Santander Value Prediction Challenge

Springboard Data Science Career Track - Capstone 2
John Peterson - February Cohort

Overview

- Background and Project Goal
- Data Wrangling
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Machine Learning
- Conclusion and Next Steps

Project Details Link

Background and Project Goal

- 80% of customers today want personalized services
- Customers more likely to do business when those services are provided
- How to anticipate customer needs in a concrete, simple and personal way?
- Determine an amount or value of a customer’s transactions

Project Goal:

- **Identify the value of transactions for each potential customer**

Data Wrangling

- Data sets are made up of anonymized customer transactions
- Test dataset 10x larger than training dataset

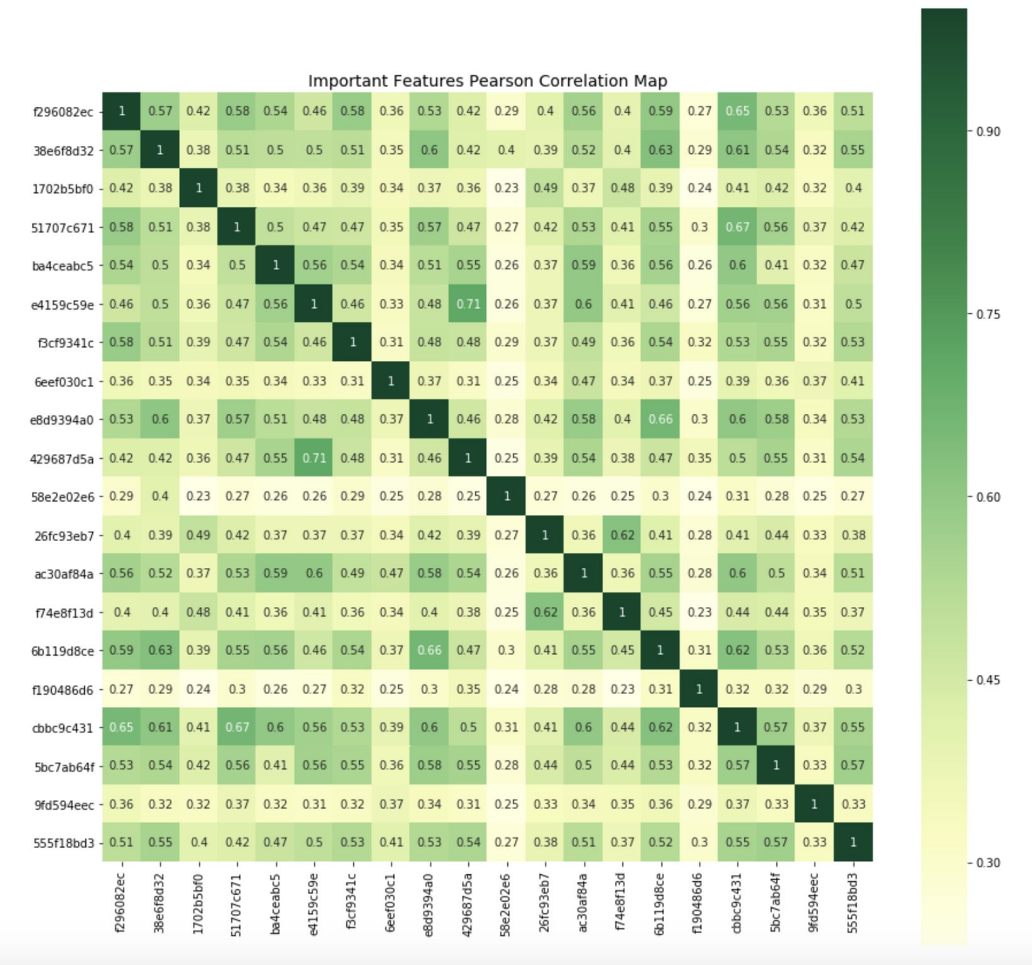
```
print("Train Shape : ", train_df.shape)
print("Test Shape : ", test_df.shape)
```

```
Train Shape :  (4459, 4993)
Test Shape :  (49342, 4992)
```

- Datasets contain many zero values
- 256 columns identified with all zero values; columns removed from datasets

Exploratory Data Analysis (EDA)

- Train and Test sets have > 4000 feature columns
- Use Pearson Correlation to find highly correlated features
- Highest correlation of 0.71 between 429687d5a and e4159c59e
- Results do not show any highly correlated features to focus on



Feature Engineering

- Difficult to create impactful features without financial background
- Data Wrangling showed zeros and non-zero values are important to datasets
- Created 3 features:
    - Sum of zero values
    - Sum of non-zero values
    - Aggregations: Max, Min, Median, Mode, VAR and Std
- Features added to help train the models

Machine Learning

- Project goal is to determine values of customer transactions
- Regression models are used to determine values
- RMSLE (Root Mean Squared Log Error) used for scoring
- Two Models used:
  - Gradient Boosting
  - Random Forest
- Baseline scores small difference with Gradient Boosting
- Random Forest scores are pointing towards overfitting

Baseline Results			
Method	Train Score	Test Score	Difference
Gradient Boosting	1.2345	1.3695	0.135
Random Forest	0.6338	1.3921	0.7583

Machine Learning

- Hyperparameter tuning can be used to improve model performance and reduce overfitting
- GridSearchCV is used to test different combinations of a range of parameters to find the optimal performing parameters
- The optimal performing parameters are shown to the right

```
GradientBoostingRegressor(alpha=0.9, criterion='friedman_mse', init=None,
             learning_rate=0.01, loss='ls', max_depth=6, max_features=0.1,
             max_leaf_nodes=None, min_impurity_decrease=0.0,
             min_impurity_split=None, min_samples_leaf=3,
             min_samples_split=2, min_weight_fraction_leaf=0.0,
             n_estimators=1000, presort='auto', random_state=20,
             subsample=1.0, verbose=0, warm_start=False))
```

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=120,
           max_features=3, max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=3,
           min_samples_split=10, min_weight_fraction_leaf=0.0,
           n_estimators=800, n_jobs=1, oob_score=False, random_state=None,
           verbose=0, warm_start=False))
```

Machine Learning

- New scores calculated using the tuned parameters (results upper table)
- Results did not show an improvement over the baseline scores
- Baseline and Tuned models were used to generate submission files using the test data set (submission scores in lower table)

Baseline Results and Tunes Score			
Method	Train Score	Test Score	Tuned Score
Gradient Boosting	1.2345	1.3695	1.3464
Random Forest	0.6338	1.3921	1.5600

Method	Public Leaderboard	Private Leaderboard
Gradient Boosting Baseline	1.49286	1.45798
Random Forest Baseline	1.54007	1.52041
Gradient Boosting Tuned	1.50086	1.46789
Random Forest Tuned	1.69940	1.67530

Conclusion and Next Steps

- Gradient Boosting was the best performing model
- Random Forest was still victim of overfitting
- Models have room for improvement

Next Steps to improve models and scoring:

- Use feature reduction tools like PCA to reduce noise
- Improve feature engineering create more impactful features
- Improve parameter tuning by reducing parameters to tune, use important features or provide more training data to model (training set was small)