Loan Price Prediction

Using Ensemble Learning

Contents:

Loan Sanction Amount Prediction:

The loan given by bank to a customer is dependent on many parameters, such as, Customer’s income, their profession, area of living, credit scores, etc.

In this machine learning competition, we have a lot of information about our customer, and we have to predict the loan amount that can be sanctioned to the customers who have applied for the home loan using the features provided in the dataset.

Libraries Used:

The code to predict Loan Sanctioned Amount is written in Python 3.8 and the libraries I used are:

1. NumPy - For numeric computation
2. Pandas - For data analysis
3. Matplotlib - For visualization

$$\left.\begin{array}{ll} 4. & \text{XGBoost} \\ 5. & \text{Scikit-Learn} \\ 6. & \text{CatBoost} \\ 7. & \text{Light GBM} \end{array}\right\} \text{Ensemble Model}$$

Missing Values Imputation Techniques:

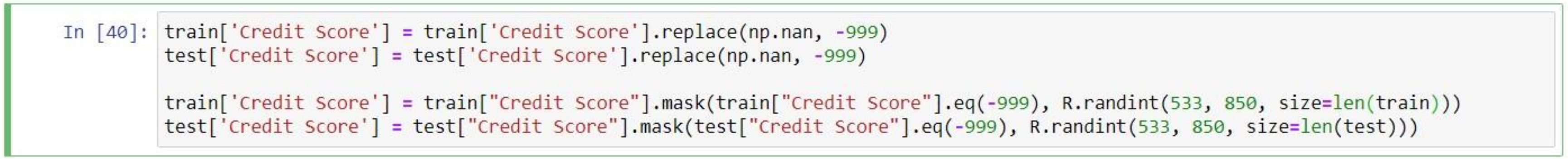
There were Three types of missing values in our dataset,

1. Numerical missing values and
2. Categorical missing values.
3. Other

Techniques I used to fill those missing value are given in the next slide.

1. Numerical Missing Values:

Columns like Credit Score and Property Price are numerical columns. We can't fill NaN values in those columns with mean or mode because it will add bias. So I filled NaNs with random values.



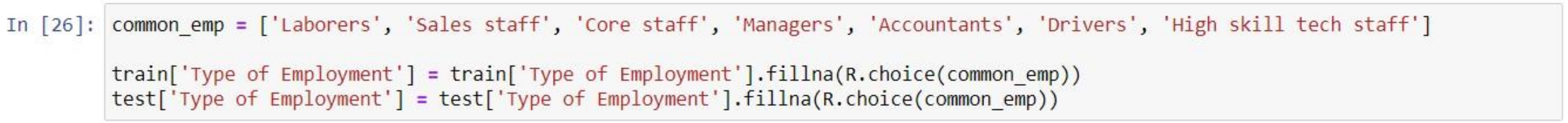
```
In [40]: train['Credit Score'] = train['Credit Score'].replace(np.nan, -999)
         test['Credit Score'] = test['Credit Score'].replace(np.nan, -999)

         train['Credit Score'] = train["Credit Score"].mask(train["Credit Score"].eq(-999), R.randint(533, 850, size=len(train)))
         test['Credit Score'] = test["Credit Score"].mask(test["Credit Score"].eq(-999), R.randint(533, 850, size=len(test)))
```

2. Categorical Missing Values:

Just like numerical NaN values, I didn't want to fill these NaN value with some wrong data, so I let the model decide.

The R in R.choice in below code is stands for Random State, so whenever I changed this random state this code filled NaNs with different value and model give me different results. So I filled NaNs with Random State of 14 because it gave me the best result.

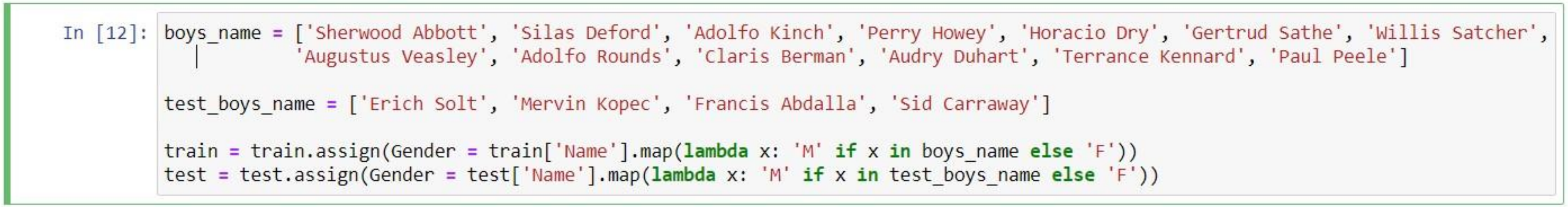


```
In [26]: common_emp = ['Laborers', 'Sales staff', 'Core staff', 'Managers', 'Accountants', 'Drivers', 'High skill tech staff']

         train['Type of Employment'] = train['Type of Employment'].fillna(R.choice(common_emp))
         test['Type of Employment'] = test['Type of Employment'].fillna(R.choice(common_emp))
```

3. Other Missing Columns:

There was a column named ‘Gender’, and it has only 52 missing values. So I filled those values by hand.

