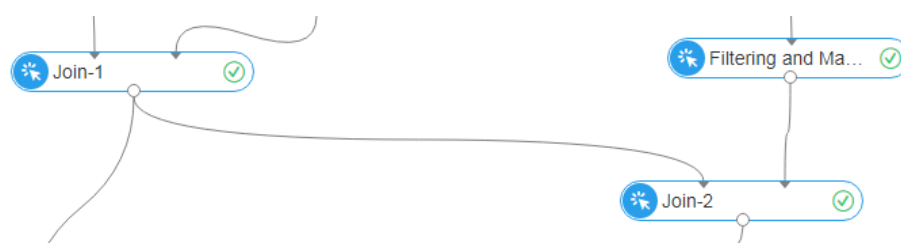
1

The following figure shows the collaborative filtering results. The **itemid** column shows the IDs of the target products. The **similarity** column shows two colon-separated items: ID of the product that is strongly correlated with the target product and the probability of this product being purchased.

itemid ▲	similarity ▲
1000	15584:0.2747133918
10014	18712:0.05229603127
10066	3228:0.2650900672
1008	24507:1
10082	18024:0.1781525919
1010	18024:0.2104947227
10133	14020:0.2070609237
1015	18024:0.2104947227
10151	26288:0.4366713611
10171	11080:0.2401992435

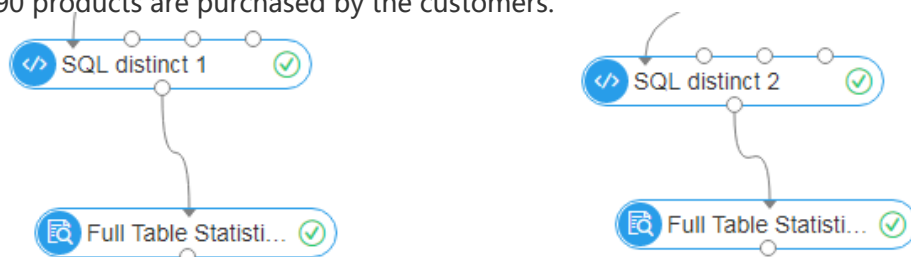
Product recommendations

The preceding steps show how to list all strongly correlated products. The following figure shows the workflow of using the product similarity list to make recommendations and predicting the recommendation results. For example, if customer A purchased product X and product X is strongly correlated with product Y, product Y then is recommended to customer A.



Recommendation results

This figure shows the statistics components. The full table scan component 1 shows the recommendation list created based on the shopping behaviors before July. By removing any duplicate rows, the final list contains 18,065 entries. The full table scan component 2 shows the number of products (in the recommendation list) that are purchased by the customers. In this project, 90 products are purchased by the customers.



Conclusions

By referencing the recommendation results, we can still make the following improvements to the project:

The project should include all factors that may influence the recommendation results. For example, the shopping behaviors must be time effective. In this project, the dataset includes shopping behaviors recorded in several months. Using outdated data may prevent you from getting the expected recommendation results. Additionally, the project only focuses on the hidden correlations between the products. The attributes of the recommended products are not taken into consideration. For example, whether the products are frequently rated products or not. If customer A bought a cell phone last month, he may not buy another cell phone the next month. In this case, cell phones are infrequently rated products.

To increase the accuracy of the prediction, this project should use a model trained by machine learning. The latent product associations should be only used as supplementary data.

Credit card bill statements-based-credit scorecard

Overview

Scorecard is not only a machine learning algorithm, but also a generic modeling framework used to

build a model for assessing credit risks. In scorecard modeling, the original data is processed by data binning and feature engineering, and then is used to build a linear model.

Scorecard modeling is typically used in credit assessment scenarios, such as for credit card applications and loan disbursements. It is also used in other industries for scoring, including customer service scoring and Alipay credit scoring. This project shows how to use the financial component on Alibaba Cloud Machine Learning Platform for AI to build a scorecard model.

Datasets

The following dataset contains client information, such as gender, education, marital status, and age, payment history, and credit card billing statements. The `payment_next_month` column (goal field) indicates the probability of a client paying off their credit card debt, as shown in the following figure. A value of 1 indicates that the client will likely pay off the debt and a value of 0 indicates that the client will not likely pay off the debt.

Source Table Columns

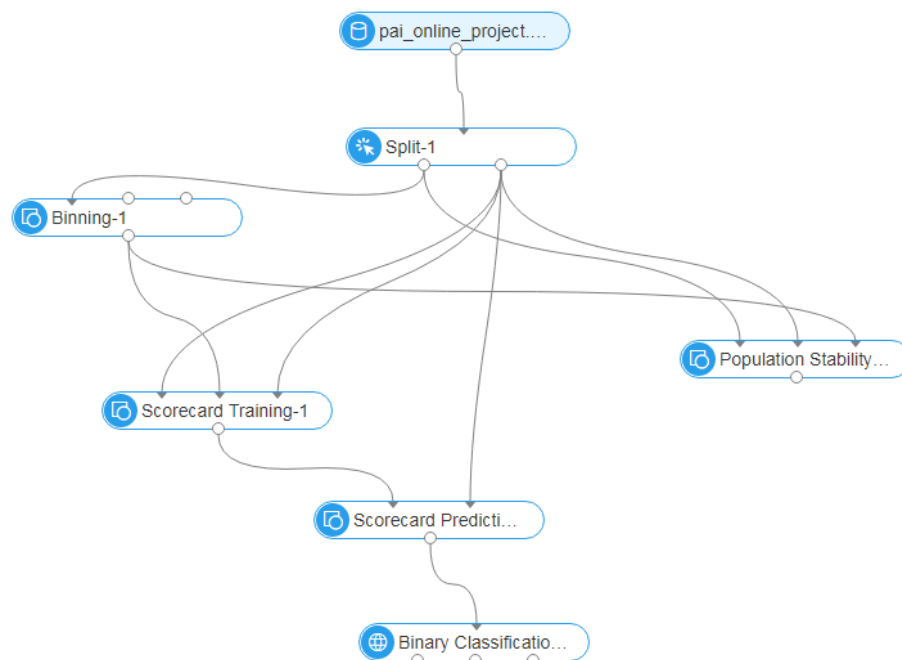


Columns	Type	Range from or
id	STRING	1,2,3,4,5
limit_bal	BIGINT	20000,50000,
sex	STRING	女,男
education	STRING	本科
marriage	STRING	已婚,未婚
age	BIGINT	24,26,34,37,5
pay_0	BIGINT	-1,0,2
pay_2	BIGINT	0,2
pay_3	BIGINT	-1,0
pay_4	BIGINT	-1,0
pay_5	BIGINT	-2,0
pay_6	BIGINT	-2,0,2
bill_amt1	DOUBLE	2682.0,3913.0
bill_amt2	DOUBLE	1725.0,3102.0
bill_amt3	DOUBLE	689.0,2682.0,
bill_amt4	DOUBLE	0.0,3272.0,14
bill_amt5	DOUBLE	0.0,3455.0,14
bill_amt6	DOUBLE	0.0,3261.0,15
pay_amt1	DOUBLE	0.0,1518.0,20
pay_amt2	DOUBLE	689.0,1000.0,
pay_amt3	DOUBLE	0.0,1000.0,12
pay_amt4	DOUBLE	0.0,1000.0,11
pay_amt5	DOUBLE	0.0,689.0,100

The dataset contains 30,000 entries. You can download the dataset from <https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset>.

Project workflow

The following figure shows the workflow of this project:



The procedure includes the following major steps:

Data split

Split the input data into two parts: one for model training and one for prediction result assessment.

Data binning

Data binning is similar to onehot encoding. It is a process of grouping the input data into data classes (bins). The data values in each bin are replaced by a value, which is the representative of the bin. As shown in the following figure, the binning component groups the age values into a number of age intervals:

	Index ▲	Label ▲	Constraint		WoE		Number			Rate		
			Operator	Value	WoE ▲	Chart	Total ▲	Positive ▲	Negative ▲	Total ▲	Positive ▲	Negative
	0	(-inf,25]	▼		0.249		3082	822	2260	12.84%	15.5%	12.09%
	1	(25,27]	▼		-0.12		2184	439	1745	9.1%	8.26%	9.33%
	2	(27,29]	▼		-0.137		2421	480	1941	10.09%	9.05%	10.38%
	3	(29,31]	▼		-0.196		2084	394	1690	8.68%	7.43%	9.04%
	4	(31,34]	▼		-0.2		2791	526	2265	11.63%	9.92%	12.11%
	5	(34,37]	▼		-0.016		2622	572	2050	10.93%	10.79%	10.96%
	6	(37,40]	▼		-0.025		2224	482	1742	9.27%	9.09%	9.32%
	7	(40,43]	▼		0.026		1823	411	1412	7.6%	7.75%	7.55%
	8	(43,49]	▼		0.083		2628	619	2009	10.95%	11.67%	10.74%
	9	(49,+inf]	▼		0.215		2141	557	1584	8.92%	10.51%	8.47%
	-2	ELSE	▼				-	-	-	-	-	-

As shown in the following figure, after data binning, each field falls into multiple intervals:

Data exploration - pai_temp_121044_1317575_1 - (Show top one hundred rows.)

Index ▲	feature ▲	json ▲
1	limit_bal	{ "bin": {"norm": [{"iv": 0.07123500000000001, "n": 2086, "p": 1155, "prate": 0.356371, "total": 3241, "value": "(-inf,30000]"}, {"woe": 0.669669, {"iv": 0.011173, "n": 2074, "p": ...
2	age	{ "bin": {"norm": [{"iv": 0.008099, "n": 2257, "p": 816, "prate": 0.265539, "total": 3073, "value": "(-inf,25]"}, {"woe": 0.243439, {"iv": 0.0004929999999999999, "n": 1744, "p": 4...}
3	pay_0	{ "bin": {"norm": [{"iv": 0.047537, "n": 5746, "p": 1052, "prate": 0.154751, "total": 6798, "value": "(-inf,-1]"}, {"woe": -0.436994, {"iv": 0.170212, "n": 10241, "p": 1515, "prate": ...}
4	pay_2	{ "bin": {"norm": [{"iv": 0.007126, "n": 2490, "p": 552, "prate": 0.18146, "total": 3042, "value": "(-inf,-2]"}, {"woe": -0.245673, {"iv": 0.031622, "n": 4077, "p": 758, "prate": 0.156...}
5	pay_3	{ "bin": {"norm": [{"iv": 0.007195, "n": 2680, "p": 599, "prate": 0.182678, "total": 3279, "value": "(-inf,-2]"}, {"woe": -0.237494, {"iv": 0.034982, "n": 4025, "p": 728, "prate": 0.15...}
6	pay_4	{ "bin": {"norm": [{"iv": 0.005376, "n": 2853, "p": 664, "prate": 0.188797, "total": 3517, "value": "(-inf,-2]"}, {"woe": -0.197027, {"iv": 0.029571, "n": 3822, "p": 711, "prate": 0.15...}
7	pay_5	{ "bin": {"norm": [{"iv": 0.004848, "n": 2950, "p": 696, "prate": 0.190894, "total": 3646, "value": "(-inf,-2]"}, {"woe": -0.183394, {"iv": 0.027291, "n": 3748, "p": 707, "prate": 0.15...}
8	pay_6	{ "bin": {"norm": [{"iv": 0.003858, "n": 3166, "p": 767, "prate": 0.195017, "total": 3933, "value": "(-inf,-2]"}, {"woe": -0.156921, {"iv": 0.020181, "n": 3839, "p": 774, "prate": 0.16...}
9	bill_amt1	{ "bin": {"norm": [{"iv": 0.001837, "n": 1813, "p": 587, "prate": 0.244583, "total": 2400, "value": "(-inf,267]"}, {"woe": 0.133103, {"iv": 1e-06, "n": 1871, "p": 529, "prate": 0.2204...}
10	bill_amt2	{ "bin": {"norm": [{"iv": 0.000424, "n": 1945, "p": 587, "prate": 0.231833, "total": 2532, "value": "(-inf,0]"}, {"woe": 0.062824, {"iv": 5.1e-05, "n": 1777, "p": 492, "prate": 0.2168...}

Population stability index

Population stability index (PSI) is an important metric to identify a shift in the population for credit scorecards, for example, the changes in the population within two months. A PSI value smaller than 0.1 indicates insignificant changes. A PSI value between 0.1 and 0.25 indicates minor changes. A PSI value larger than 0.25 indicates major changes in the population.

By comparing the stability of the population before data split, after data split, and after data binning, the model calculates the final PSI values for all features as follows:

Feature ▲	Bin ▲	Test % ▲	Base % ▲	Test - Base ▲	ln(Test/Base) ▲	PSI ▲
<input type="checkbox"/> limit_bal	-	-	-	-	-	0.0019
<input checked="" type="checkbox"/> age	-	-	-	-	-	0.0005
<input type="checkbox"/> pay_0	-	-	-	-	-	0.0002
<input type="checkbox"/> pay_2	-	-	-	-	-	0.0006
<input type="checkbox"/> pay_3	-	-	-	-	-	0.0005
<input type="checkbox"/> pay_4	-	-	-	-	-	0.0016
<input type="checkbox"/> pay_5	-	-	-	-	-	0.0015
<input type="checkbox"/> pay_6	-	-	-	-	-	0.0019
<input type="checkbox"/> bill_amt1	-	-	-	-	-	0.001
<input type="checkbox"/> bill_amt2	-	-	-	-	-	0.0025
<input type="checkbox"/> bill_amt3	-	-	-	-	-	0.0022
<input type="checkbox"/> bill_amt4	-	-	-	-	-	0.0014
<input type="checkbox"/> bill_amt5	-	-	-	-	-	0.0011
<input type="checkbox"/> bill_amt6	-	-	-	-	-	0.0009
<input type="checkbox"/> pay_amt1	-	-	-	-	-	0.0032
<input type="checkbox"/> pay_amt2	-	-	-	-	-	0.0009

Scorecard training

The following figure shows the scorecard training results:

Variable	Selected	Bin Id	Variable/Bin	Const.	Weight		WOE	Importance	Total	Train			
					Unscaled	Scaled				Positive	Negative	% Pos	% Neg
Intercept	-	-	-	-	-1.254	531	-	-	-	-	-	-	-
pay_0	✓	-	-	-	0.789	-	-	4.445e-2	-	-	-	-	-
	-	0	(-inf,-1]	-	-0.34	-20	-0.415	-	1648	266	1382	19.65	29.75
	-	1	(-1,0]	-	-0.51	-29	-0.706	-	2943	370	2573	27.33	55.38
	-	2	(0,1]	-	0.474	27	0.562	-	757	256	501	18.91	10.78
	-	3	(1,2]	-	1.618	93	2.12	-	562	398	164	29.39	3.53
	-	4	(2,+inf)	-	1.747	101	2.134	-	90	64	26	4.73	0.56
	-	-2	ELSE	-	0	0	-	-	0	0	0	0	0
	-	-1	NULL	-	0	0	-	-	0	0	0	0	0
limit_bal	✓	-	-	-	0.453	-	-	2.414e-3	-	-	-	-	-
	-	0	(-inf,30000]	-	0.299	17	0.743	-	803	305	498	22.53	10.72
	-	1	(30000,50000]	-	0.124	7	0.269	-	710	196	514	14.48	11.06
	-	2	(50000,70000]	-	0.168	10	0.208	-	337	89	248	6.57	5.34
	-	3	(70000,100000]	-	0.058	3	0.161	-	639	163	476	12.04	10.25
	-	4	(100000,140000]	-	0.02	1	0.033	-	579	134	445	9.9	9.58
	-	5	(140000,180000]	-	-0.126	-7	-0.398	-	684	112	572	8.27	12.31
	-	6	(180000,210000]	-	-0.139	-8	-0.222	-	486	92	394	6.79	8.48

The purpose of using the scorecard is to use normalized scores to indicate the weights of the features in the model.

- Unscaled: represents the original weight.
-
- Scaled: an index that indicates the amount of points that a feature gains or loses. For example, if the pay_0 feature falls into the (-1,0] bin, the feature gains 29 points. If the pay_0 feature falls into the (0,1] bin, the feature loses 27 points.
-
- Importance: represents the influence of each indicator on the prediction results. The larger the value is, the greater influence the indicator has.

Modeling results

In this project, the modeling results refer to the credit scores calculated for all clients, as shown in the following figure:

Data exploration - pai_temp_121044_1317577_1 - (Show top one hundred rows.)				
Index	payment_next_month	prediction_score	prediction_prob	prediction_detail
1	1	702	0.8426741578927107	("0":0.1573258421,"1":0.8426741579)
2	0	513	0.17196627060745318	("0":0.8280337294,"1":0.1719662706)
3	0	543	0.2534425185567956	("0":0.7465574814,"1":0.2534425186)
4	0	452	0.06944174926097901	("0":0.9305582507,"1":0.0694417493)
5	0	566	0.33592039510976124	("0":0.6640796049,"1":0.3359203951)
6	0	472	0.09238878984022982	("0":0.9076112102,"1":0.0923887899)
7	1	610	0.5314449414477093	("0":0.4685550586,"1":0.5314449414)
8	0	496	0.11714112722057633	("0":0.8828588728,"1":0.1171411272)
9	0	492	0.1258877124009584	("0":0.8741122876,"1":0.1258877124)
10	0	489	0.12060969220628287	("0":0.8793903078,"1":0.1206096922)
11	1	633	0.6240071289996736	("0":0.3759928710,"1":0.6240071290)
12	0	590	0.43668648320511594	("0":0.5633135168,"1":0.4366864832)
13	0	524	0.20197025563113366	("0":0.7980297444,"1":0.2019702556)

Conclusions

You can use the credit card billing statements of your clients to train a scorecard model to calculate credit scores for all the clients. The credit scores can be used in loans or other credit dependent

financial transactions for assessment.

Implement image classification by TensorFlow

Overview

The development of the Internet has generated large volumes of images and voice data. How to effectively make use of this unstructured data has always been a challenge for data mining professionals. The processing of unstructured data usually involves the use of deep learning algorithms. These algorithms can be daunting to use at first sight. In addition, processing this data usually requires powerful GPUs and a large amount of computing resources. This document introduces a method of image recognition using deep learning frameworks. This method can be applied to scenarios such as illicit image filtering, facial recognition, and object detection.

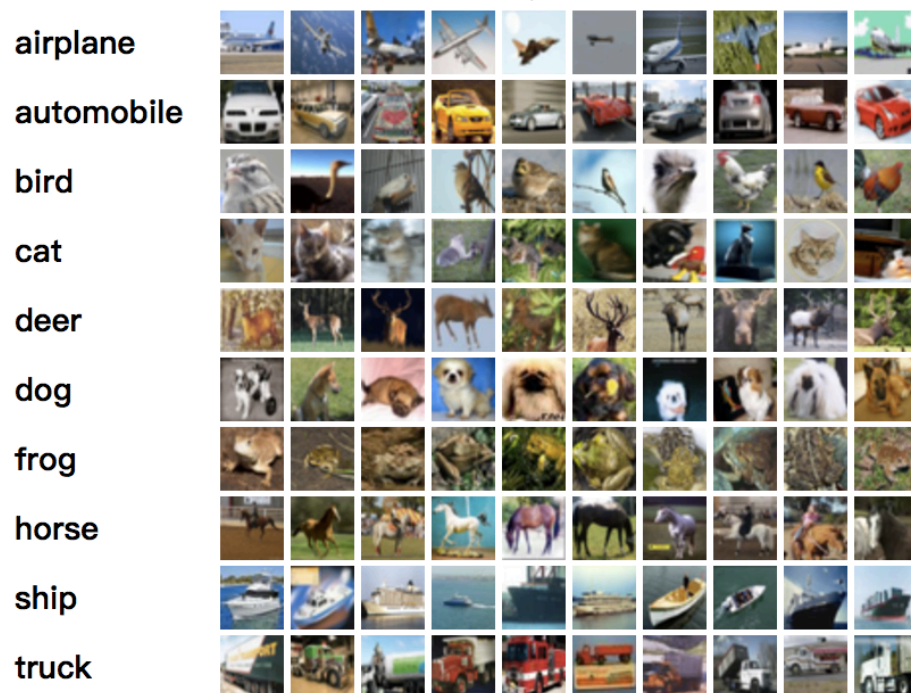
This guide creates an image recognition model using the deep learning framework TensorFlow in Alibaba Cloud Machine Learning Platform for AI. The entire procedure takes about 30 minutes to complete. After the procedure, the system is able to recognize the bird in the following image.



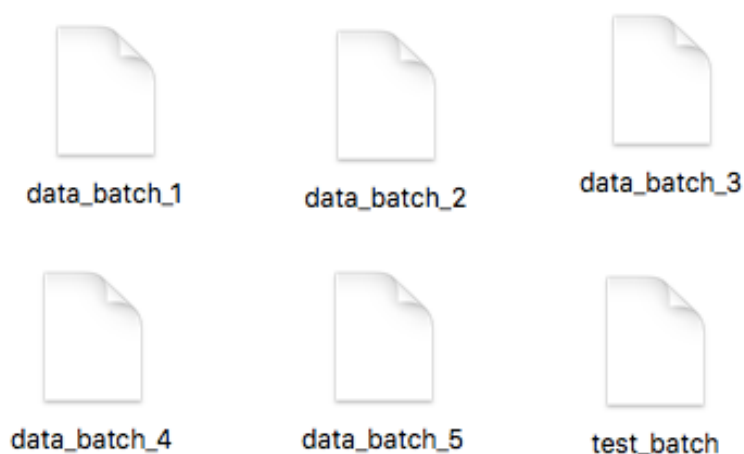
Dataset

To download the dataset and source code, click [Tensorflow_cifar10](#) case.

The CIFAR-10 dataset is used in this guide. This dataset contains 60,000 32x32 color images in 10 different categories, such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is as follows.



This source data is divided into two parts: 50,000 images are used for training and 10,000 for testing. The 50,000 training images are further divided into five data_batch files, and the 10,000 testing images form a test_batch file. The source data contains the following.



Training procedure

To create an experiment in the machine learning platform, you need to enable GPU usage and activate Object Storage Service (OSS) to store your data.

For more information about the machine learning platform, see [machine learning platform console](#).




For more information about OSS, see [OSS console](#).

1. Data preparation

Download the dataset and source code, then decompress them.

Log on to OSS, and create an OSS bucket (For more information, see [OSS Document](#)).

Create new directory in OSS bucket. An **aohai_test** directory is created in this article, and four folders are created under this directory as follows.

Folder Name	
	aohai_test/ Go back up a level
	check_point/
	cifar-10-batches-py/
	predict_code/
	train_code/

The role of each folder is as follows:

check_point: Stores the models that are generated in the experiment.

cifar-10-batches-py: Stores the training data, **file cifar-10-batcher-py**. The prediction data, **file bird_mount_bluebird.jpg**.

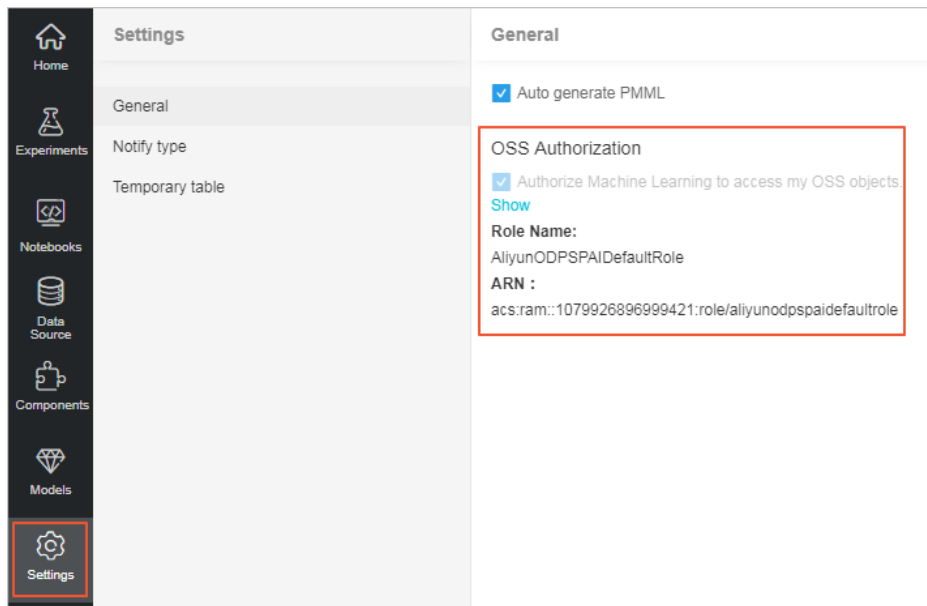
predict_code: Stores the code file **cifar_predict_pai.py**.

train_code: Stores the code **file cifar_pai.py**.

