Machine Learning Engineer Nanodegree
Capstone Proposal

Vamsavardhan Thotakura

May 27, 2019

1 Proposal

1.1 Domain Background

In the financial domain, banks are always looking for ways to help customers understand their financial health and identify which products and services might help them achieve their financial goals. One of the challenges in this business is to be able to get maximum customers registered for all the financial products. While each product has a specific monetary goal and intends to cater to a specific customer segment, it is not possible to go through all the customer data manually to figure out which products are suitable for them. The challenge here is, there are a lot of parameters specific to the customer, which could indicate his interest in a specific product. Given that there are numerous different financial products introduced in the industry continuously, it is essential to develop a model that could determine the customer interest on the specific product based on the various parameters.

In a typical day, I get at least 5 calls from marketing executives regarding various financial products, but most of them are not even relevant to me in any way. They make a call to every other customer in the database with the hope that a given product may be of interest to the customer. This process consumes much time and human resources if only a few of them become the actual customers of the specific product. It would save much time if a model can learn the patterns in existing customer data, and predict the prospective customers for various financial products. Hence, I'm motivated to take up a kaggle challenge with this usecase from one of the banks.

1.2 Problem Statement

I have taken the problem statement from the kaggle competition called "Santander-customer-transaction-prediction". Santander is a Spanish multinational commercial bank and financial services company founded and based in Santander, Spain. The data science team is continually challenging machine learning algorithms, working with the global data science community to make sure they can more accurately identify new ways to solve the most common challenge, binary classification problems such as: is a customer satisfied? Will a customer buy this product? Can a customer pay this loan?. The competition provides An anonymized dataset containing numeric feature variables, the binary target column, and a string ID_code column. The numeric feature variables represent various characteristics of the customer and the product. The target variable shall indicate the possibility of the customer making that transaction or not, irrespective of the transaction value. There are 200 feature columns in the data, based on which target value has to be predicted by the model.

1.3 Datasets and Inputs

The Santander bank provided anonymized data set containing 200K records. The data provided for the competition has the same structure as the real data that the bank has to solve this problem of customer transaction prediction. Each record contains ID_code, 200 numerical input variables from var_0 to var_199 and a target column. The target column value of True indicates that the customer has made the transaction, whereas false indicate that the customer didn't make the transaction.

A quick look at the data set indicates that each of these 200 numerical values in a given record corresponds to various characteristics of the customer. The mean and standard deviation of 200 features are different, indicating that some of the feature columns have abnormal distribution than others.

1.4 Solution Statement

The training data set with 200K records is used to train a machine learning model. This model shall identify the patterns in the data and build the relationship between the patterns and the target variable. Then, the model shall predict the target value for each record in the test data. In this problem, the target variable to be predicted by the model is known. Hence, one of the supervised machine learning algorithms is appropriate for the solution. The target variable in this problem is a boolean, meaning the model has to predict True or False. Classification class of machine algorithms are the best fit for these characteristics of the problem. The training data can be divided further into the training set and validation set. Validation set shall be used for model evaluation to identify the best model using various classification algorithms like support vector machines, random forest, gradient boosting techniques etc.

1.5 Benchmark Model

All the inputs variables are of numeric type. Hence, there is no need for data transformation/encoding techniques. The benchmark model is a random forest classifier with all the default hyper-parameters provided by sci-kit learn library. Any solution to this problem should perform better than the navie random forest classifier. Model built using random forest classifier shall predict whether the customer shall transact in future or not on validation data.

