KOALACODERS

KAGGLE USERNAME: - ds1030@rotaract.social

KAGGLE DISPLAY NAME: - ds1030

GITHUB LINK: - https://github.com/ganeshaaraj97/DATASTORM-KOALACODERS

HIGHEST F1 SCORE: - 0.82416

TEAM MEMBERS

1. **B.GOWRIENANTHAN**
2. **R.THUSHANY**
3. **G.GANESHAARAJ**

OVERVIEW

Our whole process can be divided into 5 main steps

1. **Data Pre-Processing& Building Transformation Pipelines-**We used **pandas** to import, export and maintain data frames, and **numpy** for matrix operations on the dataset. **Sklearn** was used for data analysis and making machine learning models as explained in the rest of the article. **Matplotlib** was used to plot and visualize data during various analyses. There were categorical attributes; to handle those we used **Onehotencoding** for our dataset. After encoding our features increased to 49. We created a transformation pipeline to standard scale numerical data and onehotencode categorical data
2. **Feature Selection-** We selected most important features and eliminated the leaset relevant ones with the help of" feature selection using linear models" technique. The ideas worked and justifications is elaborated in coming section
3. **Upsampling-** When we observed the training dataset, the dataset was skewed towards '0'. The biggest problem is all the models we trained showed high accuracy in predicting '0' and vice versa for '1'.To cater this situation, the initial step was over-sampling the training dataset using **SMOTE** and creates a balance in the dataset.
4. **Model Selection** – In our best model we used **XGBoost**. The process to selection of this model and reasoning is elaborated in coming sections.
5. **Train, Validation and Test sets-** using scikit train/test random split library we separated 19200 data points as training set and 4800 data points as test set. Using train set we modified and upgraded our model in pursuit to achieve high frequency. Then we tested our model in test set.

FEATURE ENGINEERING

Feature extraction is a way of reducing the dimensionality of data. The goal is to reduce the number of features in the data by generating new features from existing ones, while preserving as much information as possible. This has various benefits such as reducing load on model training, increasing accuracy, reducing the risk of over fitting, better data visualization etc. The methods for feature extraction that we chose are **Principal Component Analysis** (PCA) and **Linear Discriminant Analysis** (LDA).

Principal Component Analysis is a very popular method of dimensionality reduction. It projects the data onto the hyper plane that lies closest to the dataset. This hyperplane is of a lower dimensionality than the original data. The **PCA** algorithm tries to preserve as much variance of the data as possible. This is where the "Principal Components" come in. Principal components are the axes on which, when the data is projected, will preserve the maximum variance.

We created a function that takes as inputs, the data, and the number of components we require and it returns the reduced data. With a for loop, **PCA** was applied to the data with a range of number of components, ranging from 1 to the maximum which is one less than the number of features of the original dataset. The results obtained were that the best accuracy (81.9%) was obtained when the number of components was 22.

The other method of dimensionality reduction that was applied was **Linear Discriminant Analysis**. The **LDA** algorithm attempts to maximize the distance between the mean of each of the classes, while minimizing the spread of data points within a class. This makes it easier to separate the classes. Since the dataset has only two classes, LDA reduced all the original features into one feature. With LDA alone, the accuracy obtained was 81.5%.

Next, PCA was applied to the dataset, and then LDA was applied. This method of applying PCA before LDA will help regularize the problem while also helping to avoid overfitting. The same technique as above was used to try a range of values for the number of components for the PCA algorithm. The best accuracy was obtained when the number of components was 21. The best accuracy was 81.95%. This shows that performing PCA before performing LDA has accuracy benefits.

Even though the accuracy was promising our search to achieve high accuracy/F1 score didn't stop there. We turned our attention to not so popular feature selection method called "**FEATURE SELECTION USING LINEAR MODELS**". We made little tweak in this approach and eliminated least correlating features. In this approach we used Logistic Regression and SGDClassifier to identify least relevant feature. As first step we trained the model using the training data. After training, the model will assign weights to each and every feature in the training data. Weights can be positive or negative.