Upload the dataset and source code to the corresponding directory of the OSS bucket.

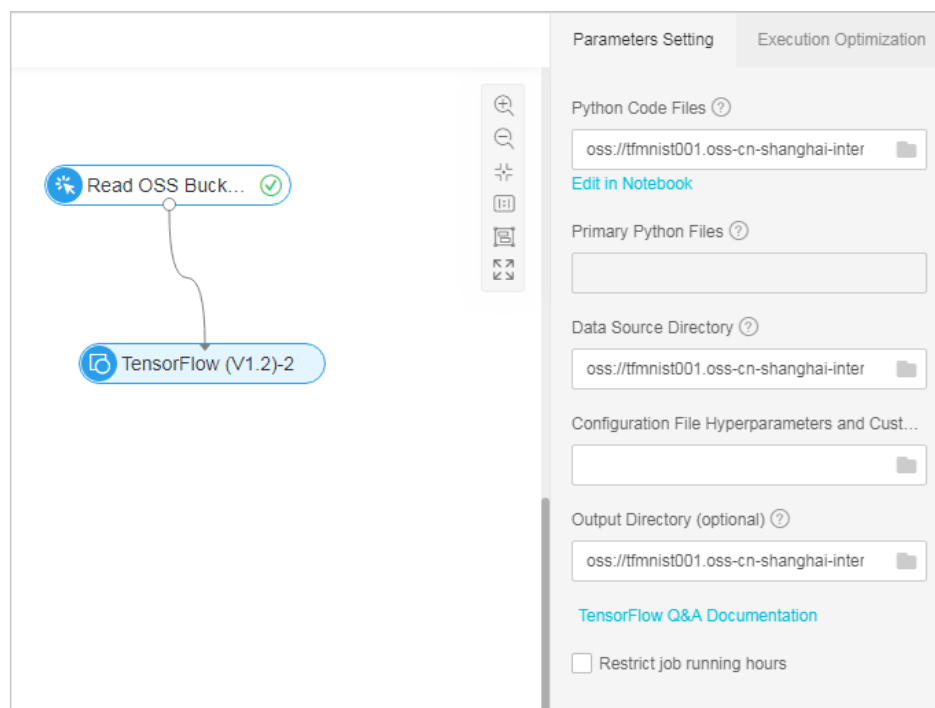
2. OSS permissions Configuration

Log on to the machine learning platform, and click **Settings** to configure OSS permissions, as shown in the following figure. For more information, see the “Read OSS buckets” chapter of Deep learning



3. Model training

Drag a **Read OSS Bucket** component and a **TensorFlow** component to the canvas, and configure the TensorFlow component as follows.



- Python Code File: Select the OSS directory of cifar_pai.py.
- Data Source Directory: Select the OSS directory of cifar-10-batches-py.

- Output Directory: Select the OSS directory of check_point.

Click **Run** to start the training procedure.

You can change the number of GPUs by changing the configuration as follows. You can also adjust the number of GPUs in the code.



4. Training code explanation

Note the following code in **cifar_pai.py**:

- The following code creates the training model using the convolutional neural network (CNN).

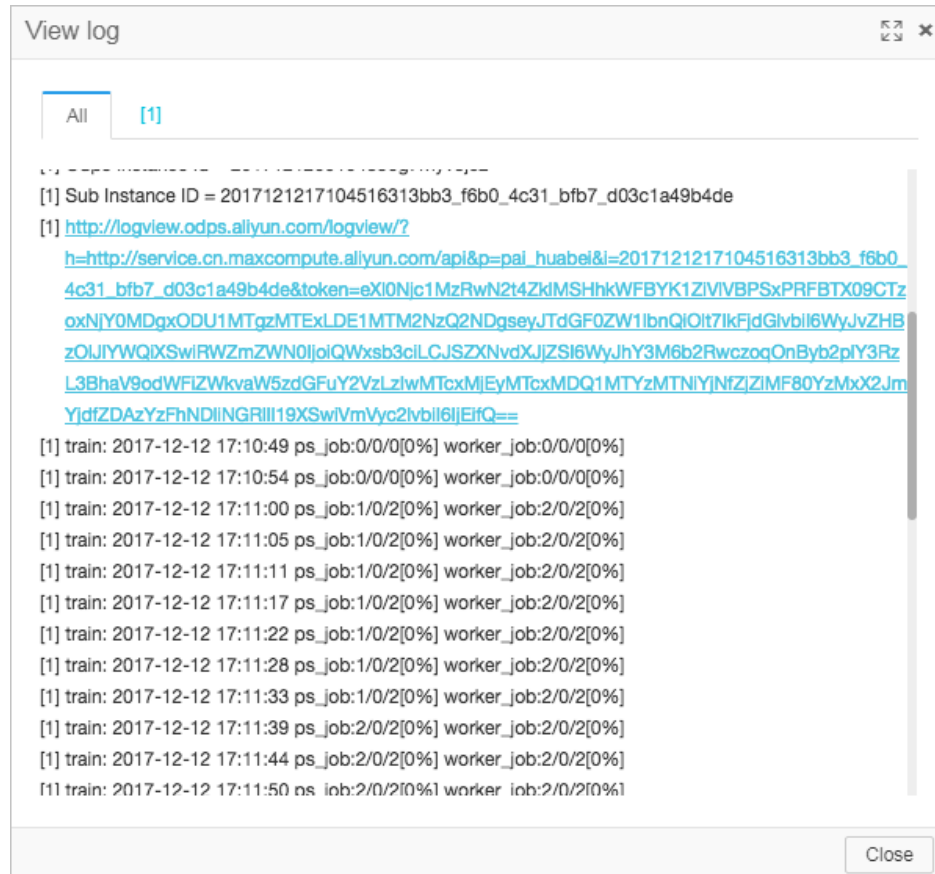
```
network = input_data(shape=[None, 32, 32, 3],
data_preprocessing=img_prep,
data_augmentation=img_aug)
network = conv_2d(network, 32, 3, activation='relu')
network = max_pool_2d(network, 2)
network = conv_2d(network, 64, 3, activation='relu')
network = conv_2d(network, 64, 3, activation='relu')
network = max_pool_2d(network, 2)
network = fully_connected(network, 512, activation='relu')
network = dropout(network, 0.5)
network = fully_connected(network, 10, activation='softmax')
network = regression(network, optimizer='adam',
loss='categorical_crossentropy',
learning_rate=0.001)
```

- The following code generates the model **model.tfl**.

```
model = tflearn.DNN(network, tensorboard_verbose=0)
model.fit(X, Y, n_epoch=100, shuffle=True, validation_set=(X_test, Y_test),
show_metric=True, batch_size=96, run_id='cifar10_cnn')
model_path = os.path.join(FLAGS.checkpointDir, "model.tfl")
print(model_path)
model.save(model_path)
```

5. Log view

Right-click the TensorFlow component to view the logs generated during the training process.



Click a logview link and run the following steps to view the logs.

Open the **Algo Task** under **ODPS Tasks**.

Double-click the **TensorFlow Task**.

Click **MWorker** on the left, and choose **All**.

	FuxiInstance	LogID	StdOut	StdErr	Status	FinishedPercentage
0	MWorker#0_0				Terminated	100%
1	MWorker#1_0				Terminated	100%

Click **StdOut** to print the training logs.


```

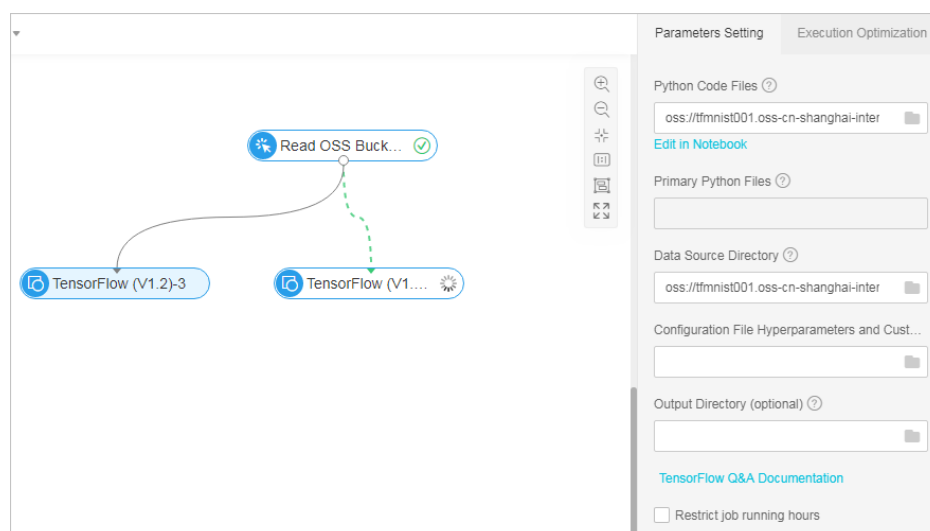
Logview [Stdout]
[2K] Adam | epoch: 100 | loss: 0.26830 - acc: 0.9044 -- iter: 49248/50000
[A [ATraining Step: 52093 | total loss: [1m [32m0.27007 [0m [0m | time: 17.023s
[2K] Adam | epoch: 100 | loss: 0.27007 - acc: 0.9056 -- iter: 49344/50000
[A [ATraining Step: 52094 | total loss: [1m [32m0.27512 [0m [0m | time: 17.057s
[2K] Adam | epoch: 100 | loss: 0.27512 - acc: 0.9088 -- iter: 49440/50000
[A [ATraining Step: 52095 | total loss: [1m [32m0.27783 [0m [0m | time: 17.090s
[2K] Adam | epoch: 100 | loss: 0.27783 - acc: 0.9075 -- iter: 49536/50000
[A [ATraining Step: 52096 | total loss: [1m [32m0.27609 [0m [0m | time: 17.121s
[2K] Adam | epoch: 100 | loss: 0.27609 - acc: 0.9053 -- iter: 49632/50000
[A [ATraining Step: 52097 | total loss: [1m [32m0.27241 [0m [0m | time: 17.153s
[2K] Adam | epoch: 100 | loss: 0.27241 - acc: 0.9043 -- iter: 49728/50000
[A [ATraining Step: 52098 | total loss: [1m [32m0.26988 [0m [0m | time: 17.182s
[2K] Adam | epoch: 100 | loss: 0.26988 - acc: 0.9066 -- iter: 49824/50000
[A [ATraining Step: 52099 | total loss: [1m [32m0.26066 [0m [0m | time: 17.215s
[2K] Adam | epoch: 100 | loss: 0.26066 - acc: 0.9087 -- iter: 49920/50000
[A [ATraining Step: 52100 | total loss: [1m [32m0.24700 [0m [0m | time: 18.614s
[2K] Adam | epoch: 100 | loss: 0.24700 - acc: 0.9136 | val_loss: 0.80838 - val_acc: 0.8175 -- iter:
50000/50000
--
oss://pai-shanghai-test/aohai_test/check_point/model/model.tfl

```

More logs are printed as the experiment continues. You can also use the print function to print key information in the code. In this example, you can use the **aac** parameter to view the accuracy of the model.

6. Result prediction

You can drag another **TensorFlow** component for use in predicting.



- Python Code File: Select the OSS directory of `cifar_predict_pai.py`.
- Data Source Directory: Select the OSS directory of `cifar-10-batches-py`.
- Output Directory: Select the OSS directory of model `model.tfl`.

The image that is used for predicting is stored in the **checkpoint** folder.



The prediction result is as follows:

```
Logview [Stdout]
load data done
oss://pai-shanghai-test/aohai_test/check_point/model/model.tfl
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
This is a bird
```

7. Predicting code explanation

The following code:

```
predict_pic = os.path.join(FLAGS.buckets, "bird_bullocks_oriole.jpg")
img_obj = file_io.read_file_to_string(predict_pic)
file_io.write_string_to_file("bird_bullocks_oriole.jpg", img_obj)

img = scipy.ndimage.imread("bird_bullocks_oriole.jpg", mode="RGB")

# Scale it to 32x32
img = scipy.misc.imresize(img, (32, 32), interp="bicubic").astype(np.float32, casting='unsafe')

# Predict
prediction = model.predict([img])
print (prediction[0])
print (prediction[0])
#print (prediction[0].index(max(prediction[0])))
num=['airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck']
print ("This is a %s"%(num[prediction[0].index(max(prediction[0]))]))
```

- Reads the image "bird_bullocks_oriole.jpg" , and scales the image to 32*32 pixels.
- Passes the image to the function model.predict to evaluate similarity scores.
- Returns the result based on the similarity scores. The class that scores the highest similarity is returned.

Note: Because of the randomness of the model training, it is not guaranteed that the model from each training can return accurate results for the predicted image. It is necessary to continuously debug the corresponding parameters to achieve a stable effect. This case is relatively simple and is for reference only.

Related download

Tensorflow_cifar10 case

Training data

Training code

Predicting code

Predicting image

Multiple workers and tasks in TensorFlow case

For detailed usage, see Deep Learning.

Code download for multiple workers and tasks in TensorFlow case.

Identify the most relevant pollutant for haze

Background



Air pollution has become one of the top 10 issues that people are worried about. Air pollution, or haze, not only affects how people travel and entertain themselves, but also presents a hazard to public health. This example analyzes the weather data of Beijing collected in 2016 and finds that nitrogen dioxide was the most relevant pollutant for haze (PM 2.5).

Log on to Alibaba Cloud Machine Learning Platform for AI (PAI) Studio to create an air pollution haze prediction experiment by using a template.

Dataset

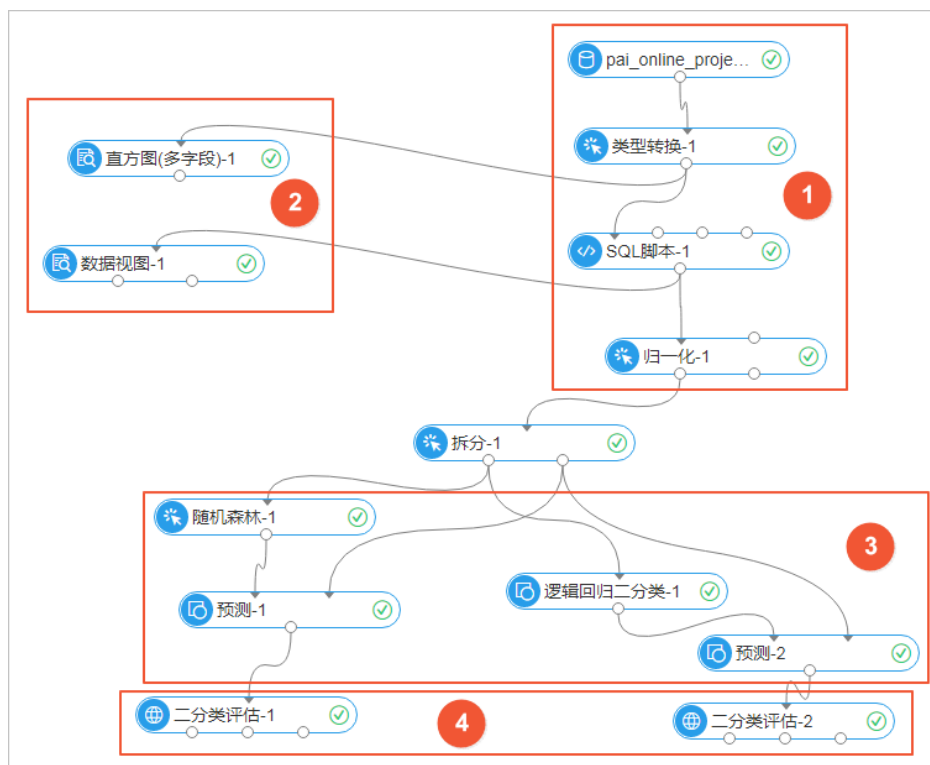
Data source: This dataset was created based on the weather data of Beijing in 2016.

Air index data for each hour since January 1, 2016 was collected. The fields are as follows.

Field	Definition	Type
time	Date, accurate to the day	string
hour	The hour of the data	string
pm2	The PM2.5 index.	string
pm10	The PM10 index.	string
so2	The sulfur dioxide index.	string
co	The carbon monoxide index.	string
no2	The carbon dioxide index.	string

Data exploration procedure

The experiment process is as follows.

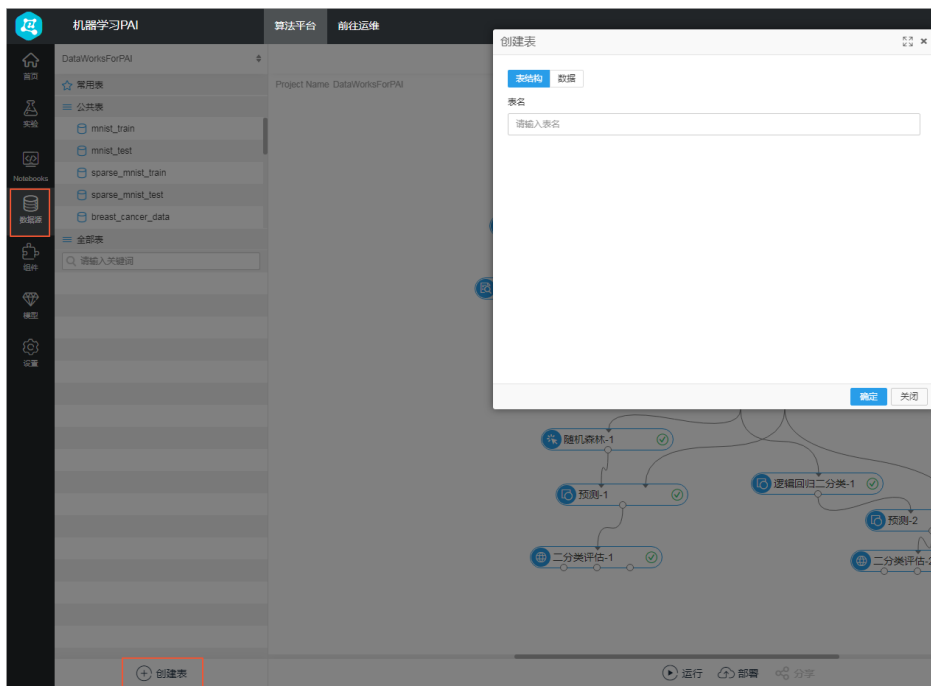


The entire experiment is divided into four parts: data import and preprocessing (1 in the preceding figure), statistical analysis (2 in the preceding figure), model training and prediction (3 in the preceding figure), and model evaluation and analysis (4 in the preceding figure). The details are as follows.

1. Data import and preprocessing

Data import

Click **Data Source**, select **Create Table**, and upload a .txt or .csv file.



After the data is imported, right-click the component and choose **View Data** from the shortcut menu. The result is as follows.

time ▲	hour ▲	pm2 ▲	pm10 ▲	so2 ▲	co ▲	no2 ▲
2016...	2	85	123	18	1.8	72
2016...	8	114	127	25	2.3	81
2016...	11	123	140	27	2.5	83
2016...	14	134	150	30	2.6	86
2016...	17	150	168	32	2.8	92
2016...	20	166	191	34	3	97
2016...	23	179	207	35	3.2	101
2016...	1	190	222	37	3.4	104
2016...	10	225	249	39	3.8	107
2016...	19	244	287	41	4	113

1. Data preprocessing

Convert data of the string type to the double type through the Data Type Conversion component.

Convert the target column to a double type of 0 and 1 through the SQL Script component.

In this experiment, "pm2" is listed as the target column. Values larger than 200 are marked as 1 for heavy haze, and values smaller than or equal to 200 are marked as 0. The SQL statement is as follows.

```
select time, hour, (case when pm2 > 200 then 1 else 0 end), pm10, so2, co, no2 from ${t1};
```

Normalization

Normalization aims to remove the dimension, that is, to unify the units of pollutants with different indexes.

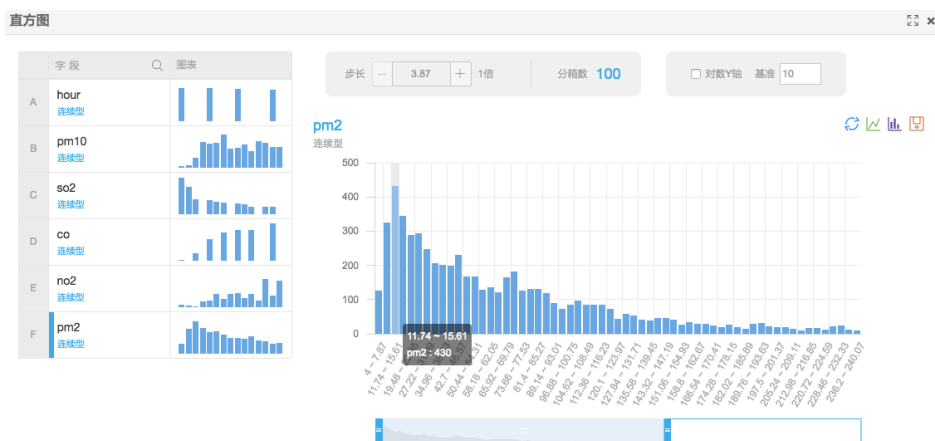
time ▲	hour ▲	_c2 ▲	pm10 ▲	so2 ▲	co ▲	no2 ▲
20160101	2	0	0.24532224...	0.21917808219...	0.36956521739130427	0.43312101910828027
20160101	8	0	0.25363825...	0.31506849315...	0.4782608695652173	0.49044585987261147
20160101	11	0	0.28066528...	0.34246575342...	0.5217391304347825	0.5031847133757962
20160101	14	0	0.30145530...	0.38356164383...	0.5434782608695652	0.5222929936305732
20160101	17	0	0.33887733...	0.41095890410...	0.5869565217391303	0.5605095541401274
20160101	20	0	0.38669438...	0.43835616438...	0.6304347826086956	0.5923566878980892
20160101	23	0	0.41995841...	0.45205479452...	0.6739130434782609	0.6178343949044586
20160102	1	0	0.45114345...	0.47945205479...	0.7173913043478259	0.6369426751592356
20160102	10	1	0.50727650...	0.50684931506...	0.8043478260869563	0.6560509554140127
20160102	19	1	0.58627858...	0.53424657534...	0.8478260869565216	0.6942675159235668
20160102	22	1	0.68191268...	0.53424657534...	0.8913043478260869	0.7197452229299363
20160103	0	1	0.74428274...	0.53424657534...	0.8913043478260869	0.732484076433121
20160105	16	0	0.06860706...	0.02739726027...	0.06521739130434782	0.16560509554140126

2. Statistical analysis

Histogram

The Histogram component allows you to view the distribution of the data in different intervals.

