**University of Technology Sydney**

**36106:** [**Machine Learning Algorithms and Applications**](https://canvas.uts.edu.au/courses/26202)

**Assignment 3 – Final Report**

**Report: Group Project**

**Student Name and ID:**

Kin Hang Chan - 14072995

Aibarna Singh Basnet - 24585717

Rohit Sharma - 24590960

**Date: 26/05/2023**

**Business Understanding**

The bank has been collecting transactional data from customers for 3 years. Bank transactions have been recorded in multiple csv files. Combining with the customer database, the data will form a strong base for analysis.

The bank’s newly established data scientist team has been engaged in this confidential project. The project objective is to utilise the data collected in order to bring direct value to the business or the end customers. After thorough review of the data on record, the team concludes the following business use cases could be adopted.

1. Regression Analysis:

For our regression analysis, we aimed to predict a customer's total spending expenditure for the upcoming month. By using historical transactional data from the bank, we developed a machine learning model that could accurately estimate the amount a customer is likely to spend in the following month. We employed linear regression, which allowed us to establish a relationship between various features such as account number, transaction year, category, and transaction month with the average monthly spending. Our collaborative work involved sharing code and insights on Google Collaboratory and conducting regular meetings on Google Meet to discuss progress, challenges, and potential improvements. We documented our meetings through meeting minutes to ensure transparency and facilitate effective collaboration within the team.

1. Classification Analysis:

In our classification analysis, we focused on identifying fraud transactions based on a set of predictors. We recognized the criticality of this problem as fraud can have severe consequences for both the bank and its customers. Collaboratively, we utilised a range of machine learning algorithms such as logistic regression, decision trees, and random forests to build robust fraud detection models. We evaluated various assumptions and potential ethical issues, particularly concerning privacy and fairness, ensuring that our models were trained and deployed in a responsible manner. Our communication style during team meetings and presentations was appropriate for the intended audience, using clear and concise language to explain the technical aspects of the models and their impact on the business and customers.

1. Unsupervised Learning:

In our unsupervised learning task, our goal was to cluster customers based on their age and expenditure, grouping them together based on similar behaviours and preferences. This approach allowed us to gain insights into customer segmentation and tailor marketing strategies accordingly. Collaboratively, we implemented clustering algorithms such as k-means and hierarchical clustering to identify distinct customer groups. We critically evaluated assumptions and potential ethical issues, ensuring that customer privacy and data protection were maintained throughout the process. Our communication style was adapted to the audience, using accessible language and visualisations to effectively convey our findings and recommendations to both technical and non-technical stakeholders.

This report will explain the business problems identified, how these applications will provide a solution, and how they will benefit our customers and our business.

**Data Understanding and Data Preparation**

The dataset being used comprises transactional data collected from the bank’s computer system and online banking platform over a four-year period, spanning from 2018 to 2022. It contains a vast amount of information, including 4,260,904 rows of records and 10 features. Additionally, customer data from a separate dataset, consisting of 1,000 rows and 15 features, has been merged, resulting in a comprehensive dataset with detailed transaction and customer information.

The dataset contains features such as Credit Card number, Customer Account number, Customer Transaction number, Transaction time, Category of purchase, Transaction amount, Fraud Indicator, SSN number etc.

It is seen that the given dataset's features are only marginally useful for machine learning tasks, highlighting the necessity for feature engineering approaches to extract more useful data.

During dataset preparation, several techniques were performed. The ‘dob’ column was converted into datetime format, and the age was calculated based on the current date. The ‘amt’ column and the calculated ‘age’ were appropriately converted to their respective data types. Categorical variables like 'category', 'gender', 'zip', 'trans\_wkday', 'trans\_hr', 'amt\_bin', 'city\_pop\_bin', 'dist\_bin', and 'age\_bin' were encoded to numeric values for modelling purposes. Additionally, data binning was performed on 'amt', 'city\_pop', 'dist', and 'age' to create binned variables for Classification purpose. Certain columns such as 'cc\_num\_x', 'acct\_num', 'merchant', 'job', and unnecessary columns related to transaction time were dropped from the dataset to simplify and avoid collinearity. Also, a new DataFrame called "avg\_monthly\_spending" was made to determine the average monthly spending by category. Overall, the code applied data cleaning, feature engineering, binning, encoding, and column dropping techniques to prepare the dataset for further analysis and modelling.

