Predicting Credit Card Spend

BUSINESS PROBLEM

Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can learn more about their customers and develop more effective marketing strategies as well as increase sales and decrease costs.

Using data science, we can understand what the major factors driving credit card spend are. This spend is used by banks to calculate the credit limit. The objective of this project is to help determine a particular credit card's limit better by predicting the spend as accurately as possible.

The objective of this case study is to understand what's driving the total spend of customers. Given the factors, the goal is to predict the credit limit for new applicants.

ABOUT THE DATA (1 of 2)

```
In [10]: df.shape
Out[10]: (5000, 130)

In [11]: df.columns
Out[11]: Index(['custid', 'region', 'townsize', 'gender', 'age', 'agecat', 'birthmonth',
                'ed', 'edcat', 'jobcat',
                ...
                'owncd', 'ownpda', 'ownpc', 'ownipod', 'owngame', 'ownfax', 'news',
                'response_01', 'response_02', 'response_03'],
               dtype='object', length=130)

In [12]: df.info()
         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 5000 entries, 0 to 4999
         Columns: 130 entries, custid to response_03
```

The data has 5,000 instances and 130 features. There are 31 float features, 97 integer features, and 2 object features.

ABOUT THE DATA (2 of 2)

```
In [13]: df.describe()
```

Out[13]:

	region	townsize	gender	age	agecat	ed	edcat	5
count	5000.00000	4998.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	
mean	3.00140	2.687275	0.503600	47.025600	4.238800	14.543000	2.672000	
std	1.42176	1.425925	0.500037	17.770338	1.308785	3.281083	1.211738	
min	1.00000	1.000000	0.000000	18.000000	2.000000	6.000000	1.000000	
25%	2.00000	1.000000	0.000000	31.000000	3.000000	12.000000	2.000000	
50%	3.00000	3.000000	1.000000	47.000000	4.000000	14.000000	2.000000	
75%	4.00000	4.000000	1.000000	62.000000	5.000000	17.000000	4.000000	
max	5.00000	5.000000	1.000000	79.000000	6.000000	23.000000	5.000000	

8 rows × 128 columns

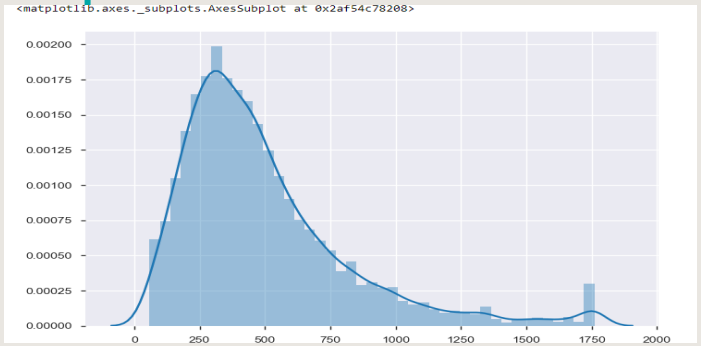
```
In [14]: df.dtypes
```

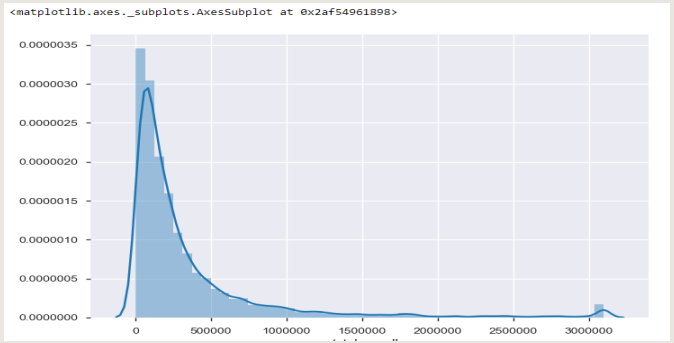
Out[14]:

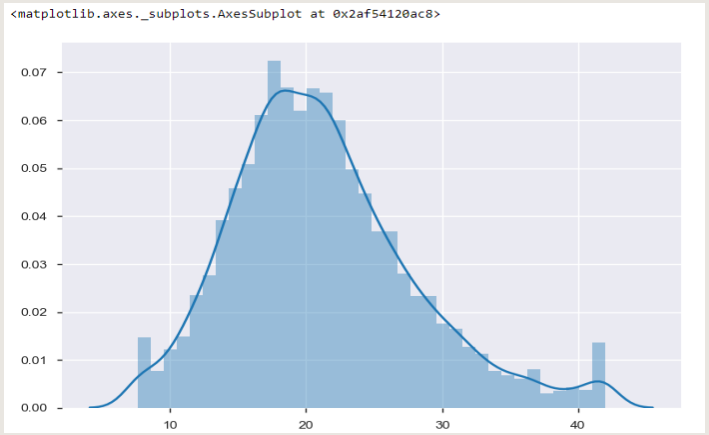
```
custid          object
region           int64
townsize       float64
gender           int64
```