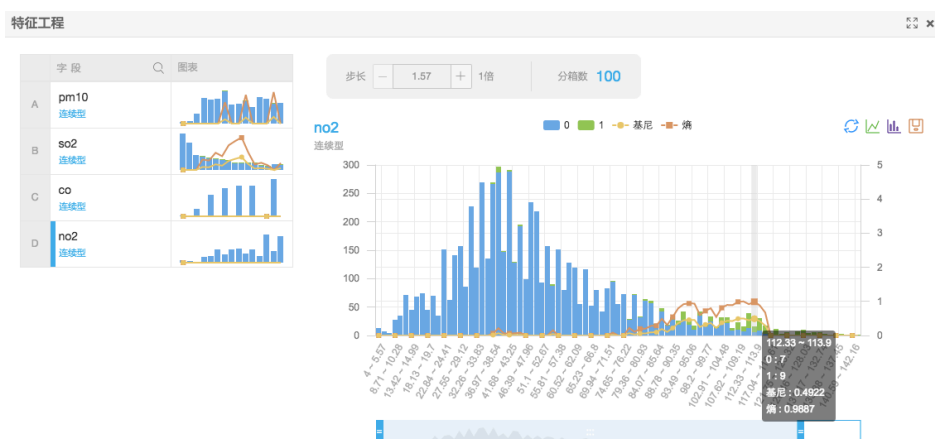
This experiment visually presents the distribution of data in each field. As shown in the following figure, taking PM2.5 (pm2) as an example, the most significant range of values is 11.74 to 15.61, with a total of 430 records.



Data View

The Data View component allows you to view the impact of intervals with different metrics for the prediction results.

For example, seven instances with value 0 and nine instances with value 1 fall into the 112.33 to 113.9 interval. This indicates that when the nitrogen dioxide index is between 112.33 and 113.9, the probability of heavy haze is large. The entropy and Gini coefficient indicate the impact of this feature range on the target value (the impact on the aspect of information), and the larger the value, the greater the impact.



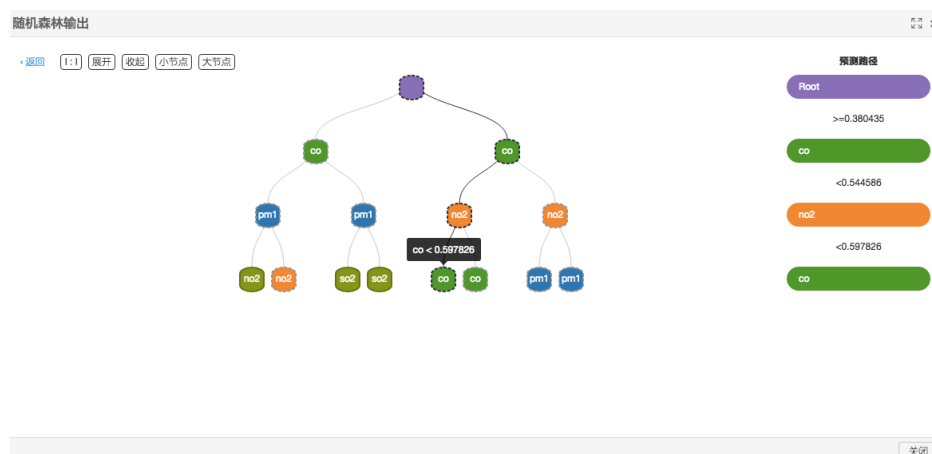
3. Model training and prediction

In this example, two different algorithms are used to predict and analyze the results: random forest and logistic regression.

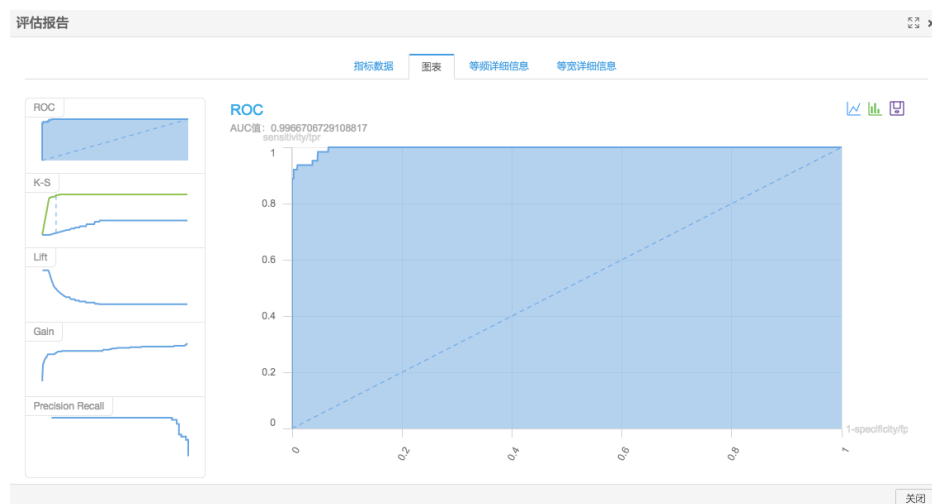
Random forest

The dataset is split, in which 80% is used for model training, and 20% is used for prediction. In the left-side navigation pane of the console, click **Models** and select **Saved Models**. Right-click the model

and choose **Show Model** from the shortcut menu. Then, the tree model of the random forest is visually shown as follows.



The prediction result is as follows.



The AUC in the preceding figure is 0.99, which indicates that with the weather index data used in this example, it can predict whether haze will occur, and the accuracy rate can reach more than 90%.

Logistic regression

A linear model can be obtained by training with the logistic regression algorithm, as shown in the following figure.

逻辑回归二分类



在输入数据为稀疏的时候，不显示 weight 全是 0 的特征

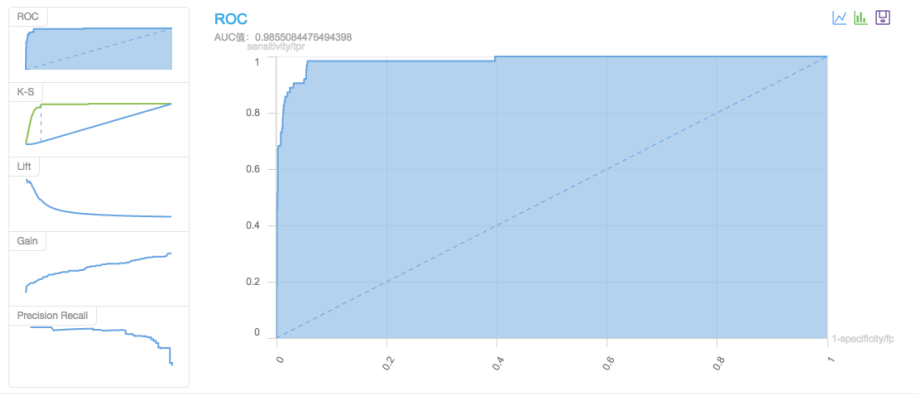
字段名 ▲	权重	
	1 ▲	0 ▲
pm10	18.32146628653672	-
so2	1.767062094833547	-
co	-0.2519492790928399	-
no2	10.95221282178011	-
常量	-16.66654139199668	0

The prediction result is as follows.

评估报告



指标数据 图表 等频详细信息 等宽详细信息



The result shows that the **AUC** is 0.98, which is a little lower than the prediction accuracy based on random forest. If you exclude the impact of parameter adjustments, the two prediction results show that random forest trains your model better than logistic regression.

Model evaluation and analysis

Based on the preceding model and prediction results, the air index with the greatest impact on PM2.5 is analyzed.

The logistic regression model generated is shown in the following figure.

逻辑回归二分类



在输入数据为稀疏的时候，不显示 weight 全是 0 的特征

字段名 ▲	权重	
	1 ▲	0 ▲
pm10	18.32146628653672	-
so2	1.767062094833547	-
co	-0.2519492790928399	-
no2	10.95221282178011	-
常量	-16.66654139199668	0

The impact on the result is proportional to the model coefficient of the logistic regression algorithm after normalized computing. The coefficient symbol is positive for positive correlation and negative for negative correlation. In the preceding figure, pm10 and no2 have the greatest positive coefficients.

- The difference between pm10 and pm2 is the size of fine particles. Therefore, the impact of pm10 is not considered.
- NO2 (nitrogen dioxide) has the greatest impact on PM2.5. You can check the relevant documents to find out which factors will cause a large amount of nitrogen dioxide emissions and identify the major factors that affect PM2.5.

The article [Source of Nitrogen Dioxide](#) from the Internet indicates that nitrogen dioxide mainly comes from vehicle exhaust.

References

You can log on to Alibaba Cloud PAI to experience this product and go to Yunqi Community to discuss it with us.

Issue algriculture loans

The data in this topic is fictitious and is only used for experimental purposes.

Background

Issuing agriculture loans is a typical data mining case. Lenders use an experience model built based on statistics of past years (including a borrower' s yearly income, types of planted crops, loan history,

and other factors) to predict that borrower's repayment ability.

This topic is based on real agriculture loan scenarios and shows how to use the linear regression algorithm to handle loan issuing business. Linear regression is a widely applicable statistics analysis method used in statistics to determine the quantitative relation that two or more variables depend on. This topic predicts whether to issue requested loan amounts to users in the prediction set by analyzing the issuing historical information of agriculture loans.

Dataset

The fields are as follows.

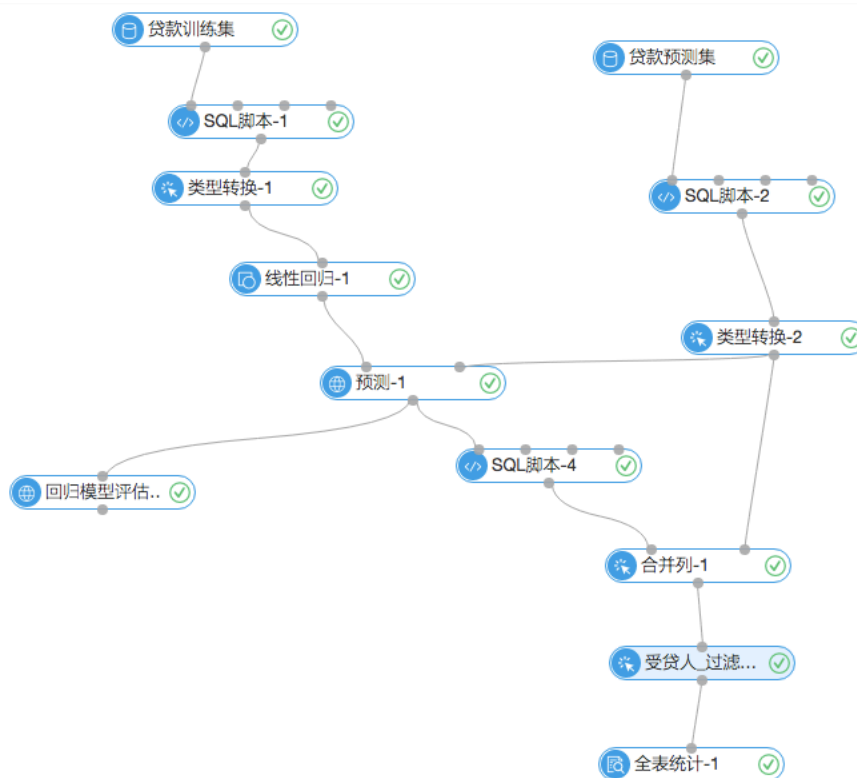
Field	Definition	Type	Description
id	The unique identifier of a data item	string	Person.
name	The name of a user	string	Person.
region	The region where the user is located	string	Arranged from north to south.
farmsize	The size of the farmland owned by the user	double	Farmland area.
rainfall	The rainfall in the region	double	Rainfall.
landquality	The land quality of the region	double	Higher land quality values indicate better land quality.
farmincome	The income of the user from the farmland	double	Yearly income.
maincrop	The crops cultivated on the farmland	string	Types of crops.
claimtype	Loan type	string	Two types.
claimvalue	Loan amount	double	Loan amount.

The following is a screenshot of the data.

id ▲	name ▲	region ▲	farmsize ▲	rainfall ▲	landquality ▲	farmincome ▲	maincrop ▲	claimtype ▲	claimvalue ▲
"id..."	"name..."	"midland..."	1480	30	8	330729	"wheat"	"decommiss..."	74703.1
"id..."	"name..."	"north"	1780	42	9	734118	"maize"	"arable_dev"	245354
"id..."	"name..."	"midland..."	500	69	7	231965	"rapeseed"	"decommiss..."	84213
"id..."	"name..."	"southw..."	1860	103	3	625251	"potatoes"	"decommiss..."	281082
"id..."	"name..."	"north"	1700	46	8	621148	"wheat"	"decommiss..."	122006
"id..."	"name..."	"southea..."	1580	42	7	445785	"maize"	"arable_dev"	122135
"id..."	"name..."	"southea..."	1820	29	6	211605	"maize"	"arable_dev"	68969.2
"id..."	"name..."	"southea..."	1640	108	7	1167040	"maize"	"arable_dev"	485011
"id..."	"name..."	"southw..."	1600	101	5	756755	"wheat"	"decommiss..."	160904
"id..."	"name..."	"southea..."	600	80	6	267928	"wheat"	"arable_dev"	90350.6

Data exploration procedure

The following figure shows the experiment process.



1. Data source preparation

Input data is divided into two parts:

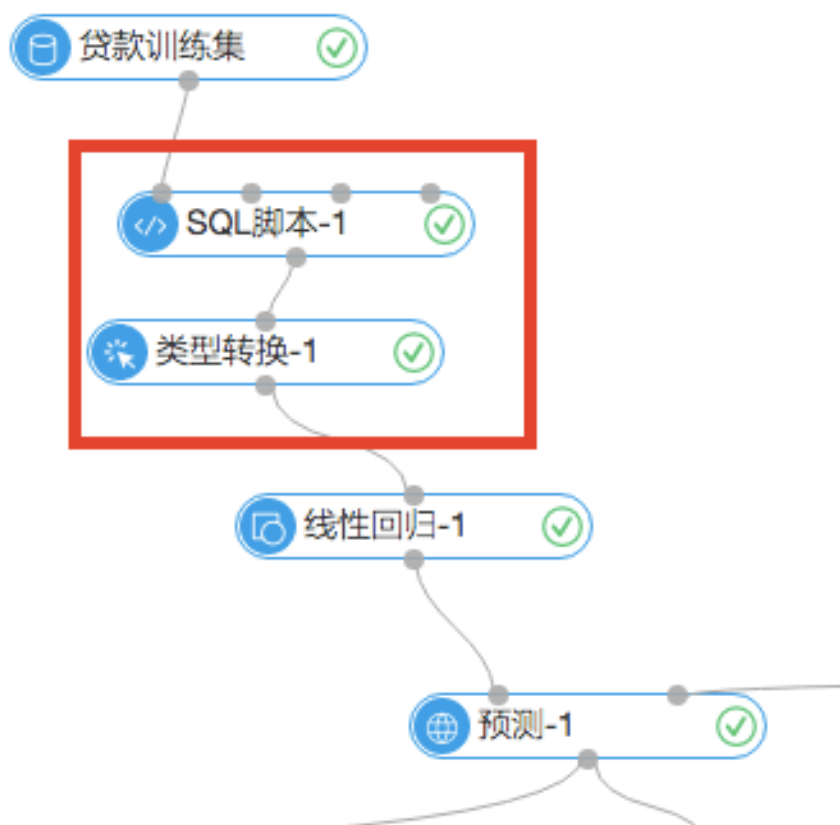
- Loan training set: More than 200 pieces of loan data are used to train the regression model. This training set includes features such as "farmsize" and "rainfall". "claimvalue" is the recovered loan amount.
- Loan prediction set: This prediction set includes a total of 71 loan applicants this year. "claimvalue" is a farmer's requested loan amount.

Predicate whom of the 71 applicants will receive loans based on the existing 200+ pieces of historical

data.

2. Data preprocessing

Map data of the string type to numbers according to data meanings. For example, for the “region” field, map “north” , “middle” , and “south” in order to 0, 1, and 2, respectively. Then, convert the field to the double type by using the Data Type Conversion component, as shown in the following figure. You can perform model training after data is preprocessed.



3. Model training and prediction

Use the Linear Regression component to train historical data and create a regression model, which is used in the Prediction component to predict data in the prediction set. Use the Merge Columns component to merge the user ID, prediction score, and claim value, as shown in the following figure. The prediction score indicates a user’s loan repayment ability (expected loan repayment amount).

claimvalue ▲	prediction_score ▲	id ▲
172753	164424.3413395547	1
93415.4	146370.52166158534	2
46800.2	41879.999271195346	3
131728	192648.19077439874	4
89040.8	76369.8134277192	5
135493	103695.67105783387	6
88906.8	136845.30246967232	7
147159	144156.81362150217	8
277397	466728.8170899566	9
67547.3	131340.40980772747	10
345394	402192.7992950041	11

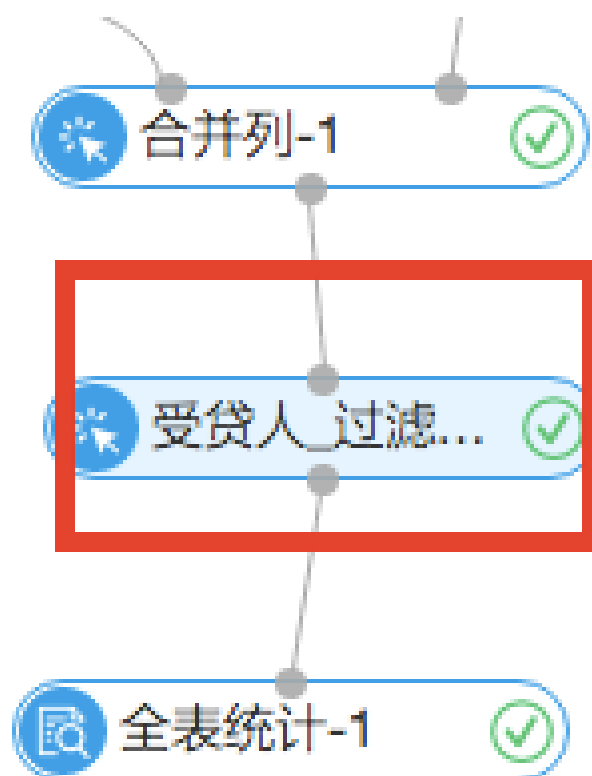
4. Regression model evaluation

Use the Regression Model Evaluation component to evaluate the model. The following table lists evaluation results.

字段名称	描述
SST	总平方和
SSE	误差平方和
SSR	回归平方和
R2	判定系数
R	多重相关系数
MSE	均方误差
RMSE	均方根误差
MAE	平均绝对误差
MAD	平均误差
MAPE	平均绝对百分误差
count	行数
yMean	原始因变量的均值
predictionMean	预测结果的均值

5. Loan issuance

Use the Filtering and Mapping component to determine the applicants who can receive loans. The principle of the experiment is that, if an applicant's repayment ability is predicted to be greater than the requested loan amount, that applicant will receive a loan. This principle applies to each potential customer.



References

You can log on to Alibaba Cloud Machine Learning Platform for AI (PAI) to experience this product and go to Yunqi Community to discuss with us.

Identify electricity theft

The conventional measures to identify power theft, electricity leakage, and electricity meter faults include periodic inspection, periodic electricity meter check, and user reporting. However, these measures depend on manual work and lack specific targets.

Currently, many power supply bureaus use the metering alert function and the power data query function to monitor users' power usage online. The bureau staff identify power theft, electricity leakage, and electricity meter faults by collecting information such as abnormal power usage, electricity load exceptions, terminal alerts, primary site alerts, and line exceptions or losses. A model is created to analyze abnormal power usage based on the metric weights by collecting statistics on the current, voltage, and load at the metering point before and after an alert is triggered. This helps

identify power theft, illegal power usage, and electricity meter faults.

The analysis model can collect information about abnormal power usage but fails to identify the users suspected of power theft or electricity leakage in a fast and accurate manner. This is a big challenge for the audit staff. The analysis model requires expert knowledge and experience to determine the weights of input metrics. This process is flawed and depends on subjective judgment, producing unsatisfactory results.

An automatic electricity metering system can collect statistics on electricity load, such as current in all phases, voltage, and power factor, as well as terminal alerts such as abnormal power usage. Alerts and electricity load data can reflect users' power usage. The audit staff can identify users suspected of power theft and electricity leakage through an online audit system and onsite audit, and import findings to the system.

The imported data is analyzed to extract the key features of power theft and electric leakage and create a model used to automatically check and identify power thieves and households with electric leakage. This greatly reduces the workload of the audit staff and ensures normal and safe power usage.

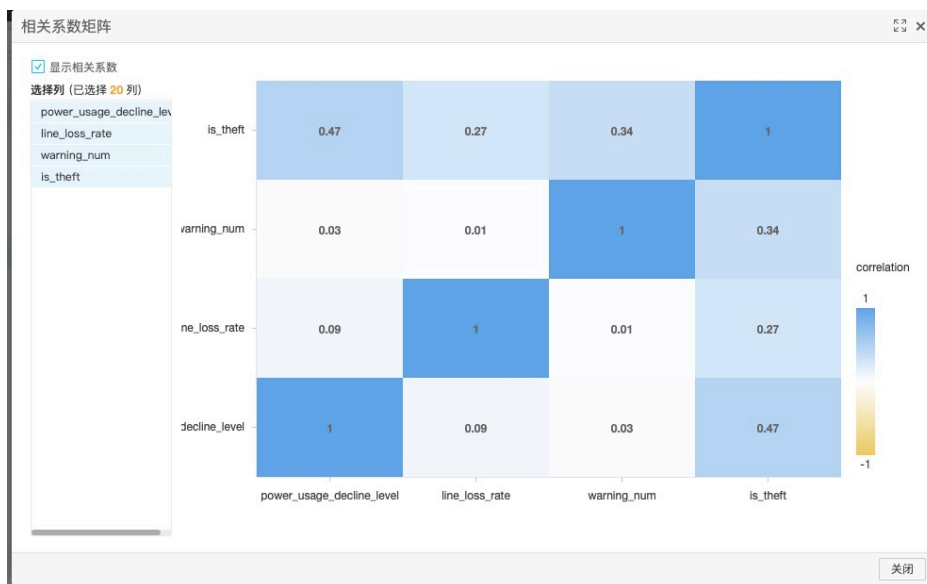
Load and explore data

You can select a dataset to view three metrics about a user's power usage and the data that indicates whether the user steals power or encounters electric leakage. The three metrics are power usage decline level, line loss rate, and warning num. The "is theft" column lists the metric analysis result.

数据探查 - anti_electricity_theft - (仅显示前一百条)

序号	power_usage_decline_level	line_loss_rate	warning_num	is_theft
1	4	1	1	1
2	4	0	4	1
3	2	1	1	1
4	9	0	0	0
5	3	1	0	0
6	2	0	0	0
7	5	0	2	1
8	3	1	3	1

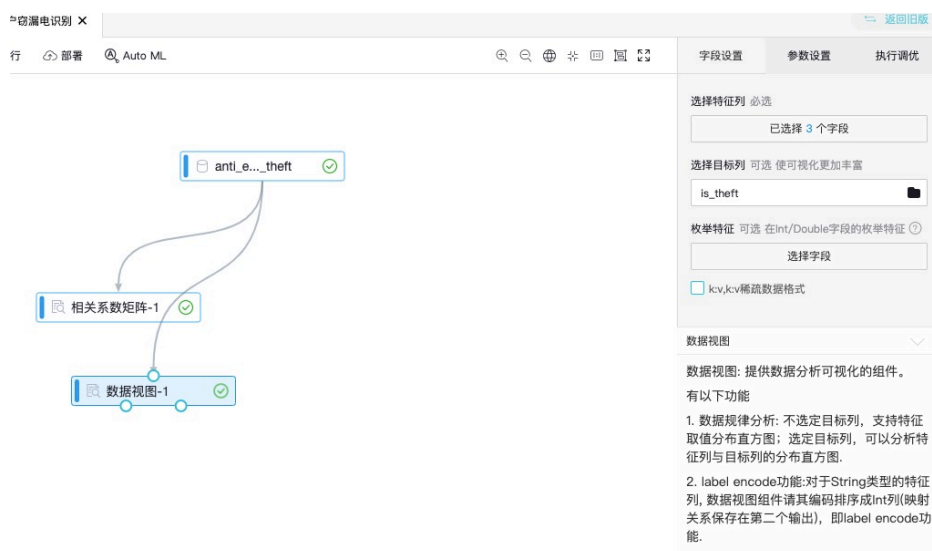
In the left-side navigation pane, choose **Components** > **Statistical Analysis**, and drag and drop **Correlation Coefficient Matrix** to the right section to view each feature related to the output power.



