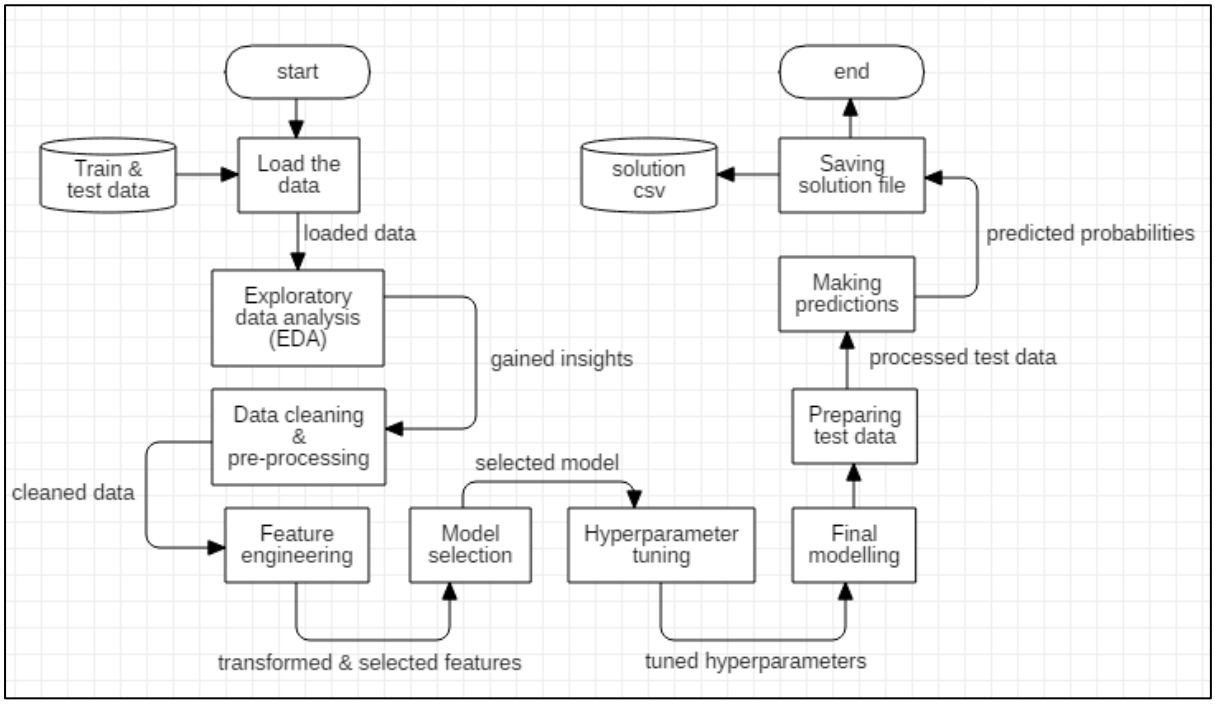
Analytics Vidhya Jobathon May 2021
Approach Document

By Rishabh Dhiman

Problem statement: Credit card lead prediction
Type of task: Classification

Approach brief:
In an approach to solve the assigned problem statement I used the strategy of 'good data with good model' so as to build a powerful yet reliable supervised machine learning model that has shown promising results on training set and has been found to be equally effective on provided test data.
With extra focus on data pre-processing and feature engineering I have tried to ensure that my model received proper and important features to train on and be able to make accurate predictions with good generalization abilities.
The selection of machine learning algorithm was carried by comparing results of different algorithms/techniques and choosing the one which gave best roc_auc score using k-fold cross validation.
Overall approach → "Good Data + Good Model + Good Hyperparameter = Best Results"

Process flowchart:



Ideas involved in each step:

1. **Exploratory data analysis (EDA):** The idea behind to conduct EDA was to check data distribution and occurrence of any mismatch or imbalance. It was done to check presence of any imbalanced categorical data in features or in target variable. Also, to see presence of uniqueness in numeric attributes to make sure that there were no columns which were constant. I also checked for any kind of mismatch which might occur between values of test and train data so to ensure uniform data elements on which the model can train.

2. **Data cleaning & pre-processing:** During this step following concerns were taken care of: -
    - Deleting any duplicate records.
    - Handling missing values of "Credit_Product" attribute by imputing null values with another category "unknown", use of this third category ensured that original distribution of "yes/no" values for the column was preserved.

3. **Feature engineering:** All the involved steps are listed below: -
    - One hot encoding to transform categorical data into numeric encoded features.
    - Feature scaling using min max scaler to get data on same scale, taking care of any possible outliers.
    - Correlation analysis to identify features which are good predictors of our target variable.
    - Dropping of irrelevant and less important features after taking into account insights gained from EDA and correlation step, this helped us to restrict number of features and taking into account only good and relevant data for modelling.

Above steps 2 & 3 ensured that "good data in → good model → good predictions out".

4. **Model selection:** For the process of selecting the best possible model, I trained the data using different algorithms/techniques and compared their scorings on test set using solution checker provided. After training a number of classical yet powerful classification like: -
    - Logistic regression
    - Support vector machine
    - Decision trees
    - Random forest
    - Gradient boosting trees
    - XGBoost classification algorithm (best score & hence selected)

    The best model which gave us maximum roc_auc score was xgboost hence I selected the same to be my final classifier.

5. **Hyperparameter tuning:** Training a model with tuned hyperparameters is necessary to ensure that model can leverage best of its power to achieve its given task. For the same I tested out different hyperparameter for xgboost using Grid Search technique.

6. **Final modelling:** After getting value of best hyperparameters I fine tuned my xgboost classifier accordingly, which was then trained on the training data provided.

7. **Preparing test data and making predictions**: After getting our final model, I used it to make predictions on cleaned test data values so as to get probabilities for each new value showcasing their interest.

Data pre-processing & feature engineering ideas that worked:
In this section, I would like to explain about a certain observation that really helped me improve my score. During EDA I observed that there were in total 35 different region code however top 25 of them accounted of about 83% alone while rest 10 accounted for 17%. Further during correlation analysis, it was revealed that region codes were in fact not good predictors for our

target variable. Hence, I decided to drop least frequent region codes in order to reduce the number of features. Training model on new reduced features helped me shoot up my score by a large margin.
The idea of over sampling using SMOTE to balance target class was also explored however it did not provide with effective results hence it was dropped.

Final Model:
This is how my final model looks like: -
XGBoost Classification

```
XGBClassifier(base_score=0.5,
        booster='gbtree', colsample_bylevel=1,
        colsample_bynode=1,
        colsample_bytree=0.8,
        gamma=0,
        learning_rate=0.03,
        max_delta_step=0,
        max_depth=6,
        min_child_weight=3,
        n_estimators=500, n_jobs=1,
        nthread=1,
        objective='binary:logistic',
        random_state=0,
        reg_alpha=0,
        reg_lambda=1,
        scale_pos_weight=1,
        silent=1,
        subsample=0.7,
        verbosity=1)
```