**NYC Taxi Trip Time Prediction**

**Krushna Chaure**

**Data Science Trainee,**

**AlmaBetter, Bangalore**

**Abstract**

Credit card default prediction is an important problem for credit card companies as it helps them reduce their financial risks and improve their profitability. In this project, we explore the process of credit card default prediction using the Credit Card Default Dataset from the UCI Machine Learning Repository. We perform data cleaning, exploratory data analysis, feature engineering, and modeling to train and test various classification algorithms such as logistic regression, decision trees, support vector machine(SVM), random forests, XGBoost Classifier. By using techniques such as cross-validation and grid search, we select the best model based on performance metrics such as accuracy, precision, recall, and F1 score. Our findings show that with careful data preparation and modeling, we can build accurate models that help credit card companies identify risky borrowers and reduce their financial losses.

**Introduction**

Credit card default prediction is an important problem for credit card companies as it helps them minimize their financial risks and improve their profitability. Default occurs when a borrower fails to repay their outstanding balance for an extended period, leading to significant financial losses for credit card issuers. Predicting credit card defaults is a complex problem that requires careful data preparation and modeling. In recent years, machine learning techniques have been used to develop accurate models that help credit card issuers identify risky borrowers and take proactive measures to prevent defaults. In this project, we will explore the process of credit card default prediction using the Credit Card Default Dataset from the UCI Machine Learning Repository. We will perform data cleaning, exploratory data analysis, feature engineering, and modeling to train and test various classification algorithms. The goal of this project is to build accurate models that help credit card issuers minimize their financial risks and improve their profitability.

**Problem Statement**

The problem of credit card default prediction is to develop a model that can accurately predict whether a credit card client is likely to default or not. This is a binary classification problem where the target variable is a binary variable indicating whether a client has defaulted or not. The goal of this problem is to help credit card issuers identify risky borrowers and take proactive measures to prevent defaults. Default can lead to significant financial losses for credit card companies, so accurately predicting defaults is essential for managing financial risks and improving profitability. The Credit Card Default Dataset from the UCI Machine Learning Repository contains information on credit card clients in Taiwan, and our goal is to use this dataset to build accurate models that can predict credit card defaults.

**Dataset Description**

The Credit Card Default Dataset used for credit card default prediction contains information on credit card clients in Taiwan from April 2005 to September 2005. The dataset has 30,001 records and 24 features. The target variable is a binary variable indicating whether a client has defaulted or not. The dataset includes the following features:

**The data set contains the following columns:**

We have records of 30001 customers. Below are the description of all features:

* ID: ID of each client
* LIMIT\_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit)
* SEX: Gender (1 = male, 2 = female)
* EDUCATION: (1 = graduate school, 2 = university, 3 = high school, 0,4,5,6 = others)
* MARRIAGE: Marital status (0 = others, 1 = married, 2 = single, 3 = others)
* AGE: Age in years
* History of past payment
* **We tracked the past monthly payment records from April to September, 2005.The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.**
* PAY\_0: Repayment status in September, 2005 (scale same as above)
* PAY\_2: Repayment status in August, 2005 (scale same as above)
* PAY\_3: Repayment status in July, 2005 (scale same as above)
* PAY\_4: Repayment status in June, 2005 (scale same as above)
* PAY\_5: Repayment status in May, 2005 (scale same as above)
* PAY\_6: Repayment status in April, 2005 (scale same as above)
* Amount of bill statement (NT dollar)
* BILL\_AMT1: Amount of bill statement in September, 2005 (NT dollar)
* BILL\_AMT2: Amount of bill statement in August, 2005 (NT dollar)
* BILL\_AMT3: Amount of bill statement in July, 2005 (NT dollar)
* BILL\_AMT4: Amount of bill statement in June, 2005 (NT dollar)
* BILL\_AMT5: Amount of bill statement in May, 2005 (NT dollar)
* BILL\_AMT6: Amount of bill statement in April, 2005 (NT dollar)
* Amount of previous payment (NT dollar)
* PAY\_AMT1: Amount of previous payment in September, 2005 (NT dollar)
* PAY\_AMT2: Amount of previous payment in August, 2005 (NT dollar)
* PAY\_AMT3: Amount of previous payment in July, 2005 (NT dollar)
* PAY\_AMT4: Amount of previous payment in June, 2005 (NT dollar)
* PAY\_AMT5: Amount of previous payment in May, 2005 (NT dollar)
* PAY\_AMT6: Amount of previous payment in April, 2005 (NT dollar)
* default.payment.next.month: Default payment (1=yes, 0=no)

**Data Cleaning:**

Before analyzing the data, we must first clean it by removing missing values, duplicates, and irrelevant features. The dataset has no missing values or duplicates, so we can proceed with feature selection**.**

**Exploratory Data Analysis:**

Exploratory data analysis is the process of visualizing and summarizing the data to gain insights into the variables' distributions and relationships. We can use various visualizations such as histograms, box plots, and scatter plots to identify trends and patterns in the data. For example, we can visualize the distribution of the target variable using a histogram to check if the dataset is balanced or imbalanced.