Right-click the completed component and select **View Analytics Report** to obtain the correlation analysis result. The correlation chart shows that the three metrics are not closely related to the result of "is theft". That is, the features are not specific enough to determine whether a user is a power thief. Then, in the left-side navigation pane, choose **Components > Statistical Analysis**, and drag and drop **Data View** to the right section to analyze the distribution of data in the label column by feature. Select the feature columns as follows.

字段	类型
power_usage_decline_level	BIGINT
line_loss_rate	BIGINT
warning_num	BIGINT

Then, select the label column.

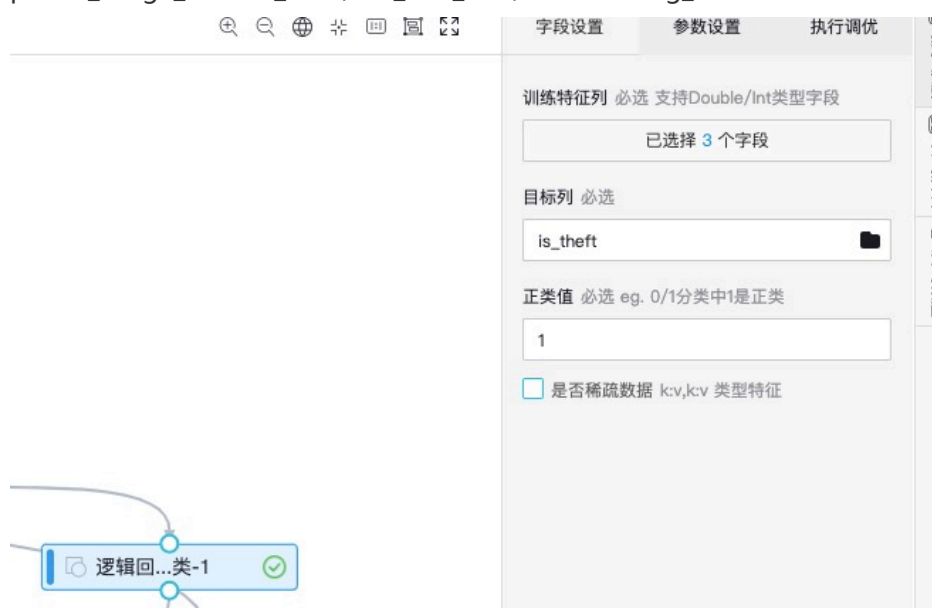


Finally, right-click Run from Here, choose the completed component, and select **View Analytics Report** to view the distribution of data in the label column by feature.

Model data

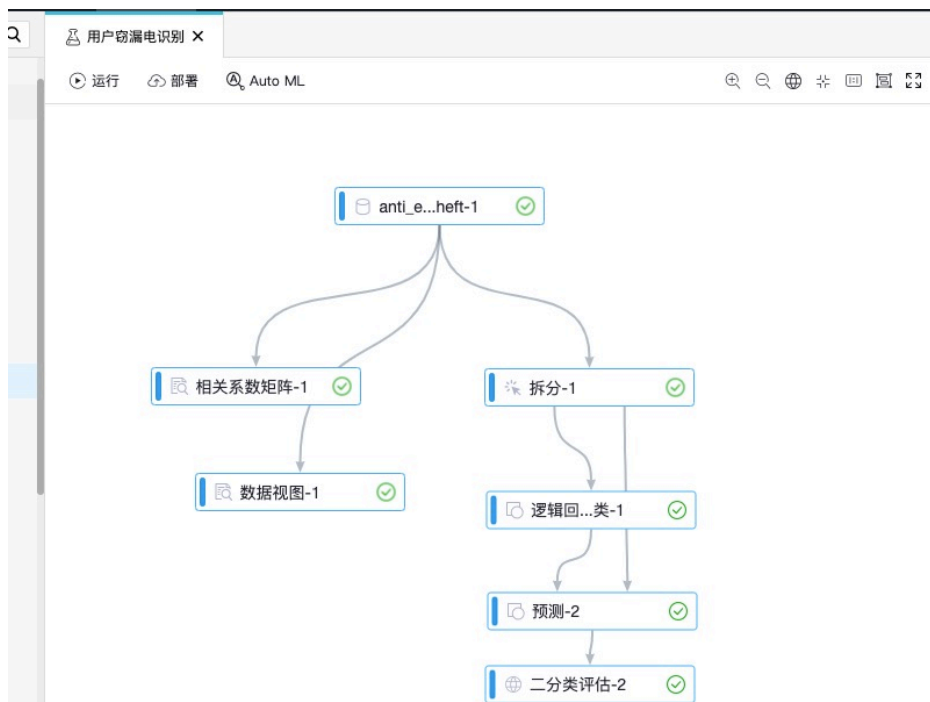
After completing a simple exploratory analysis, you can select an appropriate algorithm model for data modeling. In the left-side navigation pane, choose **Components > Data Preprocessing**, and drag and drop **Split** to the right section to split data into the training set and test set.

Choose **Components > Machine Learning > Binary Classification**, and drag and drop **Logistic Regression for Binary Classification** to the right section to perform regression modeling on data. Select the feature columns (X) and label column (Y). In this experiment, the feature columns are power_usage_decline_level,line_loss_rate, and warning_num.



Predict and evaluate the regression model

After modeling is complete, choose **Components > Machine Learning** and drag and drop **Prediction** to the right section to predict the effect of the model on the test dataset. For **Feature Columns** and **Reversed Output Column**, all options are selected by default. In the left-side navigation pane, choose **Components > Machine Learning > Evaluation**, and drag and drop **Binary Classification Evaluation** to the right section to view the model effect. The following figure shows the result of the experiment.



Right-click the Binary Classification Evaluation component to view the model effect. The AUC reaches the satisfying value 0.9827.



This completes the identification of power theft through Machine Learning Platform for AI (PAI). You can use Elastic Algorithm Service (EAS) to deploy the identification service so that it can be called online to identify power theft in power grids.

This experiment references the book “Python Practice of Data Analysis and Mining.” For copyright issues, contact the author of this topic. We respect every researcher in the academic field for their academic contribution and strive to better integrate technologies with the real life.

Use the FastNN repository

Machine Learning Platform for AI (PAI) provides Fast Neural Networks (FastNN), which is a distributed neural network repository based on the PAISoar framework. Currently, FastNN supports classic algorithms such as Inception, Resnet, and VGG. More advanced models will be available in the future. FastNN is built into PAI Studio. To try it out, log on to PAI Studio and create an experiment by clicking the corresponding template on the homepage.

Custom development method

1. Data source preparation

To facilitate trying FastNN in PAI, we have downloaded and converted the cifar10, mnist, and flowers data to tfrecord data and stored the converted data in the open Object Storage Service (OSS). The data can be accessed through the Read File Data or OSS Data Synchronization components of PAI. The following table lists the storage paths in OSS.

Dataset	Number of classes	Training set	Test set	Storage path
mnist	10	3320	350	China (Beijing): oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/fastnn-data/mnist/ China (Shanghai): oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/mnist/
cifar10	10	50000	10000	China (Beijing): oss://pai-online-beijing.oss-cn-

				beijing-internal.aliyuncs.com/fastnn-data/cifar10/China (Shanghai): oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/cifar10/
flowers	5	60000	10000	China (Beijing): oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/fastnn-data/flowers/China (Shanghai): oss://pai-online.oss-cn-shanghai-internal.aliyuncs.com/fastnn-data/flowers/

To access a data source, write its path in the component.



The FastNN repository supports reading data in the tfrecord format and implements the dataset pipeline based on the TFRecordDataset API for model training. This covers a majority of the data preprocessing time. The current implementation logic in data sharding is not refined enough. Users must ensure that data is evenly distributed to each machine during data preparation. That is:

- The number of samples for each tfrecord file must be almost the same.
- The number of tfrecord files processed by each worker must be almost the same.

If the data format is tfrecord, see the files in cifar10/mnist/flowers of datasets. The cifar10 data is used as an example.

- Assume that the key_to_features format of cifar10 data is as follows.

```
features={
  'image/encoded': tf.FixedLenFeature((), tf.string, default_value=""),
```

```
'image/format': tf.FixedLenFeature([], tf.string, default_value='png'),
'image/class/label': tf.FixedLenFeature(
[], tf.int64, default_value=tf.zeros([], dtype=tf.int64)),
}
```

- Create the data parsing file `cifar10.py` in the `datasets` directory and edit the following sample content:

```
"""Provides data for the Cifar10 dataset.
The dataset scripts used to create the dataset can be found at:
datasets/download_and_convert_data/download_and_convert_cifar10.py
"""

from __future__ import division
from __future__ import print_function

import tensorflow as tf
"""Expect func_name is 'parse_fn'
"""

def parse_fn(example):
    with tf.device("/cpu:0"):
        features = tf.parse_single_example(
            example,
            features={
                'image/encoded': tf.FixedLenFeature([], tf.string, default_value=""),
                'image/format': tf.FixedLenFeature([], tf.string, default_value='png'),
                'image/class/label': tf.FixedLenFeature(
                    [], tf.int64, default_value=tf.zeros([], dtype=tf.int64)),
            }
        )
        image = tf.image.decode_jpeg(features['image/encoded'], channels=3)
        label = features['image/class/label']
        return image, label
```

- Add `dataset_map` in `datasets/dataset_factory.py`.

```
from datasets import cifar10
datasets_map = {
    'cifar10': cifar10,
}
```

- When running the workflow script, use `cifar10` data for model training by setting `dataset_name` to `cifar10` and `train_files` to `cifar10_train.tfrecord`.

To read data in other formats, implement the dataset pipeline construction logic. For more information, see `utils/dataset_utils.py`.

2. Hyperparameter file

The following types of hyperparameters are supported:

- Dataset parameters: basic attributes of the training set, such as the training set storage path `dataset_dir`.
- Data preprocessing parameters: data preprocessing functions and parameters related to the dataset pipeline.
- Model parameters: basic hyperparameters for model training, including `model_name` and `batch_size`.
- Learning rate parameters: learning rate and related tuning parameters.
- Optimizer parameters: optimizer and related parameters.
- Log parameters: parameters of output logs.
- Performance tuning parameters: mixed precision and other tuning parameters.

Sample hyperparameter file:

```
1 enable_paisoar=True
2 batch_size=128
3 use_fp16=True
4 dataset_name=flowers
5 dataset_dir=oss://pai-online-beijing.oss-cn-beijing-internal.aliyuncs.com/
  fastnn-data/flowers
6 model_name=inception_resnet_v2
7 optimizer=sgd
8 num_classes=5
9 job_name=worker
```

2.1 Dataset parameters

#Parameter	#Type	#Description
Dataset_name	string	The name of the input data parsing file. Valid values: mock, cifar10, mnist, and flowers. For more information, see all data parsing files in the images/datasets directory. Default value: mock, indicating analog data.
dataset_dir	string	The absolute path of the input dataset. Default value: None.
num_sample_per_epoch	integer	The total number of dataset samples. This parameter is typically used with learning rate decay.
num_classes	integer	The number of sample classes. Default value: 100.
train_files	string	The file names of all training data, which are separated with commas (,), such as "0.tfrecord,1.tfrecord" .

2.2 Data preprocessing parameters

#Parameter	#Type	#Description
preprocessing_name	string	Used with model_name to specify the name of the data preprocessing method. For more information about the current value range, see the preprocessing_factory file in the images/preprocessing directory. Default value: None, indicating no data preprocessing.
shuffle_buffer_size	integer	The size of the buffer pool for sample-based shuffle when a data pipeline is created. Default value: 1024.
num_parallel_batches	integer	The number of parallel threads multiplied by batch_size to equal map_and_batch. This parameter helps specify the parallel granularity of parsing samples. Default value: 8.
prefetch_buffer_size	integer	The number of batches of data prefetched by the data pipeline. Default value: 32.
num_preprocessing_threads	integer	The number of threads used by the data pipeline to prefetch data in parallel. Default value: 16.
datasets_use_caching	bool	Specifies whether to enable caching for compressed input data with memory overhead. Default value: False, indicating that caching is not enabled.

2.3 Model parameters

#Parameter	#Type	#Description
task_type	string	Valid values: pretrain and finetune, which indicate model pre-training and model optimization, respectively. Default value: pretrain.
model_name	string	The model to be trained. The valid values include all

		models in images/models. You can set model_name based on all models defined in the images/models/model_factor_y file. Default value: inception_resnet_v2.
num_epochs	integer	The number of training rounds for the training set. Default value: 100.
weight_decay	float	The weight decay factor during model training. Default value: 0.00004.
max_gradient_norm	float	Specifies whether to perform gradient clipping based on the global normalization value. Default value: None, indicating no gradient clipping.
batch_size	integer	The amount of data that is processed by one card in an iteration. Default value: 32.
model_dir	string	The path to reload the checkpoint. Default value: None, indicating no model optimization.
ckpt_file_name	string	The name of the file that reloads the checkpoint. Default value: None.

2.4 Learning rate parameters

#Parameter	#Type	#Description
warmup_steps	integer	The number of iterations for inverse learning rate decay. Default value: 0.
warmup_scheme	string	The way of inverse learning rate decay. The valid value 't2t' indicates Tensor2Tensor, in which the learning rate is initialized to be 1/100 of the specified learning rate and then is exponentiated to inverse-decay to the specified learning rate.
decay_scheme	string	The way of learning rate decay. Valid values: luong234, luong5, and

		luong10. luong234 indicates to start 4 rounds of decay with a factor of 1/2 after 2/3 of total iterations are completed. luong5 indicates to start 5 rounds of decay with a factor of 1/2 after 1/2 of total iterations are completed. luong10 indicates to start 10 rounds of decay with a factor of 1/2 after 1/2 of total iterations are completed.
learning_rate_decay_factor	float	The learning rate decay factor. Default value: 0.94.
learning_rate_decay_type	string	The learning rate decay type. Valid values: fixed, exponential, and polynomial. Default value: exponential.
learning_rate	float	The initial learning rate. Default value: 0.01.
end_learning_rate	float	The minimum learning rate during decay. Default value: 0.0001.

2.5 Optimizer parameters

#Parameter	#Type	#Description
optimizer	string	The name of the optimizer. Valid values: adadelta, adagrad, adam, ftrl, momentum, sgd, rmsprop, and adamweightdecay. Default value: rmsprop.
adadelta_rho	float	The decay factor of Adadelta. Default value: 0.95. This parameter is specific to the Adadelta optimizer.
adagrad_initial_accumulator_value	float	The initial value of the AdaGrad accumulator. Default value: 0.1. This parameter is specific to the AdaGrad optimizer.
adam_beta1	float	The exponential decay rate in primary momentum prediction. Default value: 0.9. This parameter is specific to the Adam optimizer.
adam_beta2	float	The exponential decay rate in

		secondary momentum prediction. Default value: 0.999. This parameter is specific to the Adam optimizer.
opt_epsilon	float	The offset of the optimizer. Default value: 1.0. This parameter is specific to the Adam optimizer.
ftrl_learning_rate_power	float	The idempotent parameter of the learning rate. Default value: -0.5. This parameter is specific to the FTRL optimizer.
ftrl_initial_accumulator_value	float	The starting point of the FTRL accumulator. Default value: 0.1. This parameter is specific to the FTRL optimizer.
ftrl_l1	float	The regularization term of FTRL l1. Default value: 0.0. This parameter is specific to the FTRL optimizer.
ftrl_l2	float	The regularization term of FTRL l2. Default value: 0.0. This parameter is specific to the FTRL optimizer.
momentum	float	The momentum parameter of MomentumOptimizer. Default value: 0.9. This parameter is specific to the Momentum optimizer.
rmsprop_momentum	float	The momentum parameter of RMSPropOptimizer. Default value: 0.9.
rmsprop_decay	float	The decay factor of RMSProp. Default value: 0.9.

2.6 Log parameters

#Parameter	#Type	#Description
stop_at_step	integer	The total number of training iterations. Default value: 100.
log_loss_every_n_iters	integer	The iterative frequency for printing the loss information. Default value: 10.
profile_every_n_iters	integer	The iterative frequency for printing the timeline. Default

		value: 0.
profile_at_task	integer	The index of the machine that outputs the timeline. Default value: 0, which corresponds to the chief worker.
log_device_placement	bool	Specifies whether to print the device placement information. Default value: False.
print_model_statistics	bool	Specifies whether to print the trainable variable information. Default value: False.
hooks	string	The training hooks. Default value: StopAtStepHook,ProfilerHook,LoggingTensorHook,CheckpointSaverHook.

2.7 Performance tuning parameters

#Parameter	#Type	#Description
use_fp16	bool	Specifies whether to perform semi-precision training. Default value: True.
loss_scale	float	The coefficient of the loss value scale during training. Default value: 1.0.
enable_paisoar	bool	Specifies whether to use PAISoar. Default value: True.
protocol	string	By default, the grpc.rdma cluster can use grpc+verbs to improve data access efficiency.

3. Master file development

If existing models cannot meet your needs, you can use the dataset, models, and preprocessing APIs for further development. Before that, you need to understand the basic process of the FastNN repository. Take images as an example. The code entry file is `train_image_classifiers.py`. The overall code architecture is as follows.

```
# Initialize the corresponding model in models based on model_name to obtain the network_fn. The input
parameter train_image_size may be returned.
```

```

network_fn = nets_factory.get_network_fn(
    FLAGS.model_name,
    num_classes=FLAGS.num_classes,
    weight_decay=FLAGS.weight_decay,
    is_training=(FLAGS.task_type in ['pretrain', 'finetune']))
# Initialize the corresponding data preprocessing function based on model_name or preprocessing_name to obtain
preprocess_fn.
preprocessing_fn = preprocessing_factory.get_preprocessing(
    FLAGS.model_name or FLAGS.preprocessing_name,
    is_training=(FLAGS.task_type in ['pretrain', 'finetune']))
# Select the correct tfrecord format based on dataset_name and synchronously call preprocess_fn to parse the
dataset to obtain dataset_iterator.
dataset_iterator = dataset_factory.get_dataset_iterator(FLAGS.dataset_name,
    train_image_size,
    preprocessing_fn,
    data_sources,
    # Call network_fn and dataset_iterator to define the function loss_fn to calculate the loss.
    def loss_fn():
        with tf.device('/cpu:0'):
            images, labels = dataset_iterator.get_next()
            logits, end_points = network_fn(images)
            loss = tf.losses.sparse_softmax_cross_entropy(labels=labels, logits=tf.cast(logits, tf.float32), weights=1.0)
            if 'AuxLogits' in end_points:
                loss += tf.losses.sparse_softmax_cross_entropy(labels=labels, logits=tf.cast(end_points['AuxLogits'], tf.float32),
                    weights=0.4)
            return loss
    # Call the PAI-Soar API to encapsulate the native optimizer of loss_fn and tf.
    opt = paioar.ReplicatedVarsOptimizer(optimizer, clip_norm=FLAGS.max_gradient_norm)
    loss = optimizer.compute_loss(loss_fn, loss_scale=FLAGS.loss_scale)
    # Give a formal definition of training tensors based on opt and loss.
    train_op = opt.minimize(loss, global_step=global_step)

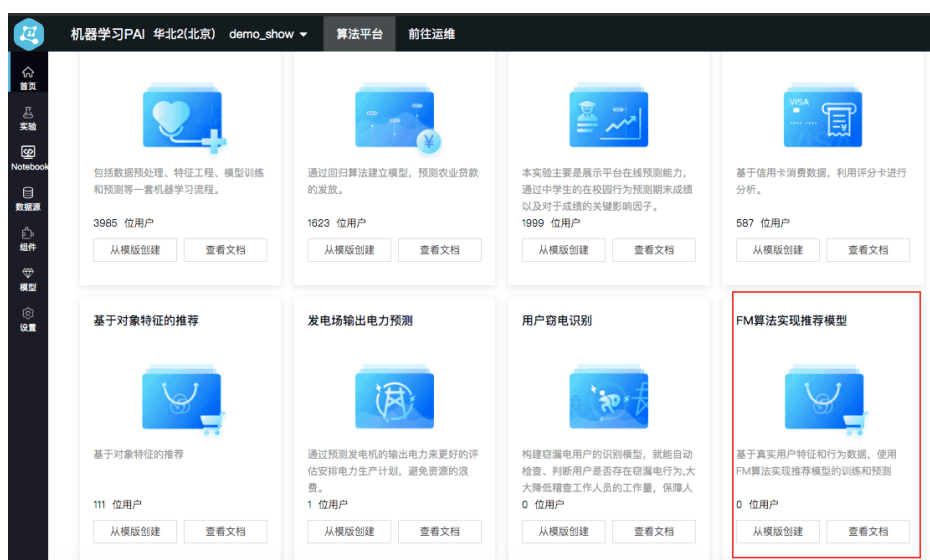
```

Use the FM algorithm of PAI to create a recommendation model

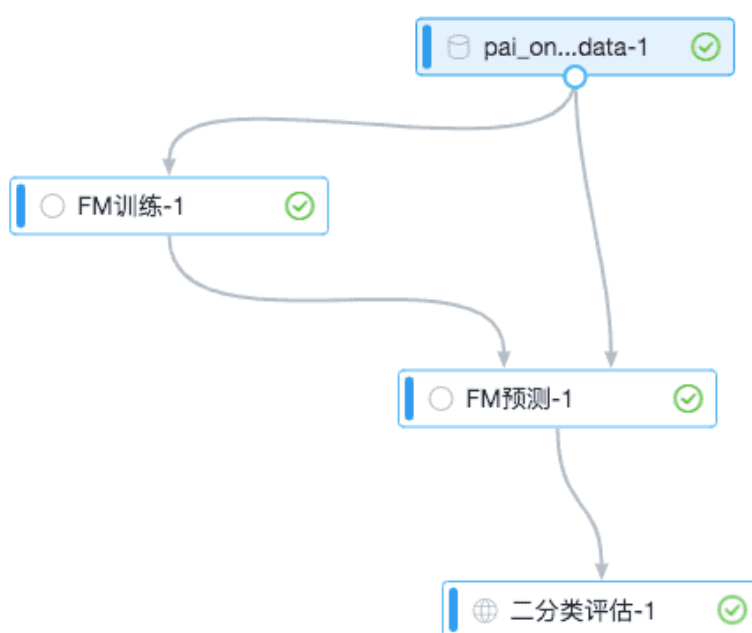
Overview

The Factorization Machine (FM) algorithm can be used for regression and binary classification prediction. It is a nonlinear model that takes into account the interaction between features. Currently, the FM algorithm is one of the proven effective recommendation solutions and is widely used in the recommendation scenarios of e-commerce, advertising, and live streaming.

The FM algorithm used by Machine Learning Platform for AI (PAI) is developed based on big data within Alibaba. It features excellent performance and outstanding results. For the FM algorithm usage, see the corresponding template on the homepage.



The FM algorithm involves the FM training and prediction components, which can be used with the evaluation component.



Required input data

Currently, the FM algorithm only supports data in the libsvm format. The data is divided into two columns: feature column and target column.

- Target column: double type

- Feature column: string type. Features must be entered in the k:v format and separated with commas (,).

See the following figure.

