American Express Kaggle Dataset: Insights into Fraudulent Transactions Detection

1st Sudhanshu Mukherjee
Advanced-Data Mining
University of Massachusetts, Dartmouth
New Bedford, United States
smukherjee3@umassd.edu

2nd Vinil Harsh
Advanced-Data Mining
University of Massachusetts, Dartmouth
New Bedford, United States
vharsh@umassd.edu

3rd Balakrishna Vardhineni
Advanced-Data Mining
University of Massachusetts, Dartmouth
New Bedford, United States
bvardhineni@umassd.edu

***Abstract*— This Project examines the effectiveness of machine learning algorithms such as XGBoost, CatBoost, and logistic regression in detecting fraudulent transactions using the American Express Kaggle dataset. The dataset includes a large sample of transaction data, both legitimate and fraudulent, and allows for the creation of predictive models for fraud detection. The study provides insights into the performance of these algorithms and their ability to accurately identify fraudulent transactions, which can be used to improve fraud detection systems in financial institutions. Overall, the results suggest that these algorithms are effective tools for detecting fraudulent transactions and can be valuable in preventing financial loss due to fraud.**

Keywords*— fraudulent transaction, American Express dataset, XG-Boost Algorithm, Cat-Boost Algorithm, Logistic Regression Algorithm

I. Introduction (*Heading 1*)

The increasing use of electronic payment systems and the growth of e-commerce have made financial transactions more convenient and accessible than ever before. However, this also comes with the risk of fraud, which can have severe consequences for individuals and businesses alike. Fraudulent transactions can result in financial losses, reputational damage, and legal consequences. Therefore, detecting and preventing fraud is of utmost importance for financial institutions and online marketplaces. To address this issue, many financial institutions and online marketplaces are turning to machine-learning algorithms for fraud detection. These algorithms are designed to analyze large amounts of transaction data, identify patterns, and predict the likelihood of fraudulent activity. One such dataset that has been widely used for this purpose is the American Express Kaggle dataset, which contains transaction data from millions of American Express cardholders. We explore the effectiveness of machine learning algorithms such as XGBoost, CatBoost, and logistic regression in detecting fraudulent transactions using the American Express Kaggle dataset. The aim of this research is to provide insights into the performance of these algorithms and their ability to accurately identify fraudulent transactions.

II. Dataset

A. Dataset Overview

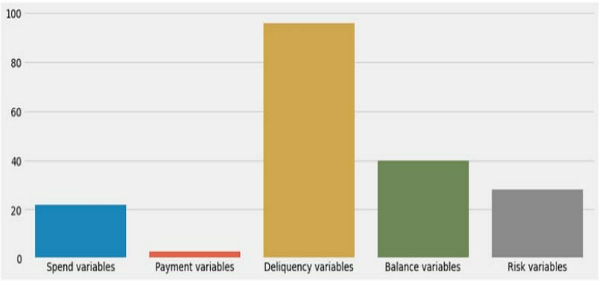
American Express dataset is built with the real-world customer's transaction details so that it can use machine learning models to detect fraudulent transactions of their real customers. Since they have features based on real customers, they have anonymized and normalized feature names and assigned Customer ID to each customer. This dataset has 5.5 million rows and 192 features. Our analysis is based on 100,000 rows and all the features considering the computation power that required this amount of Big Data.

The Features present in this Dataset are divided into 5 categories:

- Delinquency variables (D_*),
- Spend variables (S_*),
- Payment variables (P_*),
- Balance variables (B_*), and
- Risk variables (R_*).

While most of these variables represent numerical values, there are 145 delinquency variables, of which nine are categorical variables (D_114, D_116, D_117, D_120, D_126, D_63, D_64, D_66, D_68). There are 42 Balance variables, of which two are categorical (B_30 and B_38).

The graph below shows how the data is distributed with respect to each of the variable categories. We can see that these variables are not equally distributed among each category with a high number of delinquency variables (96) and very few payment variables (3).