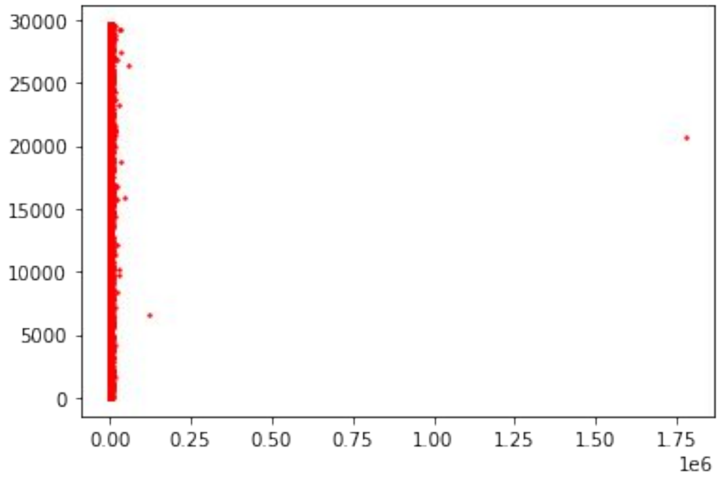


```
In [12]: boys_name = ['Sherwood Abbott', 'Silas Deford', 'Adolfo Kinch', 'Perry Howey', 'Horacio Dry', 'Gertrud Sathe', 'Willis Satcher',
                      'Augustus Veasley', 'Adolfo Rounds', 'Claris Berman', 'Audry Duhart', 'Terrance Kennard', 'Paul Peele']

         test_boys_name = ['Erich Solt', 'Mervin Kopec', 'Francis Abdalla', 'Sid Carraway']

         train = train.assign(Gender = train['Name'].map(lambda x: 'M' if x in boys_name else 'F'))
         test = test.assign(Gender = test['Name'].map(lambda x: 'M' if x in test_boys_name else 'F'))
```

Outlier Treatment:



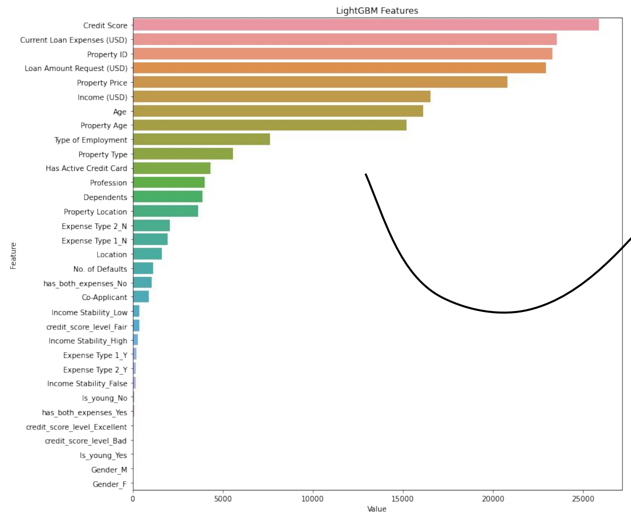
From above picture, we can see that the column named ‘Income (USD)’ has come outliers, So I removed them.

Feature Engineering Techniques:

Feature Engineering Techniques used in this model are:

1. Creating New Columns.
2. Feature Selection for model development:

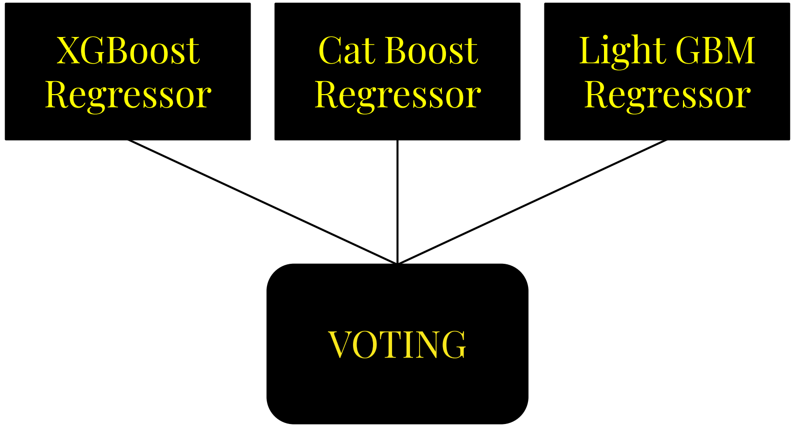
Features



Importance

Model:

I have used a Voting Regressor model to predict the loan sanctioned amount. Voting Regressor is an ensemble modelling technique which take 2 or more models and make output by taking the mean of those models.



Conclusion:

Using data analysis, feature engineering and machine learning techniques, I got 80.12% accuracy on the Public Leaderboard.

I was stuck at 79% accuracy but, using K fold cross validation, averaging two Voting Regressor models and a lot of hyper-parameter tuning gave me a little more accuracy.

It was a great competition, I researched a lot, I learned a lot.

Thank You.