Furthermore, it is clear that the dataset is very unbalanced, with the majority of samples (99.88%) representing non-fraudulent transactions and the minority of samples (0.012%) representing fraudulent transactions. This extreme class disparity makes it difficult to create efficient algorithms that can reliably detect fraudulent behaviour.

Several cities stand out for having significant transaction volumes among the 726 cities in the dataset. Notably, the top cities with much larger transaction volumes than others include Brooklyn, Denver, Austin, Houston, and Dallas. With regard to the 'shopping\_pos' category, Brooklyn has the most transactions, indicating a significant customer presence. The category 'shopping\_pos' has the most transactions out of the 14 available transaction types, demonstrating a strong preference for retail purchases. Contrarily, the analysis shows that customers typically spend less on transactions involving travel, identifying a potential area for additional research and focused marketing efforts.

Interesting insights are revealed by further research of gender-based transaction patterns. According to the research, men spend a significant amount of money on gas and other transportation-related expenses, underscoring the significance of this category in their spending patterns. Females, on the other hand, show a stronger propensity to engage in food shopping activities, suggesting their attention to home necessities. Seasonal patterns in transaction activity also offer important insights. It is noteworthy that the data shows that the summer season saw the most transactions, mostly fuelled by purchases in the Food & Dining category. This finding implies that people have a stronger tendency to go out to eat or engage in food-related activities in the summer. On the other hand, Autumn shows the lowest number of transactions, suggesting that overall consumer spending may slow down during this season.

**Modelling**

After cleaning and preparing the data, the next step is the modeling phase. As we focus on three different business problems, we tried different combination of data features and different models to achieve the results.

For the first use case, in which the model is to predict a customer’s total spending for the upcoming month, a linear regression model was selected for the regression analysis. The target variable ('avg\_monthly\_spending') and features ('acct\_num', 'trans\_year', 'category', and 'trans\_month') were taken from the 'avg\_monthly\_spending' DataFrame. Using the train\_test\_split() method from sklearn, the data was divided into training and testing datasets. A random state of 42 was chosen for reproducibility, and the testing dataset size was set to 30% of the total data. The LinearRegression() class from sklearn was used to build a linear regression model. Using the fit() method, the model was trained using the training dataset. On the testing dataset, predictions were produced using the trained model's predict() method. The 'y\_pred' variable contained the expected values.

For the second use case, in which the model is to predict fraudulent transactions, a number of models were applied such as Logistic Regression, KNN, Decision Tree and Random Forest. Given the dataset is highly imbalanced, model performance will not be measured by the accuracy score as it is a misleading indicator. There are two other types of measurement indicators, namely ‘Precision’ and ‘Recall’. ‘Precision’ measures Type I error, where the model predicts fraud while the fact is no fraud. In contrast, ‘Recall’ measures Type II error, where the model predicts no fraud while the fact is fraud. Having considered that both Type I error and Type II error are costly to both the bank and customers, we adopt a measurement metric called ‘F1’, which is the harmonic mean of precision and recall. The model with the higher F1 score will be considered as the best performing model for fraud detection.