B. Dataset Transformation

The project involves compressing a large dataset in Parquet format, which was earlier 55 GB in size but after compression, the dataset becomes 1.8GB in size. To reduce the size of the dataset, Float64 data types were converted to Float16, Int64 data types were converted to Int8, and Object data types were converted to STR. However, even after compression, the dataset remained computationally impossible to work with. As a result, the analysis was limited to only 100,000 rows of data.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

III. Data Pre-processing

Data preprocessing is a crucial step in any machine learning project and involves preparing the data in a format that is suitable for analysis. In the context of the project, data preprocessing could involve cleaning and filtering the data to remove any irrelevant or incomplete observations, handling missing data or outliers, and scaling or normalizing the features to ensure they have a similar range or distribution.

A. Reading Data

Dask is an open-source Python library for parallel computing. Dask scales Python code from multi-core local machines to large distributed clusters in the cloud. Dask provides a familiar user interface by mirroring the APIs of other libraries in the PyData ecosystem including Pandas, scikit-learn, and NumPy.

We used Dask to read and manipulate our Data as it provides parallel computing abilities for large-scale data and our data qualifies all the metrics to be considered as Big Data, thus we decided to go with Dask.

B. Handling Missing Values

- Handling missing values is a crucial step in data preprocessing for machine learning.
- When moving forward with our testing, we ran into 2 main issues:
    - A large amount of NaNs in our dataset
    - The size of our dataset — We tested various methods and ultimately found the optimal solutions to address these issues without compromising the integrity of our data.
- In this project, missing values were addressed by dropping the columns that had more than 75% of missing values, as these columns were considered to be irrelevant for analysis.
- For the remaining columns with missing values, the mean of each feature was imputed. Imputing missing values with mean values is a common technique in data preprocessing as it is a simple and effective way to fill in missing data. Also, considering the size of our data, mean is the appropriate value to impute missing values with.
- However, it is important to note that imputing missing values with mean values assumes that the missing data is missing at random (MAR) and that there is no systematic bias in the missing values.
- Overall, handling missing values is an important step in ensuring the data is of high quality and is suitable for machine learning algorithms to accurately detect fraudulent transactions.

C. Data Visualization

For Data visualization, we have used a library called Plotly which provides interactive features to make your plots more interactive and also lets you add additional information which can be viewed when you hover over the plot.

We have a dedicated page in our UI application to view the Distribution of each variable category where you can view the correlation matrix of all the features of that category with respect to the target variable and can also view their distribution.

IV. Feature Engineering

Feature engineering is a crucial step in any machine learning project and involves selecting and transforming variables to create informative features for predictive modeling.

In the context of the project, feature engineering could involve identifying the most relevant variables that are likely to have an impact on the detection of fraudulent transactions.

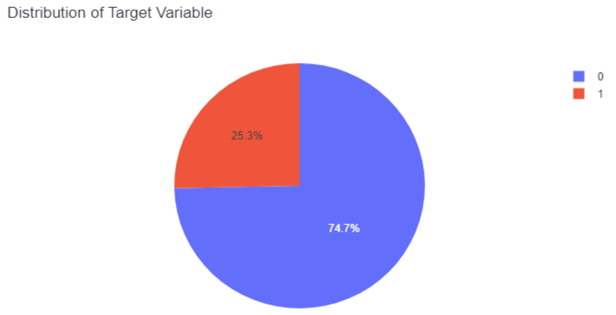
Feature engineering could also involve transforming variables or creating new ones based on domain knowledge, such as calculating the average transaction amount for a particular location or time of day.

The goal of feature engineering is to create a dataset with informative and relevant features that can help machine learning algorithms accurately predict fraudulent transactions.

A. Encoding Categorial Variables

For feature engineering, the initial task was to compute categorical variables into numerical variables, and we used encoding techniques to achieve the results.

B. Imbalanced Dataset



The pie chart above shows the distribution of our target variable and from the plot, you can see that the target variable is highly imbalanced where 75% of the value is considered as Not Defaulted and 25% of the value is defaulted.