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier()
```

1.6 Evaluation Metrics

As mentioned in the kaggle competition submission guidelines, area under roc curve (AUC - ROC) is the evaluation metric used to determine how good the model performed on the testing data. AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between customers making transaction vs not making transaction.

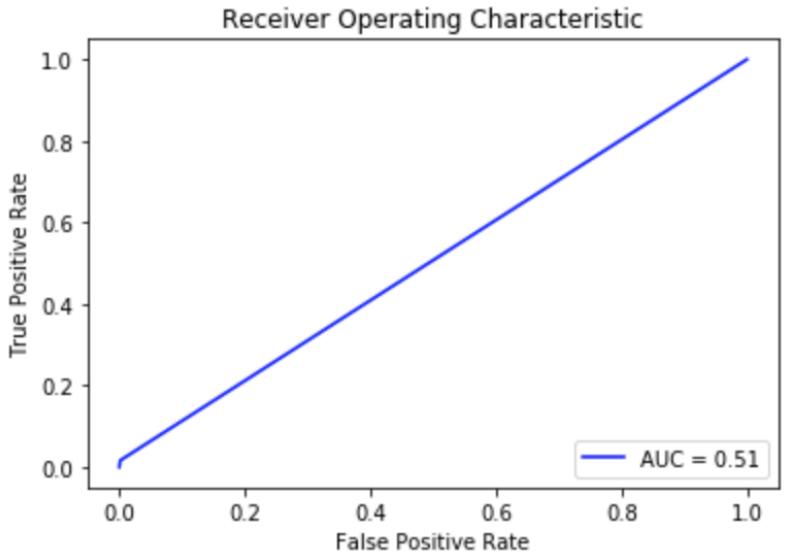
Sklearn provide the packages required to caluclate area under RUC curve -

```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import auc
```

matplotlib library shall be used to plot the ROC curve -

```
from matplotlib import pyplot as plt
```

Given below is the ROC curve for the benchmark model -



roc_benchmark.png

Area under ROC curve shall be 1 for a ideal model which can classify all the records in test data correctly.

1.7 Project Design

The detailed workflow of the project design appraching to a solution is listed below

Data exploration & pre-processing The test data is loaded into pandas data frame for data exploration. Input feature types are inspected for any non-numeric types. In this problem, all input features are of numeric type. Inspect the statistical values like mean, median, standard deviation of input variables. The input features with abnormal distribution are resclaed or normalized. A simpler naive approach is to rescale all the 200 numeric features to the range betwen 0 and 1 so that all the feature values are evently distributed. Due to data anonymity of the data set used, it is not possible to predict the important feature by its names and domain knowledge. The rest of the solution shall be approached with the assumption all the features are relavant to build the model.

Model evaluation Below classification algorithms shall be independently tried to model the data.

Random Forests with the default hyper parameters is considered as the benchmark model as mentioned in previous sections. Hyper parameter tuning may further improve the score.

```
from sklearn.ensemble import RandomForestClassifier
```

An *extra trees classifier*, otherwise known as an "Extremely randomized trees" classifier, is a variant of a random forest. Unlike a random forest, at each step the entire sample is used and decision boundaries are picked at random, rather than the best one.

```
from sklearn.ensemble import ExtraTreesClassifier
```

Gradient Boosting Classifier It will start with a (usually) not very deep tree (sometimes a decision stump - a decision tree with only one split) and will model the original target. Then it takes the errors from the first round of predictions, and passes the errors as a new target to a second tree. The second tree will model the error from the first tree, record the new errors and pass that as a target to the third tree. And so forth. Gradient boosting focuses on modelling errors from previous trees.

```
from sklearn.ensemble import GradientBoostingClassifier
```

All the above algorithms are used to build models and hyper parameter tuning shall be done using Grid search and cross validation technique to find the optimal model using each of the algorithms. I will pick the model with the best ROC-AUC value on test data.

```
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ShuffleSplit
from sklearn.metrics import make_scorer
```

The goal for the project is to get to a AUC-ROC value of alteast 0.75.

Ensemble of Ensemble models If desired AUC-ROC score is not acheived by the above ensemble models, I shall try putting two or more of the them together as an ensemble using a voting classifier or similar ensemble model.

```
from sklearn.ensemble import VotingClassifier
```

Final submission After evaluating mulitple models on the validation set, the best model shall be used to predict test data provided by the kaggle competition. The predictions shall be submitted to kaggle to determine the AUC-ROC score on the test data.

1.8 References

1. Santander Customer Transaction Prediction kaggle competition - https://www.kaggle.com/c/santander-customer-transaction-prediction/overview
2. Understanding ROC curve - https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
3. Extra treees Classifier - https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesC
4. Gradient boosting Classifier - https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.Gradi