Based on this metrics, Random Forest was selected as the best performing model. The target variable (‘is\_fraud’) was saved in a variable called y and features (‘category’, ‘merchant’, ‘gender’, ‘zip’, ‘trans\_year’, ‘trans\_month’, ‘trans\_day’, ‘trans\_wkday’, ‘trans\_hr’, ‘amt\_bin’, ‘city\_pop\_bin’, ‘dist\_bin’, ‘age\_bin’) were saved in a variable called X. Training, validation and testing sets were splitted. The Random Forest model was imported from the sklearn.ensemble package. After instantiating the model, it is fit into the training set. Predictions were made based on the features in training set. Hyperparameters were tuned in order to achieve the highest possible metrics. In particular, ‘min\_samples\_leaf’, ‘n\_estimators’, ‘criterion’, and ‘max\_features’ were adjusted to reduce overfitting and achieve a higher precision and recall. Once identified the model with the optimal hyperparameters, the model was undergoing cross validation, which basically reshuffle the train and test sets according to different combinations. The number of cross validation was set to 5, meaning the data set will be split into 5 sets. One of which will be used as a test set and the remainder will be used as train set. The test set will be reshuffled accordingly for 5 times. Each time, the F1 score will be computed for each combination of train and test set. If the average F1 is comparable to the F1 scores which the model originally achieved, the model is a valid model for final testing. The model will be fit into the testing set, which is the unseen data. Predictions were made and will be compared with the class labels in test set.

For the third use case, in which the model clusters similar customer groups for marketing campaigns, it focuses on applying the K-means clustering algorithm to segment the data based on age and transaction amount. To determine the optimal number of clusters for the K-means algorithm, we evaluate the sum of squared error (SSE) for different values of K. SSE measures the within clusters variation and helps identify the number of clusters that best explain the data’s underlying structure. The MinMaxScaler from the sklearn.preprocessing package is used to make sure that the 'amt' and 'age' columns are on a comparable scale for accurate clustering. The fit() method is first used to fit the scaler on the 'amt' column. The transform() method is then used to apply the scalar to the values in the 'amt' column. The relative disparities between the values are maintained while scaling the "amt" column to a range between 0 and 1. To guarantee that both columns have consistent scales for clustering analysis, the 'age' column is scaled using the same MinMaxScaler object. The values are transformed to suit the range of 0 to 1 by first fitting the scalar to the 'age' column.

Scaling the columns is followed by the creation of a fresh instance of the KMeans algorithm with n\_clusters set to 6. The new cluster assignments are saved in the DataFrame's 'cluster' column once the algorithm has been run once more on the scaled 'age' and 'amt' columns.

**Evaluation & Deployment**

For the first use case, the mean squared error (MSE) metric was used to assess how well the regression model performed. First, sklearn.metrics was among the necessary libraries that were imported; Then, Using the mean\_squared\_error() function, the mean squared error was calculated by comparing the actual target values ('y\_test') with the predicted values ('y\_pred'). the 'mse' variable contained the MSE value; Lastly, the print() function was used to print the calculated MSE value. The regression model's MSE value came out to be 25424.876909595147.

During the deployment phase, the 'acct\_num', 'trans\_year', 'category', and 'trans\_month' columns were used as the merge keys to combine the 'avg\_monthly\_spending' DataFrame with the original 'dt' DataFrame. The DataFrame produced by the merge was put in the 'ldf' variable when it was completed using the merge() function.

The 'amt' column from the 'ldf' DataFrame was also added to the regression model as a feature. The target variable and features were modified accordingly.

The modified features and target variable were then used to retrain the regression model and evaluate its performance. After the 'amt' column was added as a feature, the MSE value was 978.5215378139854.

The average squared deviation between the anticipated average monthly spending and the actual average monthly spending can be used to interpret the mean squared error, or MSE, number that was generated from the regression analysis. A lower MSE shows that the model's forecasts are more in line with reality. The MSE results in this instance were 12012.397436053636 and 978.5215378139854, before and after the 'amt' column was added as a feature.

For the second use case, the F1 score metric was used. The best performing Random Forest model achieved the F1 score of 0.8636 on the training set and 0.8365 on the validation set. As previously mentioned, the model will then be cross-validated by a 5-fold process. The average F1 score was 0.8065, which is a little short, but still within an acceptable range. It is therefore concluded that the model is a valid one and ready for deployment.

Before putting the model into practice, the model was fit with the unseen testing set. Predictions were made and compared with the classification in the testing set. The metrics are as follows:

|  | Precision | Recall | F1 |
| --- | --- | --- | --- |
| Test set | 0.9973 | 0.7234 | 0.8385 |

Amongst all the models that were built and evaluated, one of the random forest models achieved the highest F1 score. The confusion matrix depicts the same.