序号 ▲	label ▲	features ▲
1	0	3:1,11:1,14:1,19:1,39:1,42:1,55:1,64:1,67:1,73:1,75:1,76:1,80:1,83:1
2	0	3:1,6:1,17:1,27:1,35:1,40:1,57:1,63:1,69:1,73:1,74:1,76:1,81:1,103:1
3	0	4:1,6:1,15:1,21:1,35:1,40:1,57:1,63:1,67:1,73:1,74:1,77:1,80:1,83:1
4	0	5:1,6:1,15:1,22:1,36:1,41:1,47:1,66:1,67:1,72:1,74:1,76:1,80:1,83:1
5	0	2:1,6:1,16:1,22:1,36:1,40:1,54:1,63:1,67:1,73:1,75:1,76:1,80:1,83:1
6	0	2:1,6:1,14:1,20:1,37:1,41:1,47:1,64:1,67:1,73:1,74:1,76:1,82:1,83:1
7	0	1:1,6:1,14:1,22:1,36:1,42:1,49:1,64:1,67:1,72:1,74:1,77:1,80:1,83:1
8	0	1:1,6:1,17:1,19:1,39:1,42:1,53:1,64:1,67:1,73:1,74:1,76:1,80:1,83:1
9	0	2:1,6:1,18:1,20:1,37:1,42:1,48:1,64:1,71:1,73:1,74:1,76:1,81:1,83:1
10	1	5:1,11:1,15:1,32:1,39:1,40:1,52:1,63:1,67:1,73:1,74:1,76:1,78:1,83:1

Components

1. FM training

In **Parameters Setting**, you can set **Regression** or **Binary Classification**.



PAI commands

Parameter	Description	Value
tensorColName	The name of the feature column for training, expressed by a string in the k:v format, such as 1:1.0,3:1.0. The feature ID	Required

	must be a non-negative integer. The value range is [0, Long.MAX_VALUE). Nonconsecutive values are allowed.	
labelColName	The name of the label column. The value must be a number. If the task type is <code>binary_classification</code> , the value is either 0 or 1.	Required
task	The task type.	Required. Valid values: <code>regression</code> and <code>binary_classification</code>
numEpochs	The number of iterations.	Optional. Default value: 10
dim	The number of factors, expressed by a string that consists of three integers separated with commas (,) to indicate the length of constant term, linear term, and quadratic term.	Optional. Default value: 1,1,10
learnRate	The learning rate.	Optional. Default value: 0.01
lambda	The regularization coefficient, expressed by a string that consists of three floating-point numbers separated with commas (,) to indicate the regularization coefficients of constant term, linear term, and quadratic term.	Optional. Default value: 0.01,0.01,0.01
initStdev	The standard deviation of parameter initialization.	Optional. Default value: 0.05

Note:

- Reduce the learning rate in the case of training divergence.

2. FM prediction

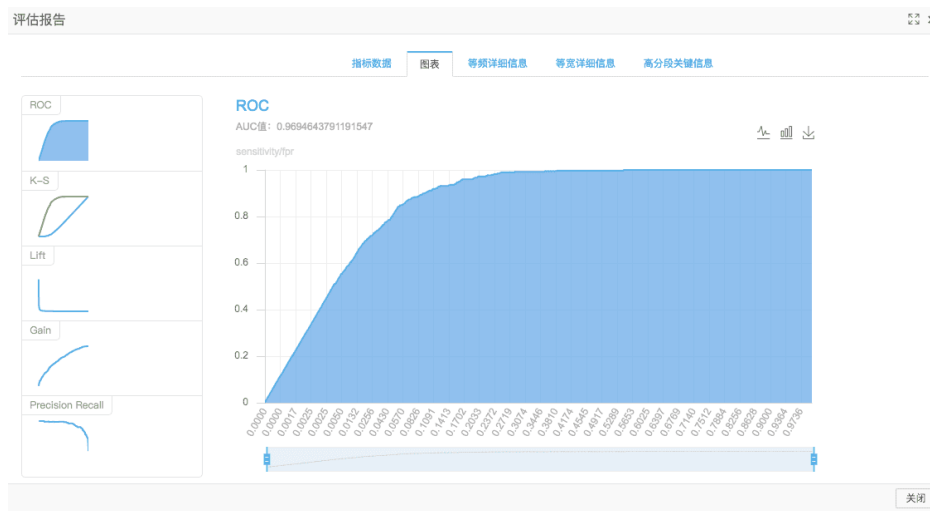
PAI commands

Parameter	Description	Value
predResultColName	The name of the prediction result column.	Optional. Default value: <code>prediction_result</code>
predScoreColName	The name of the prediction score column.	Optional. Default value: <code>prediction_score</code>
predDetailColName	The name of the prediction	Optional. Default value:

	detail column.	prediction_detail
keepColNames	The columns saved to the output result table.	Optional. Default value: all columns

Result evaluation

Using the data of the corresponding template on the homepage, the FM algorithm of PAI can create a model with an AUC close to 0.97.



Use FM-Embedding for recommendation - vector-based recall.md

Background

The data and procedure of the experiment are built in the corresponding template on the home page of Machine Learning Platform for AI (PAI) Studio at <https://data.aliyun.com/product/learn>

Log on to PAI Studio. In the lower part of the **FM-Embedding for Rec-System** template, click **Create**. The template is ready for use.

推荐场景-FM向量召回



基于FM-Embedding的推荐召回方案

0 位用户

[从模版创建](#)[查看文档](#)

AI-based recommendation is divided into two modules: sorting and recall. The recall module uses vectors to represent users and to-be-recommended items. The product of the vectorized user and item indicates the user's interest in the item. The following experiment shows how to create descriptive vectors for users and items based on real-life recommendation data by using the Factorization Machine (FM) algorithm and the Embedding algorithm that are provided by PAI.

Procedure

Flowchart:



1. Data

Raw data:

userid ▲	age ▲	gender ▲	itemid ▲	price ▲	size ▲	label ▲
1	64	male	A	500	10	1
2	42	female	B	200	4	0
3	42	male	C	425	6	1
4	53	female	D	474	3	0
5	57	male	E	64	7	0
6	86	female	F	532	3	0
7	34	female	G	42	4	1
8	23	male	H	364	6	0
9	14	female	I	57	4	0
10	35	male	J	463	9	1

Data fields:

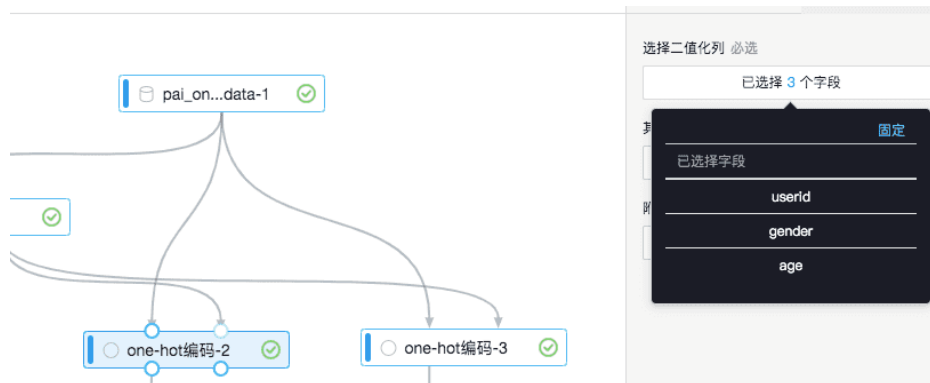
- userid: the ID of a user
- age: the age of the user
- gender: the gender of the user
- itemid: the ID of an item

- price: the price of the item
- size: the size of the item
- label: the target column, indicating whether the item is purchased. 1 indicates that the item is purchased. 0 indicates that the item is not purchased.

2. One-hot encoding

One-hot encoding converts character-type data to numeric data. In the FM-Embedding solution, one-hot encoding-1 is used to encode full data. An encoding model is created and imported to one-hot encoding-2 and one-hot encoding-3. In one-hot encoding-2, select features of the user for encoding. In one-hot encoding-3, select features of the item for encoding.

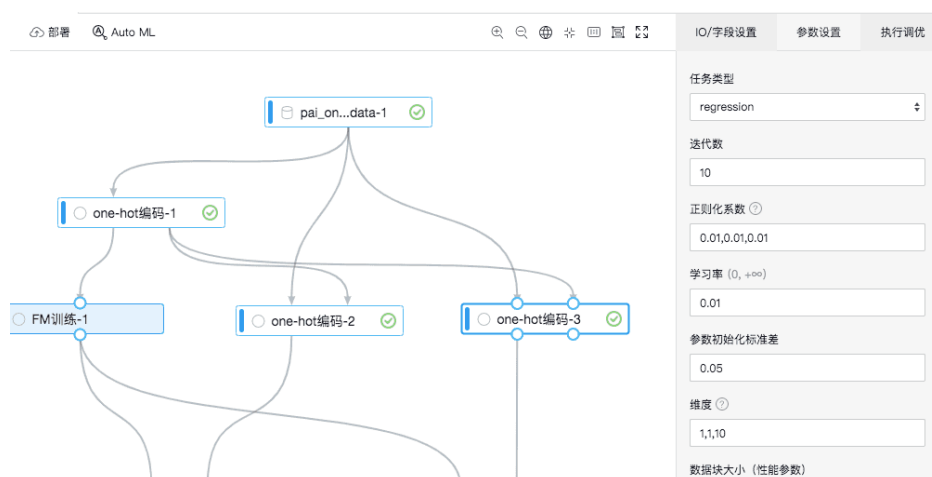
Enter userid, gender, and age in one-hot encoding-2, and select userid as the additional column.



Enter itemid, price, and size in one-hot encoding-3, and select itemid as the additional column.



3. FM training



Regularization coefficient and **Dimension** each have three parameters: constant term, monomial term, and quadratic term. The third parameter “10” of Dimension indicates the dimensions of the created Embedding node.

4. Embedding extraction



- Name of the Embedding Vector ID Column: Enter “feature_id” of the model in FM training in the left pile.
- Embedding vector column name: Enter “feature_weights” of the model in FM training in the left pile.
- Weight vector column name: Enter the sparse data column that corresponds to the right pile.
- Output result column name: Enter the name of the output Embedding field.

Final output:

userid ▲	kv ▲	embedding ▲
1	10:1,7:1,37:1	0.04015407 -0.17816195 -0.037157465 -0.06470604 -0.24434555 -0.019216094 -0.049993407 -0.06353192 -0.08150465 0.001752859 0.3356...
2	9:1,4:1,39:1	-0.067233436 -0.13599731 0.12928867 -0.014686654 -0.079268694 -0.1312892 -0.092644565 0.027404211 0.00232377 -0.109620675 0.0445...
3	10:1,4:1,40:1	-0.004508253 -0.046913035 -0.07043892 0.010427853 -0.1450108 0.021560092 -0.10439287 0.055663645 -0.08991572 -0.014267934 0.440...
4	9:1,5:1,41:1	0.0050517395 -0.0021566674 -0.07513097 -0.10988943 0.031288043 -0.0033690166 -0.08820701 0.024628945 4.7708116E-4 0.048596375 -...
5	10:1,6:1,42:1	0.043785967 0.10553776 -0.19826782 -0.041631583 -0.01759258 0.021906495 -0.03562166 0.04236281 -0.12950923 -0.13433275 0.15293656
6	9:1,8:1,43:1	-0.078507404 -0.13286367 0.075596735 -0.039212134 0.14426178 0.025733178 -0.015803259 0.0065106675 -0.024862044 -0.12671072 -0.0...
7	9:1,2:1,44:1	-0.18068565 -0.096336134 0.037038583 -0.08846839 -0.0439286 0.015447946 -0.24221739 -0.08010515 -0.008318255 -0.05676799 0.1933...
8	10:1,1:1,45:1	0.052672688 -0.004056439 -0.09321347 -0.08363886 0.0086529865 0.01378352 -0.056089412 0.002947338 0.012545784 -0.036917157 0.02...
9	9:1,0:1,46:1	-0.06683848 -0.04957156 0.101151854 0.13750216 0.019501429 -0.0941189 -0.055305757 -0.02849195 0.067301184 -0.08456889 -0.045818195
10	10:1,3:1,38:1	-0.11435607 -0.076492555 -0.21123311 0.11723561 -0.15823722 0.011994862 0.02883054 -0.06578457 -0.1195012 0.05180212 0.5513177

Summary

PAI provides the FM-Embedding solution, allowing you to quickly mine the feature vectors of a user and an item. The recall module gives a score based on the product of feature vectors of the user and item.

[Online prediction] Predict middle school students' final grades

The data in this topic is fictitious and is only used for experimental purposes.

Background

This topic uses real middle school students' data and machine mining algorithms to determine the key factors affecting middle school students' academics. The factors include parents' occupation, parents' education, and whether Internet is available at home.

This example uses a dataset that contains information about student family backgrounds and students' behavior at school. This experiment uses the logistic regression algorithm to create an offline model and an academic performance assessment report, and uses this model to predict the students' final grades. This experiment also creates an online prediction API, which allows you to apply the trained model to your online business.

Dataset

The dataset consists of 25 feature columns and 1 target column. The detailed fields are as follows.

Field	Definition	Type	Description
sex	Gender	string	F indicates female, and M indicates male.

address	Home address	string	U indicates urban, and R indicates rural.
famsize	Family size	string	LE3 indicates less than three members, and GT3 indicates more than three members.
pstatus	Living with parents or not	string	T indicates living with parents, and A indicates not living with parents.
medu	Mother' s education level	string	The value ranges from 0 to 4.
fedu	Father' s education level	string	The value ranges from 0 to 4.
mjob	Mother' s job	string	It includes education-related, health-related, and service industries.
fjob	Father' s job	string	It includes education-related, health-related, and service industries.
guardian	The student' s guardian	string	Valid values: mother, father, and other.
traveltime	The travel time from home to school	double	Unit: minutes.
studytime	The study time per week	double	Unit: hours.
failures	Failed exams	double	The number of failed exams.
schoolsup	Specifies whether additional learning aid is available	string	Valid values: yes and no.
fumsup	Specifies whether tutoring is available	string	Valid values: yes and no.
paid	Specifies whether tutoring related to examination subjects is available	string	Valid values: yes and no.
activities	Specifies whether extracurricular activity classes are available	string	Valid values: yes and no.
higher	Specifies whether the student has interest in higher	string	Valid values: yes and no.

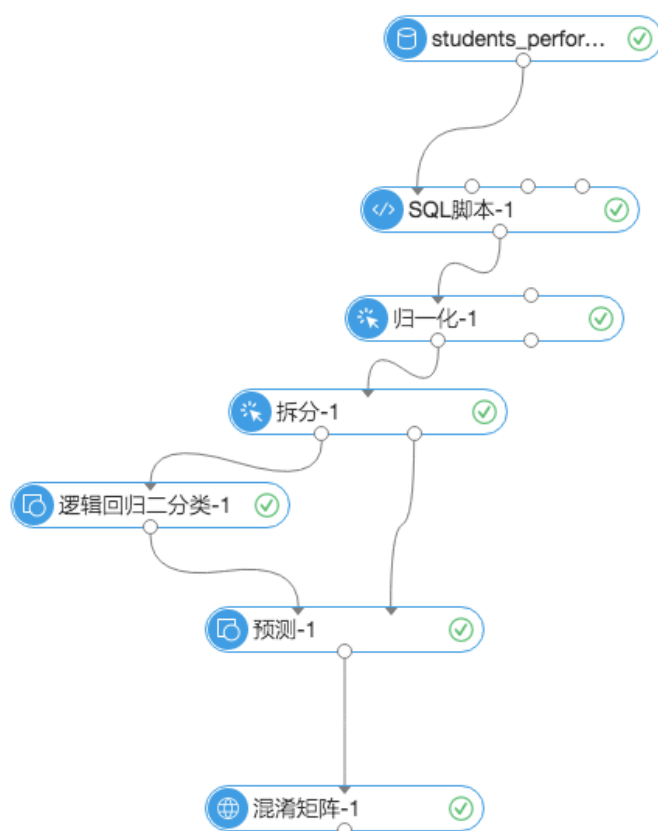
	education		
internet	Specifies whether Internet is available at home	string	Valid values: yes and no.
famrel	Family relationship	double	The value ranges from 1 to 5, indicating from bad to good family relationship.
freetime	Free time	double	The value ranges from 1 to 5, indicating from little to much free time.
goout	Frequency for going out with friends	double	The value ranges from 1 to 5, indicating from rarely to frequently going out with friends.
dalc	Daily drinking	double	The value ranges from 1 to 5, indicating from little to much drinking on a daily basis.
walc	Weekly drinking	double	The value ranges from 1 to 5, indicating from little to much drinking on a weekly basis.
health	Health status	double	The value ranges from 1 to 5, indicating from bad to good health.
absences	Absences	double	Value range: 0 to 93.
g3	Final exam	double	20-point system.

The following is a screenshot of the data.

sex ▲	address ▲	famsize ▲	pstatus ▲	medu ▲	fedu ▲	mjob ▲	fjob ▲	guardian ▲	travetime ▲	studytime ▲	failures ▲	schoolsup ▲	fumsup ▲
F	U	GT3	A	4	4	at_ho...	teacher	mother	2	2	0	yes	no
F	U	GT3	T	1	1	at_ho...	other	father	1	2	0	no	yes
F	U	LE3	T	1	1	at_ho...	other	mother	1	2	3	yes	no
F	U	GT3	T	4	2	health	services	mother	1	3	0	no	yes
F	U	GT3	T	3	3	other	other	father	1	2	0	no	yes
M	U	LE3	T	4	3	services	other	mother	1	2	0	no	yes
M	U	LE3	T	2	2	other	other	mother	1	2	0	no	no
F	U	GT3	A	4	4	other	teacher	mother	2	2	0	yes	yes
M	U	LE3	A	3	2	services	other	mother	1	2	0	no	yes
M	U	GT3	T	3	4	other	other	mother	1	2	0	no	yes
F	U	GT3	T	4	4	teacher	health	mother	1	2	0	no	yes

Offline training

The following figure shows the experiment process.



The data flows through the experiment from top to bottom, for preprocessing, splitting, training, prediction, and evaluation in sequence.

1. Data preprocessing

The SQL script is provided as follows.

```

select (case sex when 'F' then 1 else 0 end) as sex,
(case address when 'U' then 1 else 0 end) as address,
(case famsize when 'LE3' then 1 else 0 end) as famsize,
(case Pstatus when 'T' then 1 else 0 end) as Pstatus,
Medu,
Fedu,
(case Mjob when 'teacher' then 1 else 0 end) as Mjob,
(case Fjob when 'teacher' then 1 else 0 end) as Fjob,
(case guardian when 'mother' then 0 when 'father' then 1 else 2 end) as guardian,
travelttime,
studytime,
failures,
(case schoolsup when 'yes' then 1 else 0 end) as schoolsup,

```

```
(case fumsup when 'yes' then 1 else 0 end) as fumsup,
(case paid when 'yes' then 1 else 0 end) as paid,
(case activities when 'yes' then 1 else 0 end) as activities,
(case higher when 'yes' then 1 else 0 end) as higher,
(case internet when 'yes' then 1 else 0 end) as internet,
famrel,
freetime,
goout,
Dalc,
Walc,
health,
absences,
(case when G3>14 then 1 else 0 end) as finalScore
from ${t1};
```

Structure text data by using the **SQL Script** component.

- For example, the value assigned to a double type field can be Yes or No. You can use value 0 to represent Yes and value 1 to represent No.
- For some multi-value text fields, the data can be abstracted based on the scenario. For example, for the field "Mjob" , 1 can indicate a teacher and 0 can indicate a non-teacher. After abstraction, this feature indicates whether the job is related to education.
- The target column is quantified so that 1 indicates more than 18 points, and 0 indicates the others. The goal is to find a model that can predict the score through training.

2. Normalization

The purpose of the Normalization component is to remove the dimension and transform all the fields to 0 and 1. This eliminates the impact of the imbalance between the fields. The result is shown in the following figure.

sex ▲	address ▲	famsize ▲	pstatus ▲	medu ▲	fedu ▲	mjob ▲	fjob ▲	guardian ▲	traveltime ▲	studytime ▲	failures ▲	schoolsup ▲	fumsup ▲
1	1	0	0	1	1	0	1	0	0.333333333...	0.333333333...	0	1	0
1	1	0	1	0.25	0.25	0	0	0.5	0	0.333333333...	0	0	1
1	1	1	1	0.25	0.25	0	0	0	0	0.333333333...	1	1	0
1	1	0	1	1	0.5	0	0	0	0	0.666666666...	0	0	1
1	1	0	1	0.75	0.75	0	0	0.5	0	0.333333333...	0	0	1
0	1	1	1	1	0.75	0	0	0	0	0.333333333...	0	0	1
0	1	1	1	0.5	0.5	0	0	0	0	0.333333333...	0	0	0
1	1	0	0	1	1	0	1	0	0.333333333...	0.333333333...	0	1	1
0	1	1	0	0.75	0.5	0	0	0	0	0.333333333...	0	0	1
0	1	0	1	0.75	1	0	0	0	0	0.333333333...	0	0	1
1	1	0	1	1	1	1	0	0	0	0.333333333...	0	0	1
1	1	0	1	0.5	0.25	0	0	0.5	0.666666666...	0.666666666...	0	0	1
0	1	1	1	1	1	0	0	0.5	0	0	0	0	1
0	1	0	1	1	0.75	1	0	0	0.333333333...	0.333333333...	0	0	1

3. Splitting

The dataset is split in a ratio of 8:2, in which 80% is used for model training and 20% is used for prediction.

4. Logistic regression

Use the Logistic Regression component to train and create an offline model. For more information

about the algorithm, see Wiki.

5. Result analysis and evaluation

You can use the Confusion Matrix component to view the accuracy of the prediction made by your model. As shown in the following figure, the prediction accuracy of this experiment is 82.911%.

混淆矩阵

混淆矩阵 比例矩阵 统计信息

模型	正确数	错误数	总计	准确率	准确率	召回率	F1指标
0	126	25	151	82.911%	83.444%	98.438%	90.323%
1	5	2	7	82.911%	71.429%	16.667%	27.027%

According to the characteristics of the logistic regression algorithm, some valuable information can be mined through the model coefficients. Right-click the **Logistic Regression for Binary Classification** component and choose **Show Model**. The results are shown in the following figure.

逻辑回归输出

字段名	1	0
medu	2.196219307541352	-
fedu	-0.6209320272631076	-
traveltime	-0.1401344554844348	-
studytime	0.2427716427155365	-
failures	-1.41108855780472	-
famrel	0.07782683210201805	-
freetime	0.3700336237892014	-
goout	-0.6294895937934885	-

1、PAI平台提供的逻辑回归可用于多分类的，采取的策略是OneVsAll，因此在多分类的情况下，会出现多个方程，每个方程针对目标特征的某个value值，即权重（weight）下方对应的列名；

2、逻辑回归的完整公式为： $\sigma(z) = 1 / (1 + \exp(-z))$ ； $z = w_0 + w_1 * x_1 + w_2 * x_2 + \dots + w_m * x_m$ 。（其中 x_1, x_2, \dots, x_m 是某样本数据的各个特征， w_1, w_2, \dots 是特征的权重值）

关闭

According to the characteristics of the logistic regression algorithm, the greater the weight, the greater the impact of the feature on the result. A positive weight indicates a positive correlation to the result 1 (high score in final exam), and a negative weight indicates a negative correlation. Several features with large weights are analyzed in the following table.

Field	Definition	Weight	Analysis
mjob	Mother' s job	-0.7998341777833717	The mother being a teacher is disadvantageous for the child to get a high score.
fjob	Father' s job	1.422595764037065	The father being a teacher is advantageous for the child to get a high score.
internet	Specifies whether	1.070938672974736	Internet at home will

	Internet is available at home		not only have no negative impact on the score, but will also promote the child' s study.
medu	Mother' s education level	2.196219307541352	The mother' s education level has the greatest impact on the child. The higher the mother' s education level, the higher the child' s scores.

Due to the small dataset in this experiment, the preceding analysis results are not necessarily accurate and are for reference only.

Online prediction deployment

After the offline model has been created, deploy the model online and call **restful-api** to make online prediction.

References

You can log on to Alibaba Cloud Machine Learning Platform for AI (PAI) to experience this product and go to Yunqi Community to discuss with us.