Synthetic Minority Oversampling Technique or SMOTE for short. This technique was described by Nitesh Chawla, et al. in their 2002 paper named for the technique titled “SMOTE: Synthetic Minority Over-sampling Technique.”

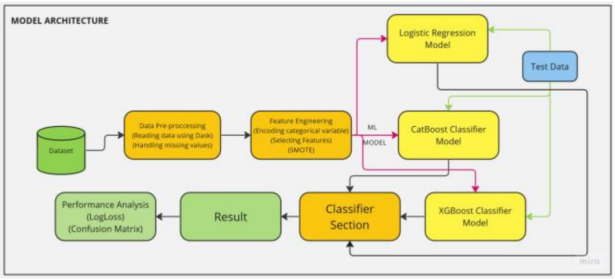
SMOTE works by selecting examples that are close to the feature space, drawing a line between the examples in the feature space, and drawing a new sample at a point along that line.

Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen, and a synthetic example is created at a randomly selected point between the two examples in the feature space.

We decided to use SMOTE technique to get rid of the problem which can arrive with the imbalanced dataset.

V. Machine Learning Models

Fraudulent Transactions can be classified as Default or Not Default, Our idea was to create a machine learning classifier model that is capable of identifying this transaction and also store the trained model and deploy it using UI so that customers can input their data and check their transaction type.



Initially, we decided to use Logistic Regression Model as our Baseline model which we can use to compare the performance of other advanced models which we were planning on using.

All the models are saved in .JSON file.

A. *Logistic Regression Model*

Logistic Regression is a popular machine learning algorithm used for binary classification problems, such as fraud detection in the context of the project. It models the probability of an event occurring, in this case, the probability of a transaction being fraudulent, as a function of the input variables or features.

In logistic regression, the output of the model is a probability value between 0 and 1, which is then thresholded to make binary predictions. The algorithm learns the parameters of the logistic regression model using a training dataset and then uses those parameters to make predictions on a test dataset.

Logistic regression has the advantage of being interpretable and easy to implement, making it a popular algorithm for many classification problems. However, it can be limited in its ability to model complex relationships between variables and may not perform as well as more advanced algorithms like XGBoost. In the context of the project, logistic regression can be used as a baseline model to compare the performance of more advanced algorithms, such as XGBoost.

B. *XGBoost Classifier Model*

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm that is widely used in predictive modeling and has been particularly successful in a number of data science competitions. It is a type of gradient-boosting algorithm that combines the predictions from multiple decision trees to make accurate and robust predictions.

XGBoost uses a gradient-boosting framework that adds new trees to the model to correct the errors made by previous trees. Each new tree is trained on the residuals of the previous trees, with the goal of minimizing the overall error of the model. Additionally, XGBoost uses regularization techniques to prevent overfitting and improve the generalization ability of the model.

XGBoost has become a popular algorithm for a variety of tasks, including classification, regression, and ranking problems. In the context of the project, XGBoost can be used as a classifier to detect fraudulent transactions based on the features that have been engineered and pre-processed. The model can be trained on a subset of the data and evaluated on a holdout set to assess its performance and tune the hyperparameters. Overall, XGBoost is a powerful algorithm that can be used to accurately detect fraudulent transactions and improve the security of financial systems.