So feature with largest (positive/negative) value of weight is highly correlated feature. In our case we plotted the weights given by two different models and analysed them. We observed that features **–["DUE_AMT_SEP" and "PAY_SEP"]** were least relevant features. So we eliminated both the features from the dataset and up sampled the data. Then we used the up sampled data to train XGBoost. **We achieved F1 score of 0.82416.(which is our best score)**

Final Model

XGBoost is the best model we used,but before coming onto this decision we studied and tested lot of various models. As usual we started with Logistic Regression and we got accuracy of 0.6578 then we tried an SVM with the help of sklearn-python and with that we were able to obtain an accuracy of 0.691. Even after using Random Forest and Neural network we were not able to achieve accuracy over 0.7500. That's when we observed the training dataset, the dataset was skewed towards '0'. The biggest problem is all the models we trained showed high accuracy in predicting '0' and vice versa for '1'.To cater this situation, the initial step was over-sampling the training dataset using **SMOTE** and to create a balance in the dataset. After trying with lot of models we finalised two models to work with to achieve high F1 score. The models were **Multiple random forests with feature splitting and XGBoost.**

Multiple Randomforests with feature splitting- we chose this approach because during our initial model search, Randomforest gave high F1 score than other models. So our motive was to split the features and to train multiple random forests on each of the split-sets. We split the features into two and trained distinct two Randomforests on them. We used Gridsearch library in sklearn to fine-tune parameters. Then we used another Randomforest to train results from previous two Randomforests and all features to make final predictions. We tested this model in our test set, we got F1 score of 0.82400.

XGBoost- This is our final and best model. It is an efficient implementation of the stochastic gradient boosting algorithm and offers a range of hyper parameters that give fine-grained control over the model training procedure. XGBoost performs better with unbalanced dataset comparing to other models offered by sklearn. At the beginning we tried default XGBoost models with default parameters. Then using GridsearchCV we tried to fine tune parameters. Surprisingly default XGBoost performed way better than XGBoost model with changed parameters.

We tested this model in our test set, we got F1 score of 0.82416. During the 3 days of our work, this was our highest " F1 score" achieved.

Business Insights (Specifications and Recommendations)

The model leads to a prediction accuracy of 82.416%. As per the correlation heat map and values it is evident that pay amount for every month have a moderate positive correlation with the default values and it is also affecting the due amounts positively. Therefore, PAY_AMOUNT is considered as the influencing predictor for our model.

Compared to due amounts and paid amounts the influence of pay amount to decide the possibility of a customer default payments seems to be likely. If a creditor could make his/her credit payments in advance or on time without delay it is a good credit practice and more chances for the creditor to avoid being a default payee. Whereas the due amount and paid amount can significantly vary in large ranges which might lead to misleading predictions. Also, if we consider November and December due to festive seasons there can be a chance of excessive purchasing and over spending as a result there will be a significant pattern in the values of due_amount and paid_amonts for November and December which might lead to incorrect predictions.

MARKETING INSIGHTS TO AVOID DEFAULT PAYMENTS FORM PRIORITY CLIENTS.

- It is identified the male population is high compared to the female population in the priority clients therefore increased rates of fines can be imposed for some overpriced products like branded men clothes, etc. during the periods of over-spending as identified by the model.
- It is identified during the months of festival there is a large due amount prevalent. Therefore, fund release can be controlled and purchasing over-priced products can be restricted if the client is predicted to reach default payment situation.
- Special budget for limited spending can be planned and encouraged to follow by customers predicted in the priority list. In this way disagreements between the credit card holders and lenders could be prevented and this is a best precaution to avoid payment default.
- Negotiate to improve the credit score of the creditor and offer creditworthiness in exchange for full payment of the due amount once identified as a potential default payee by the model prevalent.
- The potential default credit clients who are identified through the model can be transferred to a third party debt collecting agent timely for under contract abiding the necessary accounting standards and suitable commission. This is an efficient risk management method.