**Hypothesis Testing**

Hypothesis Testing is a type of statistical analysis in which you put your assumptions about a population parameter to the test. the theory, methods, and practice of testing a hypothesis by comparing it with the null hypothesis. The null hypothesis is only rejected if its probability falls below a predetermined significance level, in which case the hypothesis being tested is said to have that level of significance.

**Feature Engineering**

feature engineering is an important step in credit card default prediction because it involves transforming the original features into more informative ones that can improve the performance of the model. Here are some feature engineering techniques that we can apply to the Credit Card Default Dataset:

**1. SMOTE(Synthetic Minority Oversampling Technique):-**

As we have seen earlier that we have imbalanced dataset. So to remediate Imbalance we are using SMOTE(Synthetic Minority Oversampling Technique)

In SMOTE, synthetic examples of the minority class are created by interpolating new instances between existing instances of the minority class. Specifically, SMOTE creates new instances by selecting two random instances from the minority class and interpolating a new instance along the line segment joining the two selected instances. The number of new instances to be generated can be specified as a multiple of the original minority class size.

**2. Training and Scaling the Data**

In summary, training and scaling the data are important steps in preparing the Credit Card Default Dataset for building a credit card default prediction model. Training the data involves splitting the dataset into a training set and a test set, while scaling the data involves transforming the numerical features to have a common scale using techniques such as standardization. These steps can help improve the performance of the model and make it more robust to new, unseen data.

**Performance Matrix:-**

The performance metrics used to evaluate the credit card default prediction model depend on the specific problem and the goals of the project. Here are some common performance metrics used in credit card default prediction:

**1. Accuracy:** Accuracy measures the proportion of correctly classified instances. It is a common metric used in binary classification problems, such as credit card default prediction.

**2. Precision:** Precision measures the proportion of true positive predictions among all positive predictions. It is a metric that is useful when the cost of a false positive prediction is high.

**3. Recall:** Recall measures the proportion of true positive predictions among all actual positive instances. It is a metric that is useful when the cost of a false negative prediction is high.

**4. F1 Score:** The F1 Score is the harmonic mean of precision and recall. It is a metric that is useful when both precision and recall are important.

**4. Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** The AUC-ROC measures the performance of a binary classifier as the discrimination threshold is varied. It is a metric that is useful for comparing the performance of different models.

**5. Confusion matrix:** A confusion matrix is a table that summarizes the number of true positives, true negatives, false positives, and false negatives for a binary classifier. It can be used to calculate other performance metrics such as precision, recall, and accuracy.

These performance metrics can help evaluate the performance of a credit card default prediction model and provide insights into how well the model is performing. Depending on the specific problem and goals of the project, some metrics may be more important than others. Therefore, it is important to carefully consider the appropriate performance metrics for the specific credit card default prediction problem.

**Machine Learning Models Implementation**

Here we implemented 5 ML Model:-

1. Logistic Regression
2. Decision Tree Classifier
3. Support Vector Machine(SVM)
4. Random Forest Regressor
5. XGBoost Classifier

**1. Logistic Regression**

Logistic regression is a commonly used machine learning algorithm in credit card default prediction. In this problem, the goal is to predict whether a credit card holder is likely to default on their payment in the next month based on a set of input features such as age, credit limit, payment history, etc.

Logistic regression is particularly well-suited for this problem because it is a binary classification algorithm that can output the probability of default. By setting a threshold on the predicted probability, the logistic regression model can be used to make binary predictions of whether an individual is likely to default or not.

One advantage of logistic regression is that it is a simple and interpretable model. The coefficients of the logistic regression model can provide insights into the relationships between the input features and the target variable. For example, the model may reveal that the credit limit is the most important feature in predicting credit card default.

Another advantage of logistic regression is that it can handle both numerical and categorical input features. The model can be trained on a combination of numerical features (such as age, credit limit, and payment amount) and categorical features (such as education level, marital status, and payment history).

In summary, logistic regression is a commonly used machine learning algorithm in credit card default prediction. It is a binary classification algorithm that can output the probability of default and is particularly well-suited for this problem due to its simplicity and interpretability.

**2. Decision Tree Classifier**

Decision tree classifier is another commonly used machine learning algorithm in credit card default prediction. It is a non-parametric algorithm that is well-suited for handling both numerical and categorical input features, making it a popular choice for credit card default prediction problems.

A decision tree classifier works by recursively splitting the input features into subsets based on the values of the features that provide the most information gain. The goal is to create a tree that can classify instances by asking a series of yes or no questions based on the input features. Each node in the tree represents a test on an input feature, and the edges represent the possible outcomes of the test.

In credit card default prediction, a decision tree can be used to identify the most important features that are associated with credit card defaults. For example, the decision tree may reveal that customers with high credit card balances and who have missed payments in the past are more likely to default on their payments.

One advantage of decision trees is that they are easy to interpret and visualize. The decision tree structure can be easily visualized, allowing for insights into the most important features and decision rules.

Decision trees can also handle missing values in the input data by making decisions based on the available features.