Out of the 852,181 samples in the testing set, it is known that 851,176 samples (or 99.88%) are not fraud, whereas 1,005 samples (or 0.12%) are fraud. The model classified 729 fraud transactions, of which 727 transactions were true, and 2 transactions were faulty. On the contrary, the model classified 851,452 transactions as no fraud, of which 278 transactions were actually fraudulent. It demonstrates that the model has successfully identified over 72% fraudulent transactions.

As for the third use case, to evaluate the quality of clustering, it calculates the sum of squared errors (SSE) for different numbers of clusters ranging from 1 to 14. The SSE measures the overall variability within each cluster. A line plot is then generated to visualize the relationship between the number of clusters (k) and the SSE. This plot helps identify the optimal number of clusters based on the "elbow" point, where the SSE no longer decreases significantly.

The deployment phase involves integrating the model into the production environment so that it can be used to make predictions.

**Conclusion**

Based on the figures that we have predicted, we can say that the model generated predictions can be used by the bank to allocate resources and plan strategic marketing campaigns. To elaborate on the conclusion further, the analysis has provided important insights into the factors that are most strongly associated with use cases. In addition, the regression model successfully predicted customer spending and achieved a mean squared error (MSE) value of 978.52. Additionally, multiple classification models were used to evaluate fraud detection, with the Random Forest model performing the best, achieving an F1 score of 0.8385 on the testing set. K-means clustering was also performed to segment customers based on age and expenditure.

Moving forward, the next steps could include deploying the regression and fraud detection models that performed well on the testing set, integrating them into the bank's production environment. Continuing to iterate and improve the models by fine-tuning hyperparameters, exploring different algorithms, and monitoring their performance in real-world scenarios. Additionally, we can initiate new projects or explore other valuable insights or models to address different business challenges faced by the bank or its customers.

**Reflection on Team Challenges**

To ensure efficient coordination and progress throughout the project, our team promoted effective communication. We used a mix of face-to-face meetings, scheduled Google Meet sessions, and Google Colaboratory for our teamwork. Meeting in person allowed us to give work, create ideas, and have in-person talks. We used Google Meet to conduct online meetings, which allowed us to share screens, go over project updates, and address any issues or concerns. Real-time collaboration was made possible by using Google Colaboratory, which allowed us to execute code, exchange notebooks, and give feedback while simultaneously editing it. These channels of communication were essential in ensuring that everyone was on the same page, sharing updates and knowledge, and fostering a strong team atmosphere throughout the project.

Our team did run into a few obstacles while employing different communication channels, which had an effect on our teamwork and productivity. First of all, in-person gatherings were helpful for face-to-face conversations, but scheduling everyone and figuring out a time that worked for everyone to meet proved difficult because of competing obligations and varying availability. To guarantee that everyone could participate, it required careful planning and flexibility.

When using Google Meet for virtual meetings, there were occasionally technical problems like slow internet connectivity or audio/video hiccups that interfered with the conversation. Sometimes, these interruptions made it difficult to communicate clearly and led to delays in responding to requests for clarification or questions about projects. It became crucial to provide reliable internet connections and prepare for technical challenges.

The challenges of working together on Google Colaboratory were unique. Multiple team members making changes at once can often cause disputes and version control problems when updating notebooks simultaneously. Clear communication and an orderly approach were essential for organising and communicating regarding notebook changes and preventing disagreements.

Despite these difficulties, our team handled them head-on by keeping lines of communication open, outlining their expectations, and making adjustments as needed. To prevent misunderstandings, we built backup communication routes, organised regular check-ins, and documented decisions and progress. We overcame these difficulties and were able to successfully work on the project by remaining open and adaptable.

**Ethical Considerations**

Any data science project, like the one we worked on using the provided dataset, must take ethical considerations into account. We must uphold ethical standards as responsible data scientists and make sure that the project complies with applicable laws and ethical guidelines. The following are some significant ethical factors that we took into account:

Data protection and confidentiality: We understood how crucial it was to uphold these values. We were careful to handle the data with the utmost care and adhere to any confidentiality agreements in place because our tutor had given it to us. We took precautions to prevent unauthorised access to the data and only utilised it for the project's specific purposes.