C. *CatBoost Classifier Model*

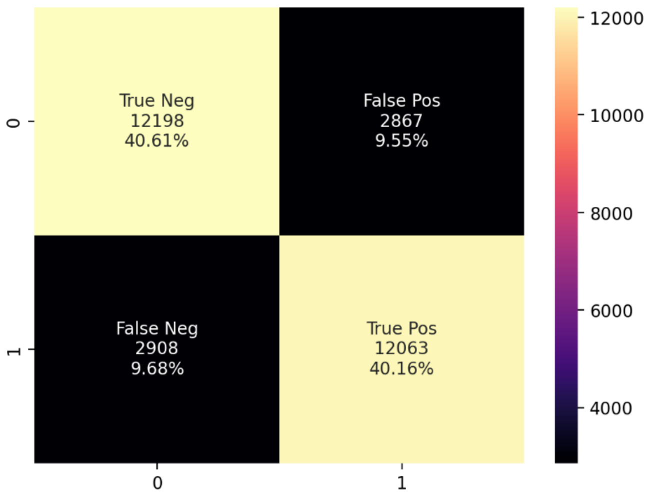
CatBoost is a state-of-the-art machine learning algorithm that is widely used for classification and regression problems. It is a gradient-boosting algorithm that works by iteratively adding new decision trees to the model, with each new tree learning to correct the errors made by the previous trees.

CatBoost has several features that make it stand out from other gradient boosting algorithms, including the ability to handle categorical features without the need for one-hot encoding, as well as the ability to handle missing values. In addition, CatBoost uses a novel approach called ordered boosting, which improves the quality of the model by selecting the most informative split points based on a ranking criterion.

In the context of the project, CatBoost can be used as a classifier to detect fraudulent transactions based on pre-processed and engineered features. The algorithm can be trained on a subset of the data and evaluated on a holdout set to assess its performance and tune the hyperparameters. Overall, CatBoost is a powerful algorithm that can provide accurate and robust predictions for a variety of classification problems, including fraud detection.

VI. Model Results

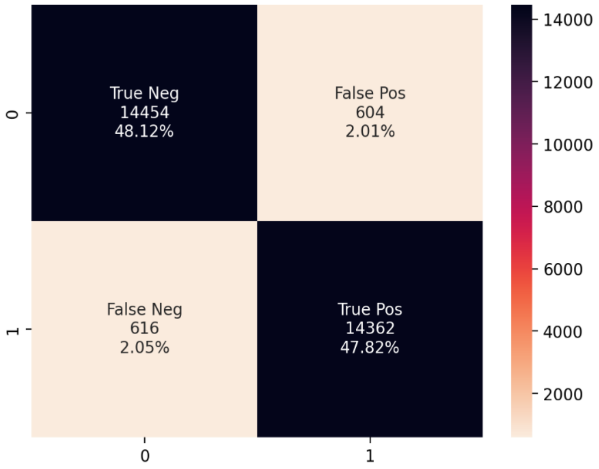
A. *Logistic Regression Model*



In this confusion matrix, we can notice that Logistic Regression predicts a True Negative value correctly only on 40.61% of the data and a True Positive value correctly for 40.16% of the data.

Logistic Regression makes the Type 1 error of 9.55% and Type 2 error of 9.68%.

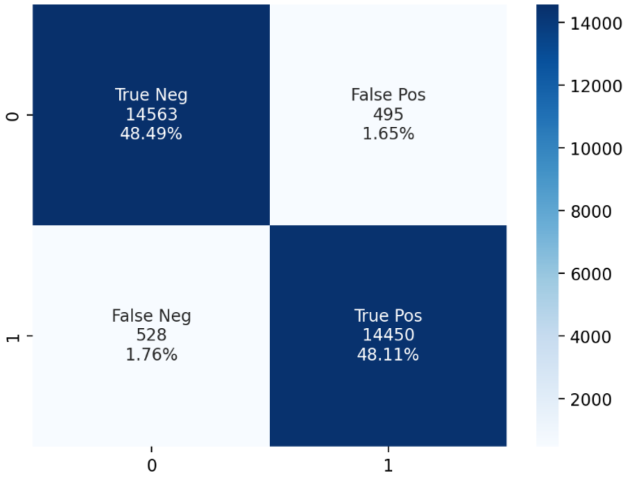
B. XGBoost Classifier Model



In this confusion matrix, we can notice that XGBoost Classifier predicts a True Negative value correctly on 48.12% of the data and a True Positive value correctly for 47.82% of the data which is way better than Logistic Regression.