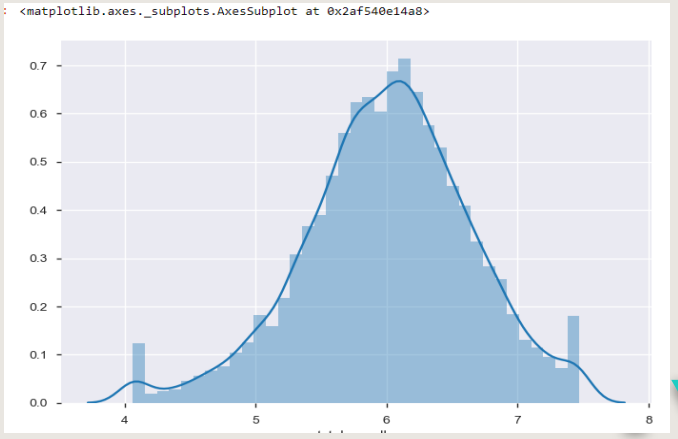
The describe function only gives a summary of numerical data. Therefore, 128 out of 130 features have numerical values. Two features- ‘custid’ and ‘birthmonth’ are of type object. ‘cardspent’ and ‘card2spent’ are the features providing the credit card spend for the primary and secondary card for a particular customer. The aim is to predict the total card spend, that is, cardspent+card2spent

TARGET VARIABLE







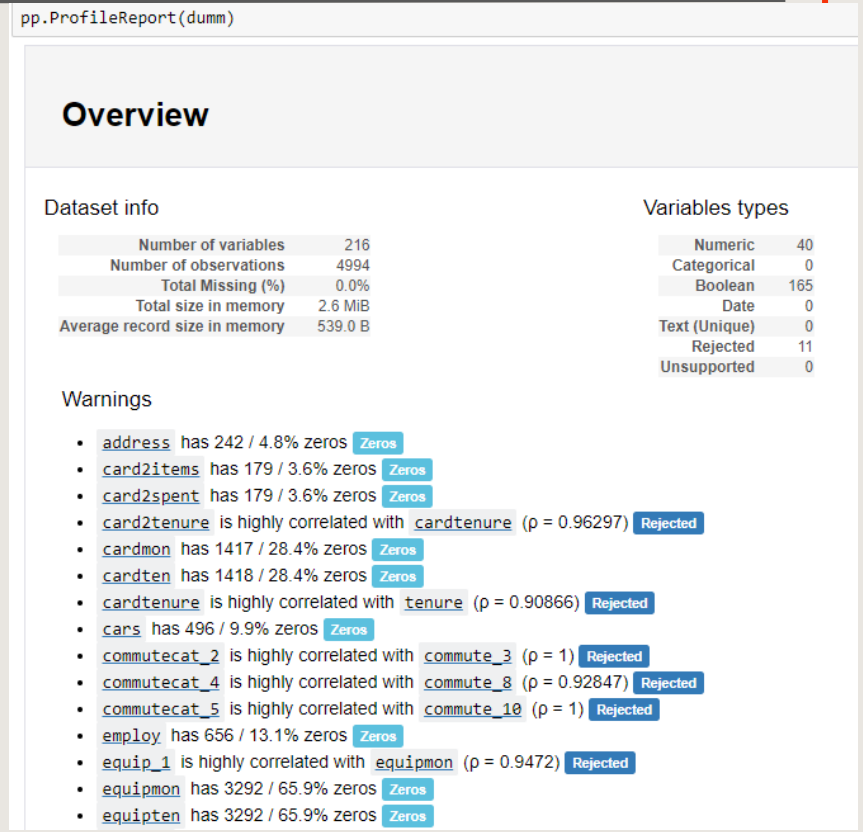


The closest operation on totalspend to a normal distribution was the log of totalspend. Therefore, totalspendln was added as a features to the dataset and totalspend was dropped

FEATURE SELECTION

Using pandas profiling, the features with high inter-correlation were recognised and dropped.