Although we had access to the data, we still took into account the significance of informed permission. We recognised that the data belonged to specific people, and that it was our moral obligation to handle it in a way that met their expectations. We made sure that any analysis or conclusions drawn from the data were made without violating anyone's privacy or causing harm to the individuals concerned.

Fair and Open Data Use: We made an effort to use the data in a fair and open way. We took care to ensure the objectivity of our analysis and modelling techniques and to steer clear of any unfair or discriminatory practices. Our goal was to ensure transparency in our decision-making process and to clearly explain our processes.

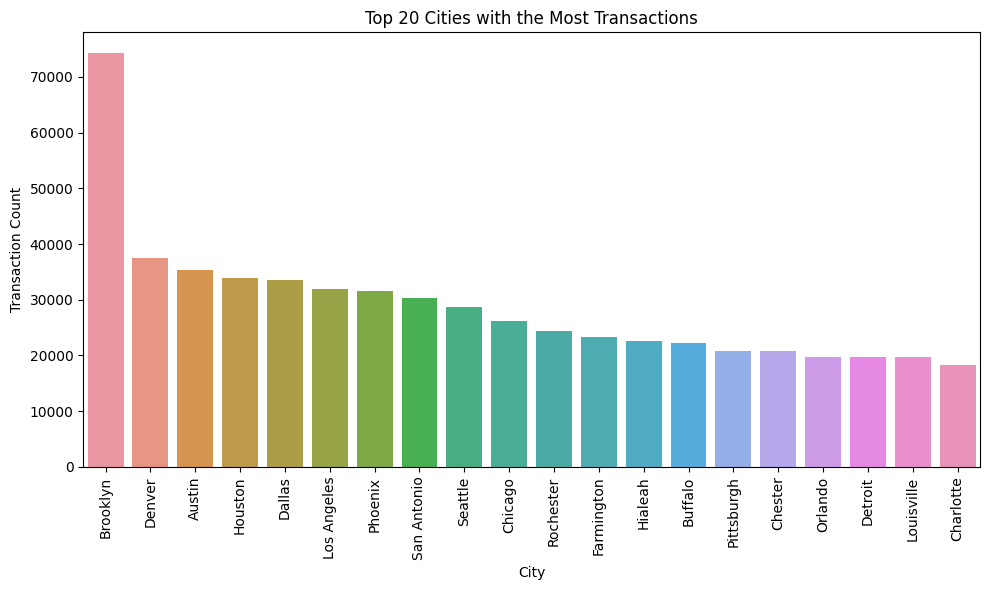
Ethical Application of Results: We understood the significance of using the findings of our study in an ethical manner. We made sure that any conclusions or forecasts drawn from the data were applied ethically and in a way that would benefit the company or its clients. We avoided any misuse or harm that might have resulted from using our findings.

We showed our dedication to undertaking responsible and ethical data science practises by taking these ethical issues into account throughout the project, thereby safeguarding the rights and interests of the people whose data was involved.