However, decision trees can be prone to overfitting, especially when the tree is deep and complex. Overfitting occurs when the tree is too specific to the training data and does not generalize well to new data. One way to prevent overfitting is to use regularization techniques such as pruning, which involves removing branches that do not improve the performance of the tree.

In summary, decision tree classifiers are commonly used in credit card default prediction because they can handle both numerical and categorical input features and provide interpretable insights into the most important features and decision rules. However, they are prone to overfitting and require regularization techniques to prevent overfitting.

**3. Support Vector Machine(SVM)**

Support vector machine (SVM) is another commonly used machine learning algorithm in credit card default prediction. SVM is a binary classification algorithm that works by finding a hyperplane that separates the positive and negative instances in the input feature space.

In credit card default prediction, SVM can be used to find the hyperplane that best separates the features of credit card holders who are likely to default from those who are not. The SVM algorithm tries to maximize the margin between the hyperplane and the closest data points in each class, which helps to reduce the risk of overfitting and improve the generalization performance of the model.

SVM can handle both linearly separable and non-linearly separable datasets by using kernel functions that map the input features into a higher-dimensional space where the data may be separable by a hyperplane. This allows SVM to capture more complex relationships between the input features and the target variable.

One advantage of SVM is that it is a powerful algorithm that can handle high-dimensional feature spaces and large datasets. SVM can also handle imbalanced datasets by using techniques such as weighted classes or cost-sensitive learning to account for the unequal distribution of the positive and negative classes.

However, SVM can be computationally expensive to train and may require careful tuning of the hyperparameters to achieve good performance. The choice of kernel function and its associated hyperparameters can also have a significant impact on the performance of the SVM model.

In summary, SVM is a powerful machine learning algorithm that can handle both linearly and non-linearly separable datasets in credit card default prediction. It can also handle high-dimensional feature spaces and imbalanced datasets. However, SVM can be computationally expensive to train and requires careful tuning of the hyperparameters.

**4. Random Forest Classifier**

Random Forest Classifier is a popular machine learning algorithm that belongs to the ensemble learning category. It is a powerful technique that is used for both classification and regression tasks. The random forest algorithm works by creating a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The trees in the random forest are constructed using a random subset of the features, which helps to reduce the variance and overfitting of the model.

Hyperparameter Tuning: Finally, you can fine-tune the model by adjusting its hyperparameters, such as the number of trees in the forest, the depth of the trees, and the number of features considered at each split. This can be done using techniques such as grid search or random search.

Overall, random forest classifier can be an effective tool in credit card default prediction, as it can handle non-linear relationships between features and can provide insight into which features are most important in predicting default.

**5. XGboost Classifier**

XGBoost (Extreme Gradient Boosting) Classifier is another popular machine learning algorithm that can be used in credit card default prediction. Here are the steps to build an XGBoost Classifier model for credit card default prediction.

XGBoost (Extreme Gradient Boosting) Classifier is a popular machine learning algorithm that belongs to the ensemble learning category. It is a powerful technique that is used for both classification and regression tasks. The XGBoost algorithm works by combining the output of many decision trees that are trained sequentially to improve the accuracy of the predictions.

The XGBoost algorithm is trained on the dataset using the selected features. This involves splitting the data into training and testing sets, and fitting the model to the training set. XGBoost uses an ensemble of decision trees to create a predictive model.

Hyperparameter Tuning: Finally, the model is fine-tuned by adjusting its hyperparameters, such as the learning rate, maximum depth of the trees, and the number of estimators. This can be done using techniques such as grid search or random search.

XGBoost Classifier is widely used in various domains, including finance, healthcare, and marketing, for tasks such as fraud detection, disease diagnosis, and customer segmentation. It is a powerful algorithm that can handle high-dimensional data, noisy data, and missing values, and can provide insight into which features are most important in predicting the target variable. XGBoost is known for its speed, scalability, and accuracy, making it a popular choice in many machine learning applications.

**Conclusion**

In conclusion, credit card default prediction is a crucial task in the banking and financial industry. It involves using historical data of customers' credit card usage and payment behavior to predict the likelihood of future credit card defaults. Machine learning algorithms such as Logistic Regression, Random Forest, and XGBoost Classifier have been widely used to build predictive models for credit card default prediction.

The success of a credit card default prediction model depends on several factors, such as the quality and size of the data, the relevance of the selected features, the performance of the algorithm, and the interpretability of the results. Therefore, it is important to carefully select and preprocess the data, perform feature selection and engineering, train and evaluate the model using appropriate metrics, and interpret the results to gain insights into the factors that contribute to credit card defaults.

Overall, credit card default prediction models can help financial institutions identify high-risk customers, prevent potential losses, and improve their decision-making processes. These models can also help customers to better understand their credit card usage behavior and take necessary steps to avoid defaulting on their payments.

**Acknowledgement**

This project was completed by Krushna Chaure. I am extremely grateful to the celebrated authors whose precious works have been consulted and referred to in this project work. I also wish to convey my appreciation to our peers who provided encouragement and timely support in the hour of need.

**References**

1. Stackoverflow
2. Geeksforeeks
3. Jovian
4. Towardsdatascience
5. Medium
6. Google
7. Chatgpt