To the features left after Pandas profiling, the dataset is checked for correlation using the corr() function in Pandas. The correlation table is saved to excel and the excel file is analysed to select the features correlated with ‘totalspend’. The correlation range considered is between 0.1 to 0.7 and -0.7 to -0.1. These are the features which will be used for further analysis.



```
pp.ProfileReport(dumm)
```

Overview

Dataset info

Number of variables	216
Number of observations	4994
Total Missing (%)	0.0%
Total size in memory	2.6 MiB
Average record size in memory	539.0 B

Variables types

Numeric	40
Categorical	0
Boolean	165
Date	0
Text (Unique)	0
Rejected	11
Unsupported	0

Warnings

- address has 242 / 4.8% zeros Zeros
- card2items has 179 / 3.6% zeros Zeros
- card2spent has 179 / 3.6% zeros Zeros
- card2tenure is highly correlated with cardtenure (ρ = 0.96297) Rejected
- cardmon has 1417 / 28.4% zeros Zeros
- cardten has 1418 / 28.4% zeros Zeros
- cardtenure is highly correlated with tenure (ρ = 0.90866) Rejected
- cars has 496 / 9.9% zeros Zeros
- commutecat_2 is highly correlated with commute_3 (ρ = 1) Rejected
- commutecat_4 is highly correlated with commute_8 (ρ = 0.92847) Rejected
- commutecat_5 is highly correlated with commute_10 (ρ = 1) Rejected
- employ has 656 / 13.1% zeros Zeros
- equip_1 is highly correlated with equipmon (ρ = 0.9472) Rejected
- equipmon has 3292 / 65.9% zeros Zeros
- equipten has 3292 / 65.9% zeros Zeros

MODEL BUILDING

```
score=lm.score(test_X, test_y)
print(score*100)
```

```
88.62977955524775
```

```
score=lm.score(train_X,train_y)
print(score*100)
```

```
89.01035949291361
```

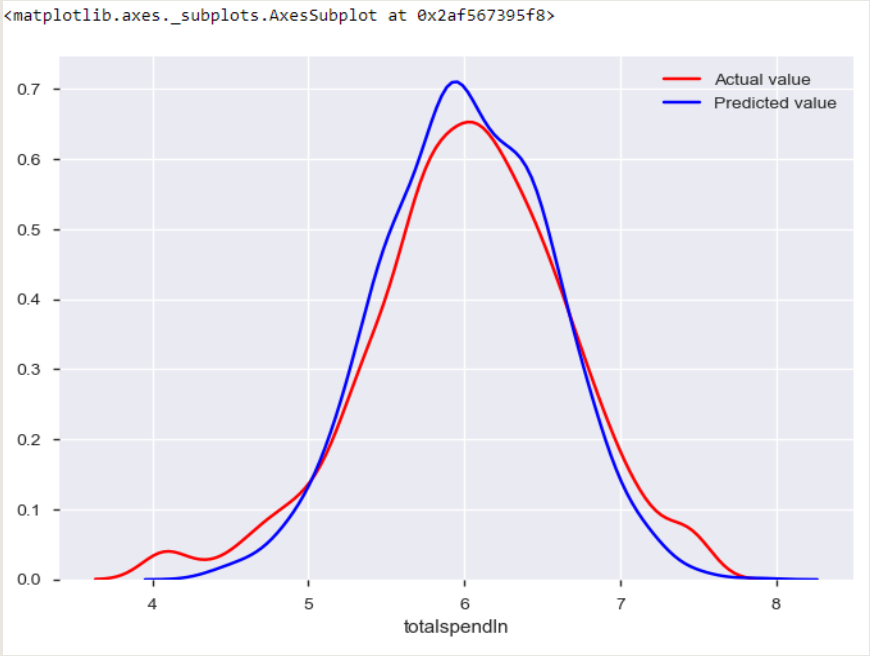
To fit the dataset with the selected features into a model, the dataset is split into training and testing parts. This is done using the train_test_split from sklearn.model_selection.

The training dataset is then fit into a linear regression model ‘lm’ and based on the trained model, predictions are made for the testing dataset.

The model gives us an 89.01% prediction score in the training dataset, and 88.62% prediction score in the testing dataset.

MODEL EVALUATION

The graph depicts a comparison between the actual values of the test dataset and the predicted values.



THANK YOU!