[Text analysis] - Perform news classification

The data in this topic is fictitious and is only used for experimental purposes.

This experiment is intended to introduce text components. To improve the final results, please contact us. We will provide you with complete solutions and business cooperation.

Background

News classification is a common scenario in the field of text mining. At present, many media or content producers often use manual tagging for news text classification, which consumes a lot of human resources. This topic classifies news text through smart text mining algorithms. It is completely implemented by the machine without any manual tagging.

In this document, automatic news classification is implemented through the PLDA algorithm and topic weights clustering. It includes processes such as word breaking, word type conversion, deprecated word filtering, topic mining, and clustering.

Dataset

The data screenshot is shown as follows.

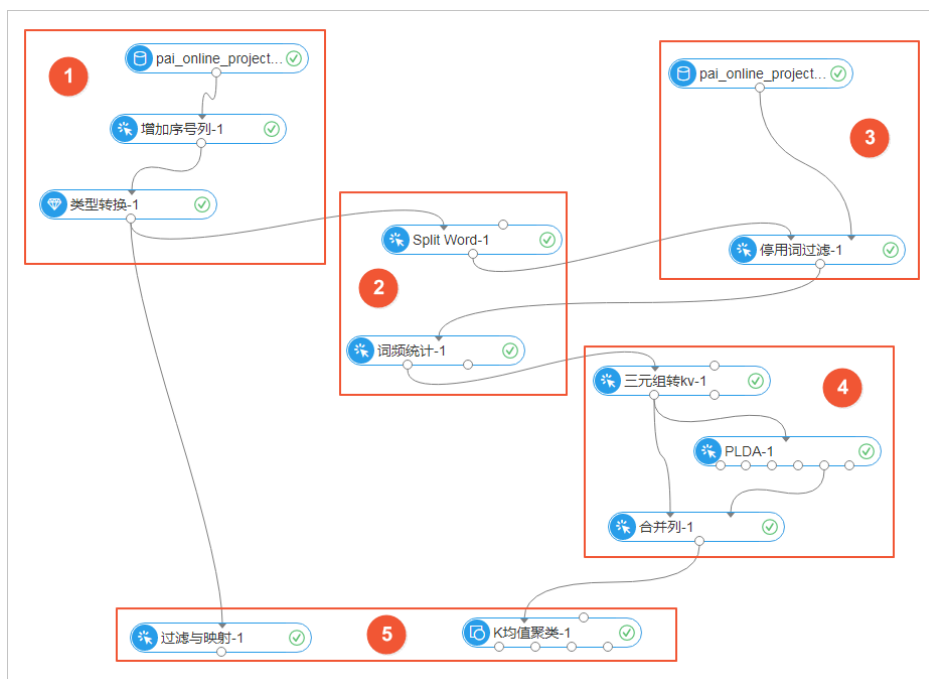


The following table describes the fields:

Field	Definition	Type	Description
category	News type	string	Sports, women, society, military, and technology.
title	Title	string	News title.
content	Content	string	News content.

Data exploration procedure

The following figure shows the experiment process.



The experiment is roughly divided into the following five steps:

- 1: Add an ID column
- 2 : Perform word breaking and word frequency analysis
- 3 : Filter deprecated words
- 4 : Mine text topics
- 5 : Analyze and evaluate results

1. Add an ID column

The data source of this experiment is based on a single news unit. It is necessary to add an ID column as a unique identifier for each news unit, which is convenient for computing the following algorithm.

2. Perform word breaking and word frequency analysis

This step is a common practice in the field of text mining.

Use the Split Word component to break the **content** field (news content). Filtered words include punctuation marks and auxiliary words. The following figure shows the result.

append_id ▲	word ▲	count ▲
0	山	1
0	分分	1
0	别墅	1
0	勇敢	1
0	包装	1
0	博爱	1
0	却	1
0	又	2
0	发	1
0	句	1

3. Filter deprecated words

Use the Deprecated Word Filtering component to filter the input deprecated-word lexicon. This typically filters punctuation and auxiliary words that have less impact on the news content.

4. Mine text topics

1. Before using the PLDA component, convert the text to a ternary form (text to numeral), as shown in the following figure.

append_id ▲	key_value ▲
213	337:1,412:1,667:3,861:1,1096:2,1582:1,1693:1,2109:1,2283:1,2371:1,2659:1,3054:3,3092:1,3232:1,4170:1,4376:1,4889:1,5206:1,5427:1,5595:1,5692:1,5739:1,6116:1,6133:1,6529:...
216	10:1,127:1,436:1,675:1,891:1,915:1,1096:2,1468:1,1757:1,2013:1,2109:1,2562:1,2783:1,3054:1,3400:1,3427:1,3443:1,3459:1,4597:1,6116:1,6183:1,6190:1,6529:1,6552:1,6871:1,7...
219	228:1,339:1,394:1,430:2,539:3,662:1,926:1,1224:1,1421:1,1488:2,1528:1,1670:2,1822:1,1909:2,2109:1,2301:1,2325:1,2411:1,2783:1,2959:1,2983:2,3209:1,4168:1,4188:1,5111:1,5...
221	10:1,18:1,200:1,387:1,412:1,436:1,450:2,472:4,555:2,563:2,637:1,639:2,667:1,813:1,856:1,913:1,1416:1,1502:1,1604:1,1636:1,2448:1,2641:2,2659:1,2929:1,3054:3,3092:2,3100:1,...
224	1582:1,3288:1,3702:1,5582:1,5932:1,6077:1,6249:1,6430:1,6529:1,6734:1,7638:1,8888:1,9418:1,9425:1,9925:1,10017:1,10176:1,11681:1,11683:1,12744:2,12748:2
227	10:1,368:1,539:1,675:1,915:1,926:1,960:1,1096:2,1423:1,1757:1,1759:1,2057:1,2109:1,2812:1,3024:1,3092:1,3181:1,3359:1,3591:1,4514:1,5464:1,6077:1,6116:1,6295:1,6529:1,65...
23	10:10,18:3,23:1,30:1,36:1,99:2,102:6,146:1,181:2,183:1,234:1,299:1,430:1,436:1,535:1,539:2,667:2,753:1,813:5,854:1,917:1,920:1,922:1,969:5,978:2,996:1,998:1,1001:4,1096:1,11...
232	12:1,13:1,18:1,69:2,146:1,200:1,234:2,328:1,370:2,565:2,571:2,605:1,608:2,667:7,813:3,891:6,1008:5,1065:1,1096:1,1104:1,1189:5,1190:2,1293:1,1572:1,1636:1,1816:1,2117:1,21...
235	12:2,13:2,18:1,88:1,204:1,478:1,523:1,558:1,575:1,606:1,667:2,670:1,754:2,803:1,872:1,921:1,1119:1,1398:2,1421:1,1498:1,1704:1,1947:1,2109:2,2132:1,2352:1,2783:3,3019:1,30...
238	10:3,202:2,539:1,667:1,892:1,1096:3,1127:1,1584:1,1806:2,2109:1,2122:1,2143:1,3024:1,3054:2,3364:1,3701:2,3765:1,3879:1,3984:1,5500:1,5685:1,6116:1,6529:1,6832:1,7460:1,...
240	10:1,107:1,115:1,148:1,412:1,430:1,450:2,586:1,667:1,800:1,931:1,1478:1,1584:1,1604:1,1652:2,1848:1,2352:1,2641:1,2676:1,2783:1,3000:2,3019:1,3054:2,3078:1,3577:1,3901:1,...

- **append_id** is the unique identifier of each news unit.
- **The number preceding the colon in the key_value field** indicates the numeral identifier that the word is abstracted into, and the colon is followed by the frequency at which the corresponding word appears.

Apply the PLDA algorithm to the data.

The PLDA algorithm is also known as topic model, which can locate words that represent the topic of each news unit. This experiment sets 50 topics. PLDA has six output piles, and the fifth output pile outputs the probability of each topic corresponding to each news unit, as shown in the following figure.

docid	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9	topic_10	topic_11	topic_12
0	0.0015625	0.0015625	0.0015625	0.0171875	0.0015625	0.0484375	0.0015625	0.0015625	0.0015625	0.0015625	0.0015625	0.0328125	0.0015625
1	0.001298...	0.014285...	0.001298...	0.014285...	0.001298...	0.001298...	0.014285...	0.001298...	0.001298...	0.014285...	0.001298...	0.001298...	0.001298...
2	0.011224...	0.021428...	0.001020...	0.011224...	0.011224...	0.001020...	0.001020...	0.001020...	0.001020...	0.011224...	0.001020...	0.021428...	0.001020...
3	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.000884...	0.0716814...	0.000884...
4	0.039285...	0.003571...	0.003571...	0.289285...	0.003571...	0.003571...	0.003571...	0.003571...	0.003571...	0.039285...	0.003571...	0.0035714...	0.075
5	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.001408...	0.043661...	0.0295774...	0.0014084...	0.11
6	0.002736...	0.010199...	0.010199...	0.000248...	0.000248...	0.040049...	0.000248...	0.000248...	0.000248...	0.000248...	0.0201492...	0.0002487...	0.000248...
7	0.000543	0.000543	0.000543	0.000543	0.000543	0.027717	0.000543	0.000543	0.000543	0.000543	0.000543	0.000543	0.000543

5. Analyze and evaluate the mining results

The preceding steps represent the news unit as a vector from the dimension of the topic.

News units can be classified by clustering the distances of the vectors. The classification results of the K-means Clustering component are shown in the following figure.

docid	cluster_index
115	0
292	0
248	0
166	0
116	2
210	3
8	4
15	4

- **cluster_index** indicates the name of each class.
- Find class 0. There are a total of 4 news units with the **docid** of 115, 292, 248, and 166.

The 4 news units 115, 292, 248, and 166 are queried through the Filtering and Mapping component. The following figure shows the result.

append_id ▲	category ▲	title ▲	content ▲
115	体育	"欧洲通...	来源：重庆晚报"欧洲通行证"考验门将每次大赛，新推出的用球都会成为球员和市场关注的焦点，此次欧洲杯的用球"欧洲通行证"估计也会让门将们大伤脑筋...
166	财经	新旗舰...	机构：周四上证指数快速击穿新低进一步摧毁了市场在3000点一带进行抵抗的信心，大盘如同自由落体，直至2900点附近才出现抵抗，最终当天再...
248	体育	图文：...	来源：体育体育讯 北京时间6月15日凌晨，08欧洲杯D组第二轮开战，在奥地利因斯布鲁克的蒂沃利球场，西班牙2比1险胜瑞典，斗牛士军团以6...
292	科技	L G第...	赛迪网讯6月30日消息，据台湾媒体报道，随着第二季度摩托罗拉在全球的手机市场的表现持续低迷，L G电子第二季度手机出货量有望突破3,000...

The experiment results are unsatisfactory. In the preceding figure, a financial news unit, a technology news unit, and two sports news units are grouped together.

The main reasons are as follows.

- There is no detailed optimization.
- There is no feature engineering for the data.
- The data volume is too small.

Mine headline news through the Online Learning solution of PAI

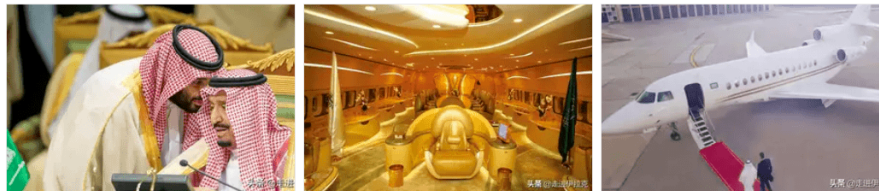
Mine headline news through the Online Learning solution of PAI

News websites often push headlines to viewers. As a news website publishes a large amount of news in real time,

it is necessary to mine headlines from the latest news. This determines the quality of news recommendations.

关注 推荐 热点 北京 视频 国风 三

沙特价值数亿美元私人飞机在机场“搁浅”，富商因害怕而不敢乘坐



走进伊拉克 14评论 50分钟前



Machine learning algorithms are required to identify potential headlines from among massive amounts of news. A conventional approach is to download historical consultations that are collected every day, train a model offline based on the collected data, and push the created headline discovery model online for use on the next day. However, this offline trained model lacks timeliness because it predicts daily headlines that are reported in real time based on historical data.

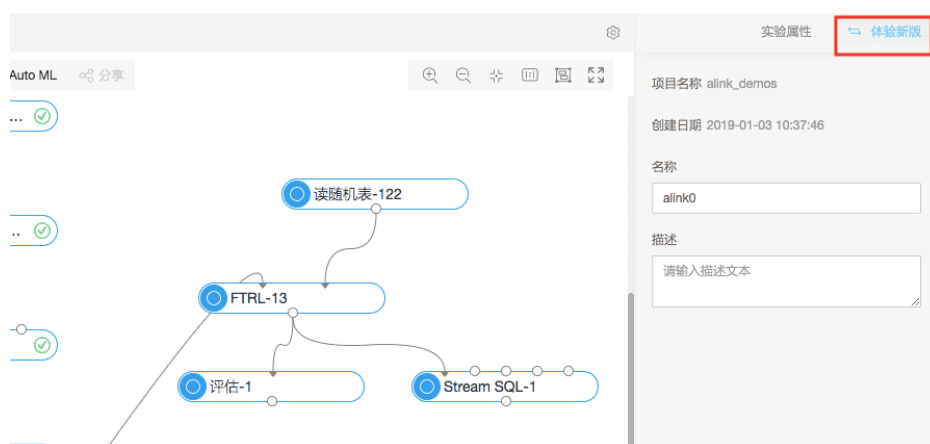
To solve this problem, Machine Learning Platform for AI (PAI) introduces the Online Learning solution that combines streaming and offline algorithms. The solution processes massive data through offline training and updates real-time models by using streaming algorithms of machine learning. This enables simultaneous running of different batches of streams. The following experiment shows how to use the Online Learning solution of PAI to mine headline news.

Experiment process

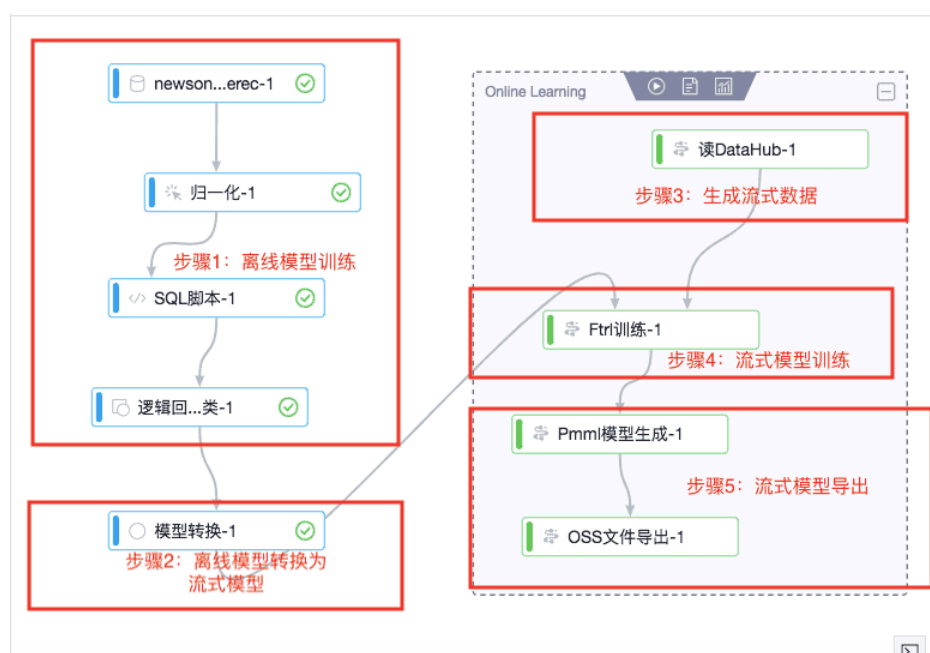
1. Activate and use the Online Learning solution

Currently, the Online Learning solution of PAI is in public review. If you want to use the solution, please fill out the questionnaire:

After activating the Online Learning solution, click **Try New Version** to start the trial.



2. Experiment process



Note: The offline computation components of PAI are marked in blue, and the stream computation components are marked in green. The stream components are interconnected to form a computing group and must all be in the running or stopped state.

Step 1: Train an offline model

This experiment uses 30,000 news items from UCI open datasets.

URL: <https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity>

The used data includes the URLs and publication time of news, 58 features, and 1 target value. The target value "share" indicates the number of news item shares. During the modeling process, use the SQL Script component to perform binary classification based on the "share" field and classify news into headlines (with more than 10,000 shares) and non-headlines (with less than 10,000 shares).

The following figure shows the feature composition.

Feature	Type (#)	Feature	Type (#)
Words		Keywords	
Number of words in the title	number (1)	Number of keywords	number (1)
Number of words in the article	number (1)	Worst keyword (min./avg./max. shares)	number (3)
Average word length	number (1)	Average keyword (min./avg./max. shares)	number (3)
Rate of non-stop words	ratio (1)	Best keyword (min./avg./max. shares)	number (3)
Rate of unique words	ratio (1)	Article category (Mashable data channel)	nominal (1)
Rate of unique non-stop words	ratio (1)	Natural Language Processing	
Links		Closeness to top 5 LDA topics	ratio (5)
Number of links	number (1)	Title subjectivity	ratio (1)
Number of Mashable article links	number (1)	Article text subjectivity score and its absolute difference to 0.5	ratio (2)
Minimum, average and maximum number of shares of Mashable links	number (3)	Title sentiment polarity	ratio (1)
Digital Media		Rate of positive and negative words	ratio (2)
Number of images	number (1)	Pos. words rate among non-neutral words	ratio (1)
Number of videos	number (1)	Neg. words rate among non-neutral words	ratio (1)
Time		Polarity of positive words (min./avg./max.)	ratio (3)
Day of the week	nominal (1)	Polarity of negative words (min./avg./max.)	ratio (3)
Published on a weekend?	bool (1)	Article text polarity score and its absolute difference to 0.5	ratio (2)
Target		Target	
		Number of article Mashable shares	number (1)

Use the logistic regression model to train and create a binary classification model, which is used to evaluate whether a piece of news will become a headline.

Note: Currently, the Online Learning solution of PAI only supports the logistic regression algorithm.

Step 2: Convert the offline model to a streaming model

Use the Model Conversion component to convert the offline logistic regression model to a streaming model that can be read by streaming algorithms.

Step 3: Generate streaming data

Step 3 and subsequent steps involve streaming algorithms. PAI provides multiple streaming data sources. This experiment uses Datahub as an example.

Datahub URL: <https://datahub.console.aliyun.com/datahub>

Datahub is a type of streaming data queue that supports multiple languages such as Java and Python. You can use Datahub to link user-created real-time data and the training service of PAI. Note: The data streams imported by Datahub must be in the same format as the fields of the data streams used for offline training so that offline models can be updated in real time.

Step 4: Train a streaming model

The Follow the Regularized Leader (FTRL) algorithm is basically equivalent to the streaming logistic regression algorithm. Set parameters based on the logistic regression algorithm. Pay attention to the Model Save Time Interval parameter, which determines the time interval at which models are created through real-time computing.

IO/字段设置

参数设置

学习率参数alpha 默认值0.1

学习率参数beta 默认值0.1





L1正则化系数 默认值0.1

L2正则化系数 默认值0.1

模型保存时间间隔 可选，默认：1800 (s)

Step 5: Export the streaming model

Export the classification model in the PMML format and write the model to Object Storage Service (OSS). The write interval is the same as the model creation interval. Model write example:

<input type="checkbox"/>	 newsRec_2019-01-10-11:00:00_0.dat	13.872KB	标准存储	2019-01-10 11:00	预览 更多 v
<input type="checkbox"/>	 newsRec_2019-01-10-11:30:00_1.dat	13.873KB	标准存储	2019-01-10 11:30	预览 更多 v
<input type="checkbox"/>	 newsRec_2019-01-10-12:00:00_0.dat	13.873KB	标准存储	2019-01-10 12:00	预览 更多 v
<input type="checkbox"/>	 newsRec_2019-01-10-12:30:00_1.dat	13.815KB	标准存储	2019-01-10 12:30	预览 更多 v

If streaming evaluation data is available, the system can store real-time model evaluation metrics together with the model in OSS.

3. Model usage

After the headline prediction model is created and stored in OSS, you can deploy the model through Elastic Algorithm Service (EAS) of PAI or download the model to be used by the local prediction engine. Perform feature engineering on incoming news data based on the instructions in “Step 1: Train an offline model.” Enter the feature engineering result in Headline Mining Service, and you can see whether the news is a potential headline.

Perform public opinion risk control based on the feedback from a takeaway platform

Background

Currently, many merchants provide online platforms for consumers to write comments and give feedback on purchased items. Consumer feedback includes praises and criticisms. Merchants need to determine whether the product quality meets consumer needs based on consumers’ opinions on products, and read consumer comments to analyze the consumer opinion trend and guide future product development.

Business pain points

At present, a large number of comments are created on the comment platforms of hotels, restaurants, and retail stores every day. The approach of manually collecting statistics on public opinion is inefficient and fails to produce accurate data on extensive public opinion. We need to devise an approach to automatically collect statistics on public opinion to determine the public opinion trend of comment platforms.

Solution

Machine Learning Platform for AI (PAI) provides a set of algorithms based on text vectorization and classification, which are used to create a classification model based on the positive (praising) and negative (critical) comments with historical flags. The created model can be used to automatically predict new comments. The overall modeling framework has been developed based on PAI by using 11,987 labeled comments collected from a takeaway comment platform. The framework implements risk control of positive and negative public opinions, with an accuracy of about 75%.

Required knowledge: basic knowledge of natural language processing (NLP) and classification algorithms, especially how such knowledge is applied to model debugging.

Development cycle: one to two days.

Required data: more than one thousand labeled data items. The prediction effect is better when more labeled data items are available.

Data

序号 ▲	label ▲	review ▲
29	1	这次的麻辣教父一点也不辣诶。。。不知道为啥。。。
30	1	真的是太好吃了太帅了吃的我美美的送餐也很快以后外卖就百度这家餐厅了
31	1	今天的牛肉烧烤饭，感觉牛肉有些不新鲜，送餐员的速度还是很快的。
32	1	大雾霾，外卖小哥记得戴口罩哦！给爸妈带回天津尝尝稻香村的的小肚~~~
33	1	"棒棒哒棒棒哒棒棒哒,师傅辛苦"
34	1	挺好的，不错
35	1	"外卖速度快,饭菜依然好吃,点赞"
36	1	饭菜很好吃
37	1	前几天点的卤肉饭要是单卖里面的泡菜就更好了
38	0	"糟糕,继续努力吧"
39	0	这个很难吃
40	0	菜明显是剩菜，跟之前买的完全不一样
41	0	煎饼不错，但等了两个小时，什么情况
42	0	晚上七点订的外卖，九点还没送到，电话说是忘了我的订单，说好的退款一直没有退还
43	0	煎饼的量太小了，，
44	0	因为楼层很多所以让人去校门口自取，好懒.....
45	0	香菇鸡肉不太好吃，果汁也太袖珍了吧....不过速度巨快

Parameter	Description
label	Label. 1 indicates a positive comment, and 0 indicates a negative comment.
review	Actual comment data.

Procedure

Log on to PAI Studio at <https://pai.data.aliyun.com/console>

The solution data and experiment environment are built in the corresponding template on the homepage .