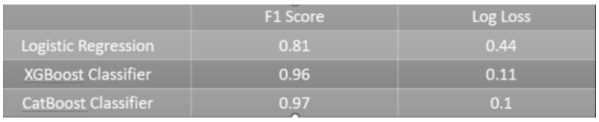
We can notice that XGBoost makes the Type 1 error of 2.01% and Type 2 error of 2.05%.

C. CatBoost Classifier Model



In this confusion matrix, we can notice that CatBoost Classifier predicts a True Negative value correctly for 48.49% of the data and a True Positive value correctly for 48.11% of the data which is slightly better than the XGBoost Classifier model and way better than the Logistic Regression model.

We can notice that CatBoost makes a Type 1 error of just 1.65% and a Type 2 error of 1.76%.



	F1 Score	Log Loss
Logistic Regression	0.81	0.44
XGBoost Classifier	0.96	0.11
CatBoost Classifier	0.97	0.1

VII. GUI FOR OUR MACHINE LEARNING MODEL

We wanted to create a UI that is capable of displaying every piece of information which is happening in the background from displaying exploratory data analysis to actually observing the working of the machine learning model. We have it all in our UI.

There are very few tools that provide the flexibility of including everything in the UI which we wanted to do. We decided to use Streamlit as it provided lots of functions and flexibility to include all sorts of information.

Streamlit is an open-source web application framework that is used to build interactive and responsive data science applications. With Streamlit, data scientists can create custom web-based interfaces for their machine-learning models and data visualizations without the need for advanced web development skills.

Streamlit is designed to make the process of building data science applications as easy and efficient as possible. It provides a simple and intuitive syntax for creating interactive widgets and visualizations and allows for rapid iteration and prototyping of new ideas.

CONCLUSION

The results showed that both XGBoost and CatBoost performed significantly better than Logistic Regression. However, CatBoost outperformed XGBoost with an accuracy of 98.1%, compared to XGBoost's accuracy of 97.7%. This indicates that CatBoost is a superior algorithm for the task of fraud detection, providing more accurate and robust predictions.

Overall, the project demonstrated the importance of feature engineering and the selection of appropriate machine learning algorithms in achieving high accuracy in fraud detection. The use of CatBoost, in particular, highlighted the power of advanced algorithms that can handle categorical features and missing values in data, providing a more accurate and robust solution.

In conclusion, the project's findings suggest that CatBoost is a powerful algorithm for detecting fraudulent transactions and should be considered as the preferred method for such tasks in future studies.

FUTURE OUTCOMES

The results of this project suggest several future outcomes that could be explored in further research. One potential avenue is to investigate the effectiveness of other advanced machine learning algorithms, such as neural networks, for fraud detection tasks. These algorithms have shown great potential in other areas of data science and may provide even greater accuracy and performance for fraud detection.

Another potential area for future research is to investigate the impact of additional data sources, such as information on the user's behavior or location, on fraud detection accuracy. Incorporating this type of data into the machine learning model could potentially improve the accuracy of fraud detection and reduce false positives.

Furthermore, the current project only focused on the binary classification of fraudulent vs. non-fraudulent transactions. Future research could explore more complex classification tasks, such as identifying the specific type or severity of fraud, which could be useful for developing more targeted fraud prevention strategies.

REFERENCES

[1] Raj, S. Benson Edwin, and A. Annie Portia. "Analysis on credit card fraud detection methods." In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), pp. 152-156. IEEE, 2011.

[2] Ghosh, S. and Reilly, D.L., 1994, January. Credit card fraud detection with a neural network. In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on (Vol. 3, pp. 621-630). IEEE.

[3] George, Ms. Shelly Shiju, and Maneeksha C. Ashok. "Fraud Detection in Credit Card using Machine Learning." In National Conference on Emerging Computer Applications, vol. 4, no. 1. 2022.

[4] Raj, S.B.E. and Portia, A.A., 2011, March. Analysis of credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.