Machine Learning Project

Pricing DigiKala products based on their features

AIMedic

Shahryar Namdari

15/August/2022

Final results:

Result for KNeighborsRegressor

Result of DecisionTree

Result for RandomForest

Problem:

Suppose you are going to set prices of Digikala products. Use the data of similar products and their prices and calculate the price of the remaining goods according to their type and characteristics. The data is in Farsi Language.

We have two files in dataset, train and test. Train file has three columns named id, product_description and price.

- Id ➔ unique for each product.
- Product_description ➔ in the format of dictionary containing products features.
- Price ➔ showing the price of goods.

On the other hand, test file just contains first two columns and you should predict the price column and save them inside a csv file.

Link to question: https://quera.org/problemset/138168/

Solution:

Preprocessing:

The second column of train file was a little uncanny! In addition to its dirt, it had lots of unnecessary characters and in some features like weight, the key didn't match its value!

So in first step we are going to clear the unessential characters. I removed these characters:

'\\n', '\\r', '\\u200c', '\\\\/'

Before:

```
array([1,
    "{'مشخصات فنی': 'CABLE LENTH:\\r\\n10feet \\\\/3M\\r\\nPC SIDE CONNECTORS_CONNECTED TO PC : \\r\\nPS\\\\/2 Keyboard mini-
Din 6 pin, male\\r\\nPS\\\\/2 Mouse Mini-Din 6 pin, male\\r\\nVGA HDDB 15pin, male\\r\\nKVM SIDE CONNECTORS-CONNECTED TO KVM SW
ITCH: \\r\\nHDDB 15pin, male', 'دسته بندی': 'کابل کمبو', 'برند': 'ادیمکس'}",
    990000], dtype=object)
```

After:

```
"{'مشخصات فنی': 'CABLE LENTH:10feet 3MPC SIDE CONNECTORS_CONNECTED TO PC : PS2 Keyboard mini-Din 6 pin, malePS2 Mouse Mini-Din 6'
pin, maleVGA HDDB 15pin, maleKVM SIDE CONNECTORS-CONNECTED TO KVM SWITCH: HDDB 15pin, male', 'برند', 'کابل کمبو': 'دسته بندی',
'ادیمک
س'}"
```

I also normalized prices in training process and divided them to the maximum value of training prices. In addition, I used one hot encoding for categorical data to use in my models.

In next step we need to find which features has highest repetition in products. So, I counted them and I used the top features to decide the price with. These are the features I used to predict prices (because in both train and test data, all of the products had these two features):

- برند
- دسته بندی


```
[['برند', 68840],
 ['دسته بندی', 68840],
 ['وزن', 37684],
 ['ابعاد', 31583],
 ['مناسب برای', 12543],
 ['نوع اتصال', 11568],
 ['درگاههای ارتباطی', 9065],
 ['طول کابل', 8434],
 ['رابطها', 7607],
 ['قابلیتهای مقاومتی', 7240],
 ['نوع رابط', 7081],
 ['نوع کابل', 6907],
 ['ظرفیت', 6843],
 ['سایر مشخصات', 6763],
 ['منبع تغذیه', 5880],
 ['شدت جریان خروجی', 5526],
 ['تعداد درگاه خروجی', 5442],
 ['ولتاژ باتری', 5376],
 ['ولتاژ ورودی', 5124],
```

Number of keys in train data dictionaries

LinearRegression will not help us in this kind of dataset because the data is categorical and they are independent. I actually used LinearRegression but the results were awful:

```
MAPE for rows in test_data with nonzero value ==>  14309233178.594587
MSE: ==>  2.0457823602508096e+16
Model score ==>  -3.5305056901465514e+19
```

Therefore, I chose DecisionTree, RandomForest and KNeighborsRegressor to predict.

Results for DecisionTree model:

```
MAPE for rows in test_data with nonzero value ==>  15.760008352965112
MSE: ==>  0.00020833249328032114
Model score ==>  0.5446182340007164
```


Result of DecisionTree

Results for RandomForest model:

- n_estimators = 50 ➜ I chose this after different trials
- random_state = 0
- bootstrap = True ➜ for using max_samples
- max_samples = 0.1 ➜ means 0.1 of train data will be used for each tree

```
MAPE for rows in test_data with nonzero value ==>  1.7719893130371178
MSE: ==>  0.000268525996219999
Model score ==>  0.5444942579109942
```


Result for RandomForest

Results for KNeighborsRegressor model:

- n_neighbors = 5 ➔ I chose this after few trials

```
MAPE for rows in test_data with nonzero value ==>  10.404713195083703
MSE: ==>  0.00024054795443574664
Model score ==>  0.5680807425906742
```

Result for KNeighborsRegressor

Result of different three models

Model	MAPE	MSE	Score	Result in Quera
DecisionTree	15.76	0.000208	0.544	4166
RandomForest	1.77	0.000268	0.544	4061
KNeighborsRegressor	10.4	0.000240	0.568	4042

As you can see in the above table, I used these three models because they were proper for our categorical independent data. Results are clearly shown in the table. It seems that decision three got the highest score from Quera's website; Random forest and KNN got the second and third best result respectively. But in overall, results are approximately close to each other.

Final Note:

If I had used more features for fitting my models, I would have definitely had better results; but due to shortage of time I couldn't focus on cleaning other features.