基于外卖评论的舆情风控



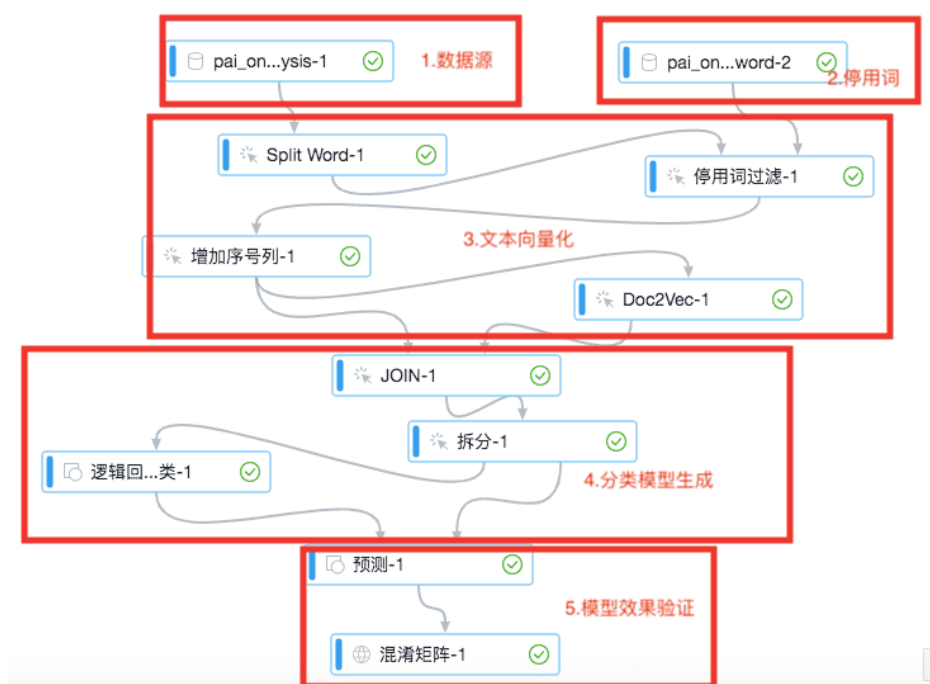
利用NLP算法分析外卖评论，判断用户的正负情感

0 位用户

从模版创建

查看文档

Open the experiment:



1. Data source

The data source is the comments described in the preceding sections.

1. Deprecated words

Manually upload the deprecated word table to filter auxiliary verbs and punctuation marks.

序号 ▲	stop_word ▲
9	7
10	8
11	9
12	?
13	_
14	“
15	”
16	、
17	。
18	《
19	》

1. Text vectorization

Use the Doc2vector algorithm to convert each comment to a semantic vector. Each line includes a vector, and each vector represents the meaning of a comment.

f0 ▲	f1 ▲	f2 ▲	f3 ▲	f4 ▲	f5 ▲	f6 ▲	f7 ▲	f8 ▲	f9 ▲	f10 ▲	f11 ▲	f12 ▲	f13 ▲	f14 ▲	f15 ▲	f16 ▲	f17 ▲
0.0...	-0.03756009414792061	0.012046359...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	0.00...	-0.0...	0.06...	0.02...	-0.0...	-0.0...	0.00...	0.00...
0.0...	-0.015270709991455079	0.008787018...	-0...	-0...	0.0...	-0...	-0...	-0...	0.0...	-0.0...	-0.0...	0.03...	0.00...	-0.0...	0.00...	0.00...	0.00...
0.0...	-0.02618148736655712	0.003506168...	-0...	0.0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	-0.0...	0.04...	-0.0...	-0.0...	-0.0...	-0.0...	0.00...
0.0...	-0.016501447185873985	-0.00243335...	-0...	0.0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	0.00...	0.01...	-0.0...	-0.0...	-0.0...	-0.0...	0.00...
0.0...	-0.008959046564996243	0.0065372115...	-0...	-0...	0.0...	-0...	-0...	-0...	-0...	-0.0...	-0.0...	0.00...	0.00...	-0.0...	-0.0...	0.00...	0.00...
0.0...	-0.008599202148616314	0.0009298113...	-0...	-0...	0.0...	0.0...	0.0...	-0...	0.0...	0.00...	-0.0...	0.01...	0.00...	-0.0...	-0.0...	-0.0...	-0.0...
0.0...	-0.020256049931049347	0.0144845861...	-0...	-0...	0.0...	-0...	0.0...	-0...	-0...	-0.0...	0.02...	0.01...	-0.0...	-0.0...	0.01...	-0.0...	0.01...
0.0...	-0.010314139537513256	0.004535630...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	-0.0...	0.01...	0.00...	-0.0...	-0.0...	0.00...	0.00...
0.0...	-0.04055945202708244	0.028456654...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	-0.0...	0.03...	0.01...	-0.0...	0.00...	-0.0...	0.02...
0.0...	-0.015246668830513954	-0.00390523...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	-0.0...	0.02...	0.00...	-0.0...	0.00...	-0.0...	-0.0...
0.0...	-0.0409286096920853	0.0108231166...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	0.00...	0.04...	0.00...	-0.0...	0.01...	-0.0...	0.01...
0.0...	-0.009084475226700306	-0.00076219...	-0...	0.0...	0.0...	-0...	0.0...	-0...	0.0...	0.00...	0.00...	0.00...	0.00...	-0.0...	-0.0...	0.00...	0.00...
0.0...	-0.0124673992395401	-0.00044502...	-0...	0.0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	0.00...	0.00...	-0.0...	-0.0...	-0.0...	-0.0...	-0.0...
0.0...	-0.05390368402004242	0.0249195415...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	-0.0...	0.07...	0.02...	-0.0...	-0.0...	0.00...	0.01...
-0...	-0.0014472039163601737	-0.00728650...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	0.01...	-0.0...	-0.0...	0.00...	0.00...	-0.0...	0.00...
0.0...	-0.047522492706775665	0.012986063...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	-0.0...	0.00...	0.04...	0.00...	-0.0...	0.00...	-0.0...	0.01...
0.0...	-0.03107563406229019	0.012634775...	-0...	-0...	0.0...	-0...	0.0...	-0...	0.0...	0.00...	-0.0...	0.04...	-0.0...	-0.0...	0.00...	-0.0...	0.00...

1. Create a classification model

Use the splitting algorithm to split vectorized text into the training set and test set. Train the training set by using the logistic regression algorithm to create a binary classification model. This model can be used to determine whether a comment is a positive or negative comment.

1. Model effect verification

Use the confusion matrix algorithm to verify the actual effect of the model.



模型	正确数	错误数	总计	准确率	精确率	召回率	F1指数
0	2089	705	2774	71.166%	74.585%	86.208%	79.977%
1	488	331	819	71.166%	59.585%	40.905%	48.509%

Summary

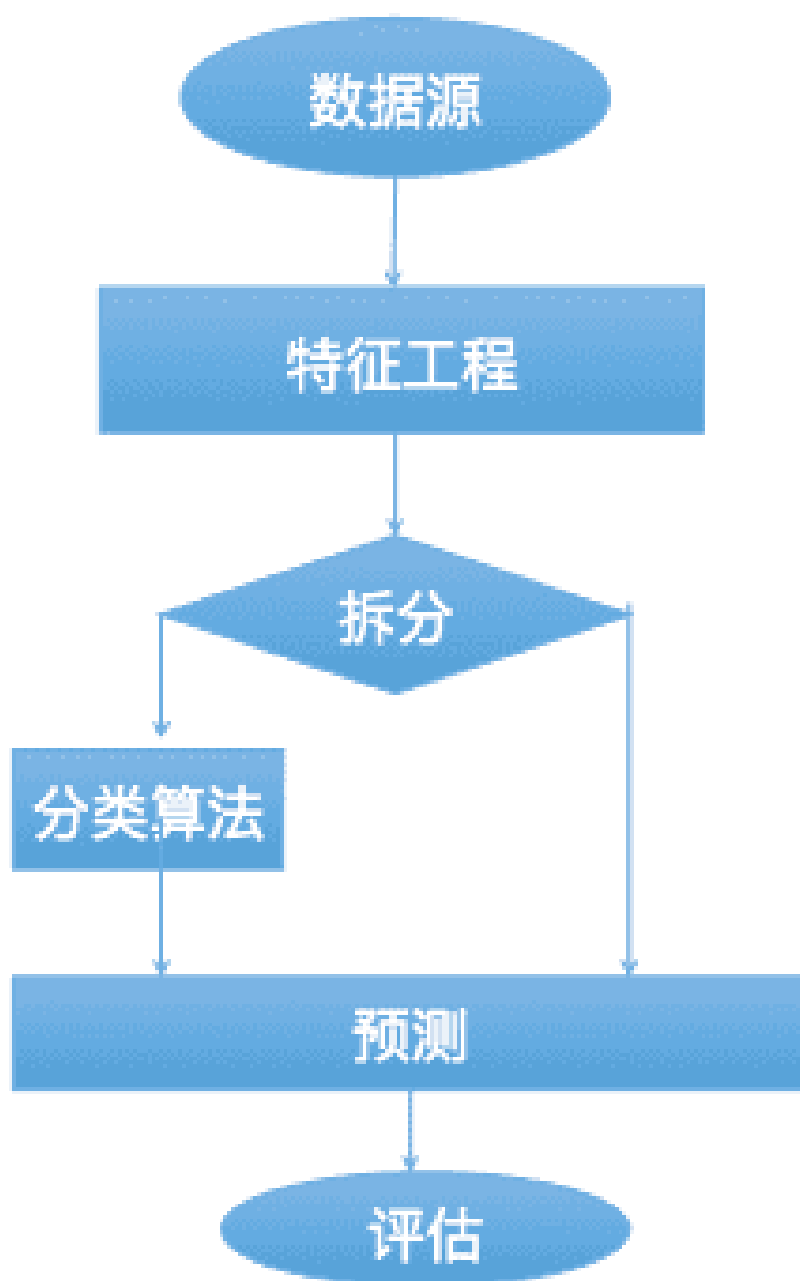
The public opinion risk control approach based on comments analysis can be developed in one to two days through PAI. The approach can intelligently analyze comments in batches. The accuracy of the model is improved with the increase in comments. This approach is applicable to textual analysis, such as spam classification and classification of positive and negative public opinions on news.

Perform recommendation based on features of recommendation targets and objects

This experiment uses data from real-life e-commerce scenarios that has been anonymized. The data is only used for learning and shall not be used for commercial purposes.

The previous issue describes how to use Machine Learning Platform for AI (PAI) to build a recommendation system based on collaborative filtering. This topic describes recommendation methods based on the features of recommendation objects and targets.

The following figure shows the general flowchart of recommendation based on object features.



- Import the supervised, structured data to MaxCompute.
- Perform feature engineering, including data preprocessing and feature derivation. Feature derivation aims to expand data dimensions so that data can reflect business features to the maximum extent.
- Split the data into two parts. One part is used to create a binary classification model by using the classification algorithm. The other part is used to test the model effect.
- Determine the model effect by using the evaluation component.

1. Business scenario

Create a prediction model by training the April and May data of a real-life e-commerce scenario.

Evaluate the prediction model based on the shopping statistics in June to determine the optimal model. Deploy the optimal model as an online HTTP service to be called in business scenarios.

This experiment is conducted in PAI Studio to build a recommendation system based on object features simply by dragging and dropping components. The data and complete business flow in this experiment are built in the corresponding template on the homepage. The template is ready for use.



2. Dataset

This experiment uses data provided by Tianchi Competition, including the shopping behavior statistics before July and the data since July. The fields are as follows.

Field	Definition	Type	Description
user_id	User ID	string	The ID of a buyer.

item_id	Item ID	string	The ID of the purchased item.
active_type	Shopping behavior	string	0: Click; 1: Buy; 2: Add to Favorites; 3: Add to Shopping Cart.
active_date	The time of shopping	string	The time when the shopping occurs.

Data entries:

10944750	8689	2	5月2日
10944750	25687	2	5月8日
10944750	7150	1	6月7日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日
10944750	13451	0	6月4日

3. Data exploring

This experiment is conducted in PAI Studio. It allows you to build a recommendation system simply by dragging and dropping components based on collaborative filtering. PAI Studio supports automatic parameter tuning and one-click model deployment.

The following figure shows the experiment flowchart.



(1) Feature engineering

Perform feature engineering to expand the dimensions of the raw data with only four fields. The recommendation scenario includes two types of features: the features of the targets to which items are recommended and the features of the items that are recommended.

In the case of item recommendation:

- The recommendation object is an item. The expanded dimensions include the number of purchases of this item, the number of clicks on this item, and the purchase-to-click ratio of this item, which is calculated by dividing the purchase quantity by the click quantity.
- The recommendation target is a user. The expanded dimensions include the total number of purchases made by this user, the total number of clicks by this user, and the purchase-to-click ratio of this user, which is calculated by dividing the click quantity by the purchase quantity. The purchase-to-click ratio indicates the number of times that the user clicks before buying an item. It describes the user's purchase intention.

The data is expanded from 4 fields to 10 fields.

user_id ▲	item_id ▲	active_type ▲	active_month ▲
10944750	13451	0	6
10944750	13451	2	6
10944750	13451	2	6
10944750	13451	0	6
10944750	13451	0	6
10944750	13451	0	6
10944750	13451	0	6

item_id ▲	user_id ▲	active_type ▲	active_month ▲	item_total_buy ▲	item_total_count ▲	item_buy_rate ▲	user_total_count ▲	user_total_buy_count ▲	user_buy_rate ▲
1000	12016750	0	5	1	4	0.25	221	18	0.08144796380090498
1000	12016750	0	5	1	4	0.25	221	18	0.08144796380090498
1000	12016750	0	5	1	4	0.25	221	18	0.08144796380090498
1000	12016750	0	5	1	4	0.25	221	18	0.08144796380090498
10000	5901250	0	6	0	2	0	50	0	0
10000	5901250	0	6	0	2	0	50	0	0
10000	5901250	0	6	0	2	0	50	0	0
10000	5901250	0	6	0	2	0	50	0	0
10010	2921750	0	5	0	2	0	528	11	0.02083333333333332

(2) Model training

Feature engineering produces a large wide table with structured data, which can be used for model training. This experiment uses the logistic regression algorithm. Model training requires proper parameter setting. It is necessary to properly set the following logistic regression parameters for optimal effect of model training.

正则项 可选

None

最大迭代次数 可选

100

正则系数 可选 正则类型为None时此值无效

1

最小收敛误差

0.000001

PAI provides the AutoML engine for parameter tuning. Open AutoML and set the parameter value range and evaluation criteria of the algorithm that requires parameter tuning. Then, the engine finds the most suitable parameter settings with minimum resource consumption. See the following figure.



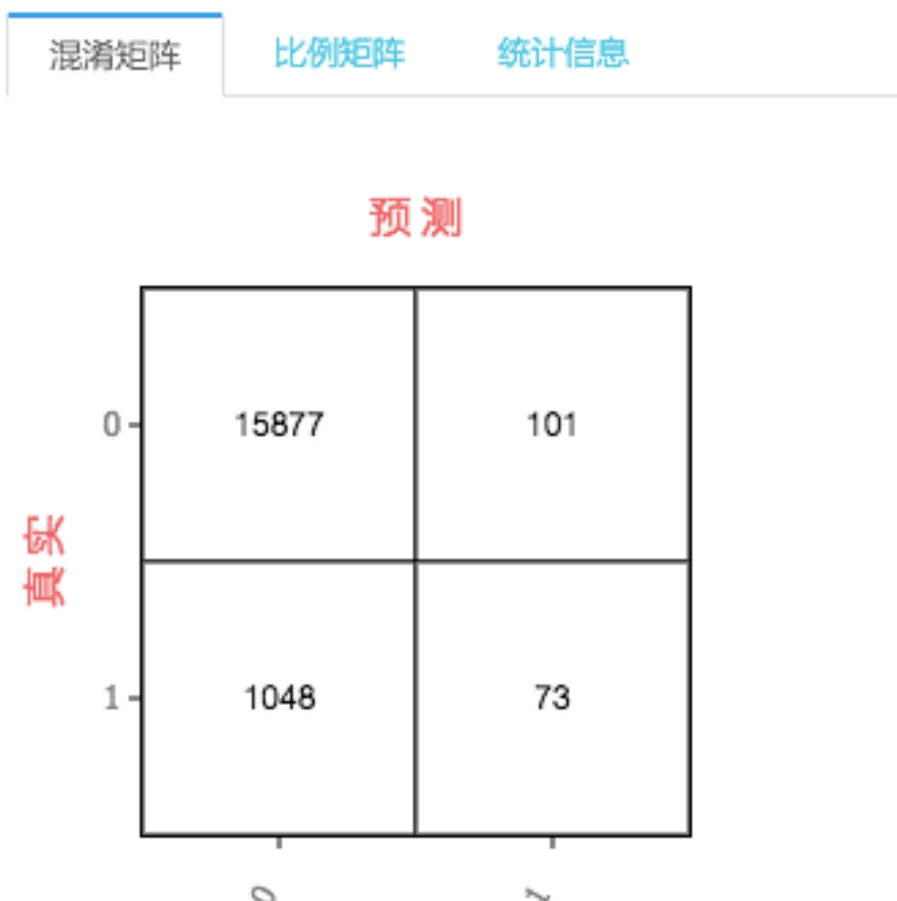
(3) Model evaluation

The model evaluation module uses the reserved data that is not used for model training to evaluate the model quality. The experiment on recommendation involves binary classification. You can use the confusion matrix and the Binary Classification Evaluation component to evaluate the model quality.

Binary classification evaluation: Choose **Components** and click the **Charts** tab. The ROC curve shown in the following figure appears. The blue area indicates the AUC value. The larger the area, the higher the model quality.



The confusion matrix can be used to determine specific metrics such as the prediction accuracy, recall rate, and F1-Score.



(4) Online model deployment

If the model effect meets expectation, deploy the model as an online service in one click through Elastic Algorithm Service (EAS) of PAI. Then, the model can be accessed over HTTP. On the canvas, click **Deploy**, select **Deploy Model Online**, and select the target model.

After the model is deployed as an online service, it can be accessed through HTTP requests in business scenarios. This streamlines the process from model training through PAI to business application.

Perform risk control on abnormal behaviors of a system

Background

A user system may encounter abnormal metrics when the CPU utilization of the O&M system increases suddenly, the user system is flooded with illegal information, or some users frequently make bargain speculation. The user system may be far less exposed to risks if we can take preventive measures and implement real-time warning for abnormal metrics through Machine Learning Platform for AI (PAI).

Business pain points

Real-time and effective measures are unavailable to monitor the metrics of user systems and improve the intelligent defense capability of user systems.

Solution

PAI provides a set of classification algorithms based on metric monitoring. These algorithms are used to abstract abnormal metric monitoring into a binary classification scenario and deploy the monitoring model to an online system for real-time calling. This helps implement near-line risk control.

Required knowledge: knowledge of the classic algorithms in machine learning, especially feature engineering and binary classification algorithms.

Development cycle: one to two days.

Required data: one thousand labeled data items, including abnormal data and normal data.

Data

The following experiment uses a system-level monitoring log with 22,544 data items, of which 9,711 are abnormal data items.

service <small>▲</small>	flage <small>▲</small>	a2 <small>▲</small>	a3 <small>▲</small>	a4 <small>▲</small>	a5 <small>▲</small>	a6 <small>▲</small>	a7 <small>▲</small>	a8 <small>▲</small>	a9 <small>▲</small>	a10 <small>▲</small>	a11 <small>▲</small>	a12 <small>▲</small>	a13 <small>▲</small>	a14 <small>▲</small>	a15 <small>▲</small>	a16 <small>▲</small>	a17 <small>▲</small>	a18 <small>▲</small>	a19 <small>▲</small>	a2
private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13
ftp_data	SF	12...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
http	SF	267	14...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4
smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
telnet	SF	129	174	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
http	SF	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	33
ftp	SF	26	157	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
telnet	SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111
telnet	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12
telnet	SF	773	36...	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
http	SF	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8
http	SF	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	24

Data:

Parameter	Description
protocol_type	The protocol used for network connection. Valid values: TCP, ICMP, and UDP.
service	The service protocol. Valid values: HTTP, Finger, POP, Private, and SMTP.
flage	Valid values: SF, RSTO, and REJ.
a2-a38	Different system metrics.
class	The label field. “normal” indicates a normal sample, and “anomaly” indicates an abnormal sample.

Procedure

Log on to PAI Studioat <https://pai.data.aliyun.com/console>

The solution data and experiment environment are built in the corresponding template on the homepage.

异常行为风控



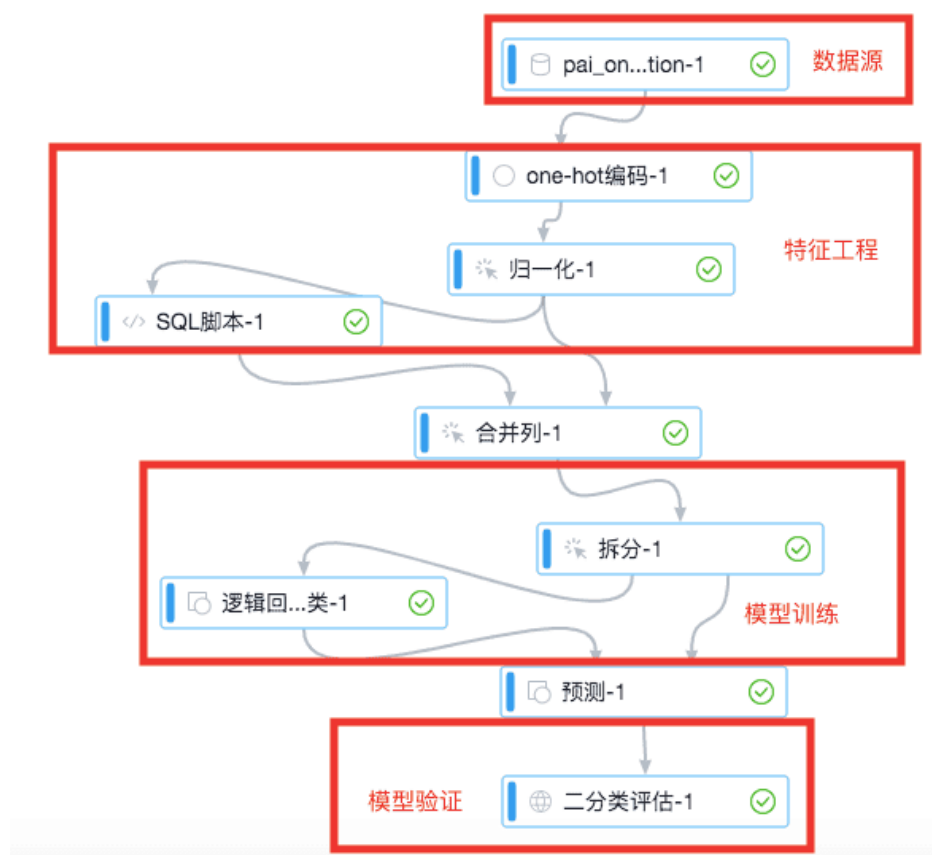
通过算法判别系统中的异常行为

0 位用户

从模版创建

查看文档

Open the experiment:



1. Data source

The data source is the data described in the “Data” section.

2. Feature engineering

The One-Hot Encoding component converts character-type features to the numeric type. This is the most common mode of data encoding in machine learning.

The Normalization component limits all data within the range of 0 to 1, without the impact of dimensions. The following figure shows the normalized data.

a1 ▲	a2 ▲	a3 ▲	a4 ▲	a5 ▲	a6 ▲	a7 ▲	a8 ▲	a9 ▲	a10 ▲	a11 ▲
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0.00003...	0.00020...	0	0	0	0	0	0	0	0	0
0	3.183413...	0	0	0	0	0	0	0	0	0
0.00001...	0	0.0...	0	0	0	0	0	0	0	0
0	0.00000...	0.0...	0	0	0	0	0	1	0	0
0	0.00001...	0.0...	0	0	0	0	0	1	0	0
0	0.00000...	0.0...	0	0	0	0	0.25	0	0	0
0	0.00000...	0.0...	0	0	0	0	0	1	0	0
0	4.138437...	0.0...	0	0	0	0	0.25	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0.00000...	0.0...	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0.00064...	0.00001...	0.2...	0	0	0	0	0	1	0	0
0	0.00000...	0.0...	0	0	0	0	0	1	0	0
0	0.00000...	0.0...	0	0	0	0	0	1	0	0

Use the SQL Script component to mark metric labeled “anomaly” as 1 and those labeled “normal” metrics as 0 in the target column.

```
select (case class when 'anomaly' then 1 else 0 end) as class from ${t1};
```

3. Model training

The binary logistic regression algorithm of logistic regression in machine learning is effective in training a monitoring model based on normal and abnormal samples.

模型描述		×
模型名称	逻辑回归二分类-1-Model	
ODPS 模型名称	pai_online_project/pai_model_1664081855183111/partition_1146807/xlab_m_logisticregres_1146807_v0.xml	
对应节点名称	逻辑回归二分类-1	
算法来源	逻辑回归二分类	
特征	a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20,a2...	
目标列	class	
参数	epsilon: 0.000001 enableSparse: false regularizedLevel: 1 maxIter: 100 kvDelimiter: : _label#labelColName: class@bigint itemDelimiter: , regularizedType: None	
创建时间	2019-12-04 12:14:11	
更新时间	2019-12-04 12:14:11	
		关闭

4. Model evaluation

PAI provides the Binary Classification Evaluation component to evaluate the model effect based on metrics such as AUC, KS, and F1Score. The model used by this experiment reaches a prediction accuracy of more than 90%.

评估报告

<

Summary

PAI provides comprehensive functions such as feature encoding, model training, and model evaluation, allowing you to create a metric anomaly monitoring model by extracting and labeling the features of abnormal behaviors of the target system.

Use ALS to predict ratings of songs

Use ALS to predict ratings of songs

Many people will visit a movie recommendation website to check the rating of a movie before they watch it. After they watch a movie, they will also assign a rating to the movie. Everyone has a rating system in their mind. The rating of a commodity, song, or movie reflects whether the user likes or dislikes it. If a content provider can estimate the ratings to be assigned by its users, it can understand its users in a better way and then make more precise recommendations. This topic describes how to use Alternating Least Square (ALS), a factorization algorithm, to predict the ratings of a song or movie assigned by users.

ALS introduction

ALS is a model-based recommendation algorithm. It factorizes models through sparse matrix factorization, and predicts the values of missing entries. In this way, a basic model is trained. The model is then used to make predictions based on new user and item data. ALS uses the alternating least squares method to calculate missing entries. The alternating least squares method is developed based on the least squares method.

ALS is a type of user-item based collaborative filtering, also known as hybrid collaborative filtering.

In this topic, we use music rating as an example to introduce how ALS works. The source dataset, Matrix A, contains the ratings of songs assigned by all listeners. The ratings may be sparse because not every listener has listened to all the songs in the library and not all the songs are rated by every listener.

	痴心绝对	小酒窝	红豆	明天你好	浮夸
听众1	5			4	
听众2		6			3
听众3	3		7		
听众4				4	
听众5		4			6

ALS factorizes Matrix A to the product of the transposes of Matrix X and Matrix Y.

$$\text{Matrix A} = \text{Transpose of Matrix X} \times \text{Transpose of Matrix Y}$$

The columns in Matrix X and rows in Matrix Y are known as factors in ALS. These factors have implicit definitions. Matrix X and Matrix Y contain three factors: personality, education level, and interests. Matrix X and Matrix Y factorized from Matrix A are expressed as follows.

	性格	教育程度	兴趣爱好
听众1	X_{11}	X_{12}	X_{13}
听众2	X_{21}	X_{22}	X_{23}
听众3	X_{31}	X_{32}	X_{33}
听众4	X_{41}	X_{42}	X_{43}
听众5	X_{51}	X_{52}	X_{53}

(Matrix X)

	痴心绝对	小酒窝	红豆	明天你好	浮夸
性格	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
教育程度	Y_{21}	Y_{22}	Y_{23}	Y_{24}	Y_{25}
兴趣爱好	Y_{31}	Y_{32}	Y_{33}	Y_{34}	Y_{35}

(Matrix Y)

Based on the factorized data, rating predictions can be easily made. For example, Listener 6 has never listened to the song Red Bean but we have obtained the Vector M of Listener 6 from Matrix X. To predict the rating of Red Bean by Listener 6, we only need to multiply Vector M of Listener 6 by Vector M of Red Bean in Matrix Y.

Use ALS in Alibaba Cloud Machine Learning Platform for AI (PAI)

Now we create an experiment in Alibaba Cloud PAI based on the preceding ALS use case. The experiment consists of the input data and ALS components. You can find the template of this use case on the Home page of PAI Studio.

ALS实现音乐推荐



利用ALS实现音乐、电影相关的内容推荐

2 位用户

从模版创建

查看文档

The following figure shows the created experiment.



1. Data source

The input data contains the following fields.

id ▲	<u>user</u> ▲	score ▲	item ▲
5	3249	1	978245916
5	3176	2	978243085
5	1719	3	978244205
5	2806	2	978243085
5	2734	2	978242788
5	1649	4	978244667
5	321	3	978245863

- user: user ID.
- item: song ID.
- score: the rating of the song assigned by the relevant user.

2. ALS matrix factorization

You must specify the fields as shown in the following figure.

字段设置	参数设置	执行调优
user列名		
<input type="text" value="user"/> ✕ 📁		
item列名		
<input type="text" value="item"/> 📁		
打分类名		
<input type="text" value="score"/> 📁		

Parameter	Description	Valid value	Required or not and default value
userColName	The name of the user column.	The column type must be bigint. The entries do not need to be continuously numbered.	Required.
itemColName	The name of the item column.	The column type must be bigint. The entries do not need to be continuously numbered.	Required.
rateColName	The name of the score column.	The column type must be numeric.	Required.
numFactors	The number of factors.	Positive integer.	Optional. Default value: 100.
numIter	The number of iterations.	Positive integer.	Optional. Default value: 10.
lambda	Regularization coefficient.	Floating point.	Optional. Default value: 0.1.
implicitPref	Specifies whether the implicit preference model is used.	Boolean.	Optional. Default value: false.
alpha	Implicit preference	Floating point larger	Optional. Default

	coefficient.	than 0.	value: 40.
--	--------------	---------	------------

3. Prediction result analysis

In this experiment, two tables are output, which correspond to Matrix X and Matrix Y described in the ALS introduction.

The Matrix X table is as follows.

user ▲	factors ▲
1	[−0.14220297,0.8327106,0.5352268,0.6336995,1.2326205,0.7112976,0.9794858,0.8489773,0.330319,0.7426911]
2	[0.7714355,0.8170629,0.14070371,0.78157544,0.40145266,0.22435305,0.5998539,0.87861717,0.9321072,0.60098845]
3	[0.06963833,0.37125903,0.66982716,0.2325376,0.036257666,0.58954036,0.65054536,0.024004433,0.0033932994,0.57789034]
4	[0.64207155,0.8115232,0.32260254,0.3855561,0.25163174,0.40492404,0.5162408,0.3814767,0.67290497,0.50865084]
5	[0.517571,0.48458508,0.098304495,0.16832124,0.9891444,0.6789138,1.0585984,0.92578393,0.81489587,0.69474304]
6	[0.86565155,0.52865344,0.51986974,0.39816418,0.5968873,0.31424767,0.74578124,0.6733258,0.55831975,0.5425565]
7	[0.4147453,−0.27837437,0.4839715,0.7758234,0.6311068,0.84274673,0.4438908,0.8602465,0.3978993,1.4290581]
10	[0.47920293,0.91401875,0.95837015,0.7224187,0.5349992,0.7437093,0.33653644,1.0294899,0.4823215,0.41025826]
11	[0.54607016,0.23469958,0.32390735,0.5483177,0.07322444,0.87607765,0.25690663,0.75714564,0.19066288,0.2303486]

The Matrix Y table is as follows.

item ▲	factors ▲
1009669227	[0.3043724,0.9211403,0.9649405,1.0043586,0.2320434,0.21626948,0.54844594,−0.3672228,0.09937295,0.9076632]
1009669181	[0.7098306,1.0229378,0.39896926,0.21804416,0.59587604,0.9355453,0.41798923,0.3523143,0.6874485,0.6521343]
1009669116	[0.3661423,0.3652928,0.8348509,0.9079304,0.7299789,0.2659982,0.26861745,0.65150297,0.6419628,1.2271518]
1009669115	[1.1625334,0.48568162,0.6818684,0.6328848,0.356604,−0.14263554,0.30305552,0.88706565,0.42701712,0.07457363]
1009669071	[0.39142805,0.06098657,0.3756292,1.0510693,0.42343494,0.86710936,0.4328914,0.09838692,−0.034022175,0.4868143]
994556636	[0.71699333,0.5847747,0.96564907,0.36637592,0.77271074,0.52454436,0.69028413,0.2341857,0.73444265,0.8352135]
994556598	[0.5234192,0.40755722,0.55578834,0.4585709,0.55235267,0.73103094,0.40249807,0.30472404,0.5356546,0.63388145]
993707035	[0.13577692,0.31378198,0.23644955,0.060735635,−0.083099656,0.16841954,0.1623567,0.21238364,0.18928273,0.123004556]
993707016	[0.1835768,0.74266636,0.49669686,0.2840153,0.8125185,0.36599895,0.31735852,0.31228343,0.9716536,0.11837222]
993706986	[0.171457,0.7812586,0.36249438,0.24480419,0.68455917,0.079008356,0.6320103,0.60387015,0.280187,0.38793203]

To predict the rating of item 994556636 made by user1, you only need to multiply the following vectors together.

- User1: [−0.14220297,0.8327106,0.5352268,0.6336995,1.2326205,0.7112976,0.9794858,0.8489773,0.330319,0.7426911]
- item994556636: [0.71699333,0.5847747,0.96564907,0.36637592,0.77271074,0.52454436,0.69028413,0.2341857,0.73444265,0.8352135]

Monitor user loss

Background

How to increase the user base while retaining existing users is key to business growth. Many technical measures are required to retain existing users. An important measure is to create a user loss model to learn the features of lost users in the past and train a risk control model through machine learning to predict the user loss trend. This helps formulate measures to prevent user loss.

Business pain points

Many businesses take warning and monitoring measures to prevent user loss, but these measures are not intelligent enough. Rule-based warning is widely used but fails to discover potential user loss in an accurate manner.

Solution

Machine Learning Platform for AI (PAI) provides a set of solutions for feature encoding, classification model training, and model evaluation based on labeled data.

Required knowledge: basic modeling knowledge.

Development cycle: one to two days.

Required data: more than one thousand labeled data items that indicate the situations under which users are lost. The prediction effect is better when more labeled data items are available.

Data

In this example, data is collected on the behaviors of 7,043 user samples in the real-life telecommunications field. The collected data includes the user attributes and the status of user loss (whether users are lost or retained).

数据探查 - pai_online_project.telco_customer_churn - (仅显示前一百条)

customerId	gender	seniorcitizen	partner	dependents	tenure	phonservice	multiplelines	internetservice	onlinesecurity	onlinebackup	deviceprotection	techsupport
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No
5575-GNVEE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No
3666-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No
9305-CDKIC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No
1452-KOQVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No
6713-KOQMC	Female	0	No	No	10	No	No phone service	DSL	Yes	No	No	No
7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes
6388-TABOU	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No
9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No
7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service	No internet ser...	No internet service	No internet
8091-TTVAX	Male	0	Yes	No	68	Yes	Yes	Fiber optic	No	No	Yes	No
0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No
5129-JLPS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes
3655-SNQYZ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes
9191-XWSZG	Female	0	No	No	52	Yes	No	No	No internet service	No internet ser...	No internet service	No internet
9959-WORKT	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No
4190-MFLUW	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes
4183-MYFRB	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No

数据探查 - pai_online_project.telco_customer_churn - (仅显示前一百条)

Feature data:

Parameter	Description
customerid	The ID of a user.
gender	The gender of the user.
SeniorCitizen	Specifies whether the user is a city resident. 1 indicates that the user is a city resident, and 0 indicates that the user is not a city resident.
Partner	Specifies whether the user has a partner.
Dependents	Specifies whether the user is affiliated.
tenure	The duration when the user has dealings with the company.
PhoneService	Specifies whether the user subscribes to mobile phone services.
MultipleLine	Specifies whether the user has multiple lines.
InternetService	Specifies whether the user subscribes to services from Internet service providers (ISPs). Valid values include DSL, Fiber optic, and No.
OnlineSecurity	Specifies whether the user faces Internet security issues.
OnlineBackup	Specifies whether the user has access to online support.
DeviceProtection	Specifies whether the user has access to service protection.
TechSupport	Specifies whether the user has applied for technical support.
StreamingTV	Specifies whether the user has access to streaming TV programs.
StreamingMovies	Specifies whether the user has access to streaming movies.
Contract	The time limit of the user' s contract. Values: Month-to-month and Two year.
PaperlessBilling	Specifies whether the user receives electronic bills.
PaymentMethod	The payment method used by the user.
MonthlyCharges	The monthly expenses of the user.
TotalCharges	The total expenses of the user.

Target data:

Parameter	Description
-----------	-------------

churn	Specifies whether the user is lost.
-------	-------------------------------------

Procedure

Log on to PAI Studio at <https://pai.data.aliyun.com/console>

The solution data and experiment environment are built in the corresponding template on the homepage .

流失用户监控



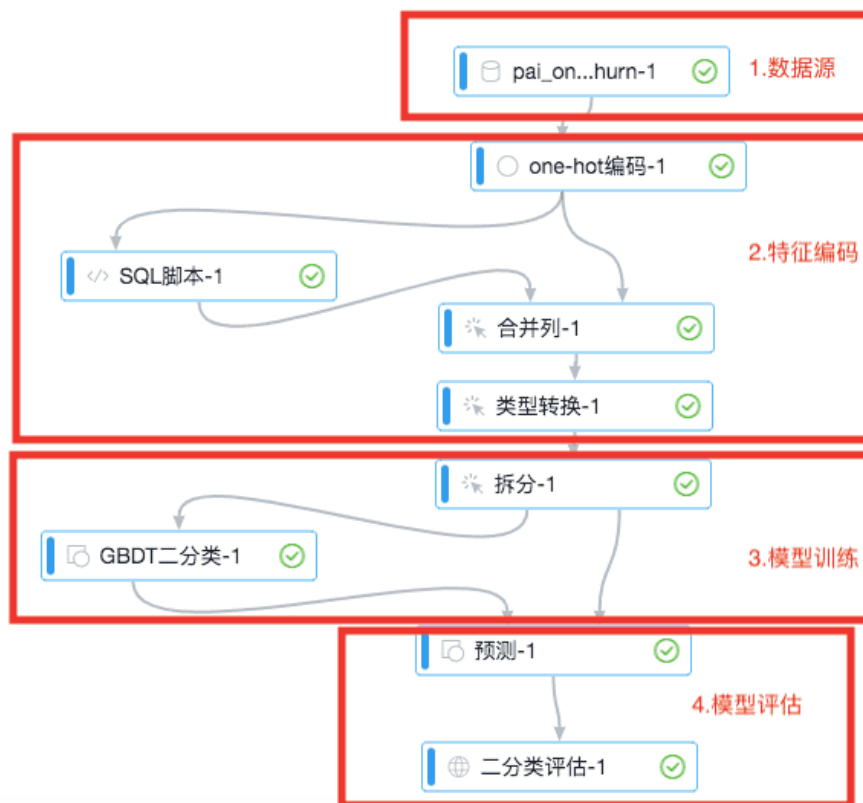
通过算法挖掘潜在可能流失的用户

0 位用户

从模版创建

查看文档

Open the experiment:



1. Data source

The data source is the streaming data received by users.

1. Feature encoding

Use the One-Hot Encoding and SQL Script components to create a feature engineering model and convert original character-type features to numeric features.

churn	contract_month_to_0	contract_one_year_1	contract_two_year_2	dependents_no_3	dependents_yes_4	deviceprotection_no_5	deviceprotection_no_inter_6	deviceprotection_yes_7	gender_female_8
0	1	0	0	1	0	1	0	0	1
0	0	1	0	1	0	0	0	1	0
1	1	0	0	1	0	1	0	0	0
0	0	1	0	1	0	0	0	1	0
1	1	0	0	1	0	1	0	0	1
1	1	0	0	1	0	0	0	1	1
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	1	0	0	1
1	1	0	0	1	0	0	0	1	1
0	0	1	0	0	1	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	0	0	1	0
1	1	0	0	1	0	0	0	1	0
0	1	0	0	1	0	0	0	1	0
0	0	0	1	0	1	0	0	1	1
0	0	1	0	1	0	0	1	0	1
0	0	0	1	0	1	0	0	1	0
1	1	0	0	0	1	0	0	1	1
0	1	0	0	1	0	0	0	1	1

The target field "churn" is used as an example. Run the following SQL statement to convert the original values Yes and No to 1 and 0, respectively:

```
select (case churn when 'Yes' then 1 else 0 end) as churn from ${t1};
```

1. Model training

Divide the data into two parts: a training set for model training, and a prediction set to verify the model effect. User loss warning falls in binary classification because a user is either lost or retained. Use the binary classification algorithm to create a classification model, which can be deployed in one click as a RESTful API service to be called in business scenarios.

1. Model effect verification

Use the Binary Classification Evaluation component to verify the model accuracy. An AUC of 0.83 indicates a prediction accuracy of about 80%.



Summary

User loss warning is widely used in business scenarios. PAI provides a full set of algorithms based on user features, helping customers to quickly train a user loss model in one to two days. This accelerates the process of experiment setup.

Predict the output power of generators from a wind power plant

Preface

Machine learning is widely used in industrial scenarios, with satisfying results. This experiment analyzes the power generation data of a combined cycle power plant to show how machine learning is applied to actual scenarios in industrial production.

This experiment uses the data of hybrid power plants collected from UCI machine learning datasets. For power plants, the output wind power determines the energy that a unit generator can produce. Power plants can collect metrics to predict the final output power. Power plants can also make production schedules with minimum resource waste by effectively predicting the output power of generators.

Load and explore data

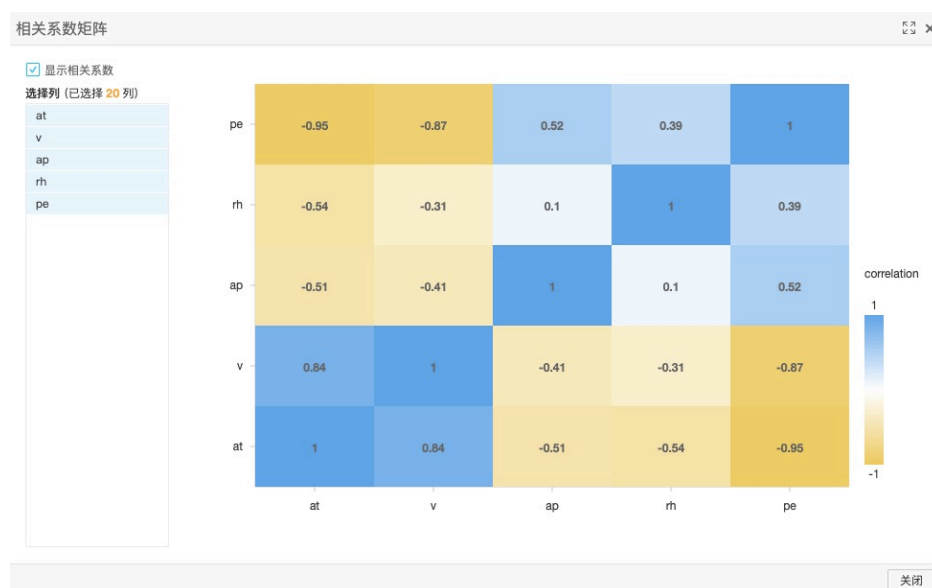
Load the dataset, which includes 9,568 data samples from a combined cycle power plant. Each data item occupies five columns: AT (atmospheric temperature), V (voltage), AP (atmospheric pressure), RH (relative humidity), and PE (output power). The following figure shows the data preview.

数据探查 - uci_cycle_power_2 - (仅显示前一百条)

序号	at	v	ap	rh	pe
1	14.96	4...	10...	73.17	463.26
2	25.18	6...	10...	59...	444.37
3	5.11	3...	101...	92.14	488.56
4	20...	5...	101...	76...	446.48
5	10.82	3...	10...	96...	473.9
6	26...	5...	101...	58...	443.67
7	15.89	4...	101...	75...	467.35
8	9.48	4...	101...	66...	478.42
9	14.64	45	10...	41.25	475.98
10	11.74	4...	101...	70...	477.5
11	17.99	4...	10...	75...	453.02
12	20.14	4...	101...	64...	453.99
13	24...	7...	101...	84.15	440.29
14	25.71	5...	101...	61.83	451.28
15	26.19	6...	10...	87...	433.99
16	21.42	4...	101...	43...	462.19
17	18.21	45	10...	48...	467.54

复制 关闭

In the left-side navigation pane, choose **Components** > **Statistical Analysis**, and drag and drop **Correlation Coefficient Matrix** to the right section. View the features related to PE (output power) to find the factor that has the greatest impact on PE (output power).



Right-click the completed component and select View Analytics Report to obtain the correlation

analysis result. The correlation chart shows the degree of correlation to PE (output power) in descending order: AT (atmospheric temperature) -> V (voltage) -> RH (relative humidity) -> AP (atmospheric pressure).

Model data

In the left-side navigation pane, choose **Components** > **Data Preprocessing**, and drag and drop **Split** to the right section to split data into the training set and test set. Then, choose **Components** > **Machine Learning** > **Regression**, and drag and drop **Linear Regression** to the right section to perform regression modeling on the data. Select the feature columns (X) and label column (Y).



Predict and evaluate the regression model

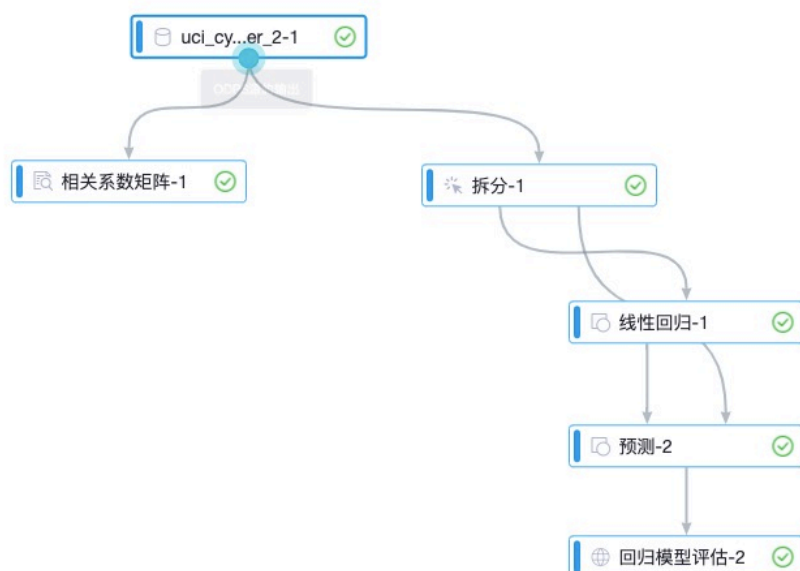
After modeling is complete, choose **Components** > **Machine Learning** and drag and drop **Prediction** to the right section to predict the effect of the model on the test dataset. Select AT, V, AP, and RH for Feature Columns, and select all options for **Reserved Output Column**.



Right-click the model and choose **Show Model** to view the weights of different features on the

number of results.

In the left-side navigation pane, choose **Components** > **Machine Learning** > **Evaluation**, and drag and drop **Regression Model Evaluation** to the right section to view the model effect. Right-click **Regression Model Evaluation** and choose **View Analytics Report**. The RMSE value reaches 4.57. The following figure shows the completed experiment.



This completes the experiment of using the linear regression model to create a power prediction model for a hybrid power plant. After being deployed, the model can predict the power generation of the power plant in real time. This helps the power plant make a better power production schedule with minimum resource waste.