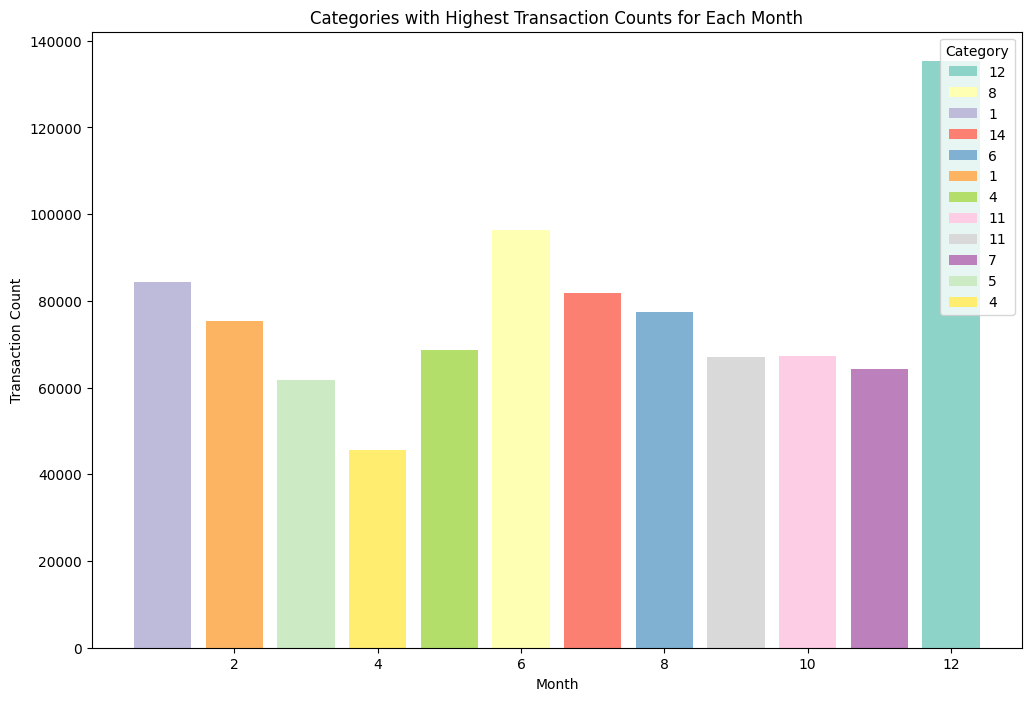
**References:**

1. Nick Hotz. (2023).<https://www.datascience-pm.com/crisp-dm-2/#:~:text=The%20CRoss%20Industry%20Standard%20Process,Is%20it%20clean%3F>
2. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
3. [Imad Dabbura](https://medium.com/@ImadPhd?source=post_page-----aa03e644b48a--------------------------------). (2018). K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>
4. Deshpande, K. (2019) Prediting my expense from historical transactions, Medium. Available at: [https://medium.com/@keyoordeshpande/prediting-my-expense-from-historical-transactions-e4d423f019b0 (Accessed: 27 May 2023).](https://medium.com/@keyoordeshpande/prediting-my-expense-from-historical-transactions-e4d423f019b0%20(Accessed:%2027%20May%202023).)
5. Building A Linear Regression Model That Predicts Retail Customer Spending. <https://www.datasource.ai/uploads/1e5660e1137e60077935fc8c1b92578e.html>
6. Scikit-learn. (2018). 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.20.3 documentation. Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

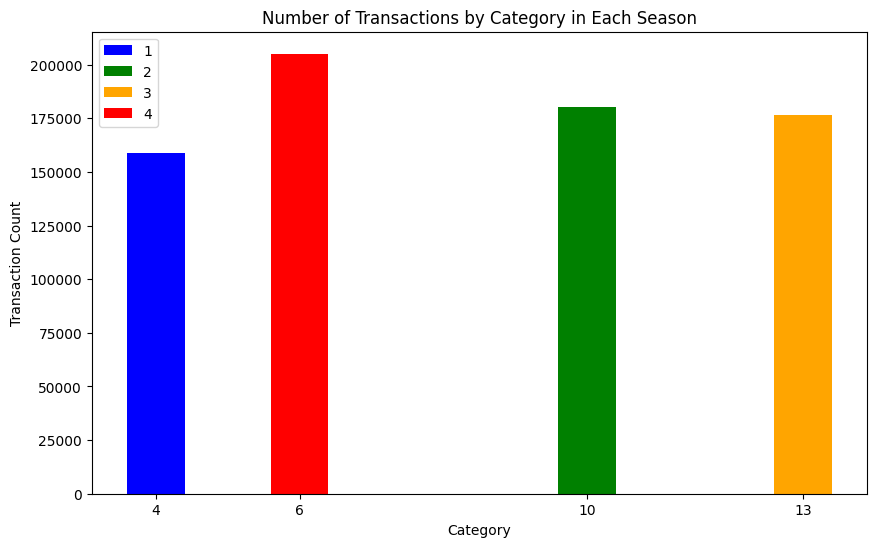
**Appendix:**



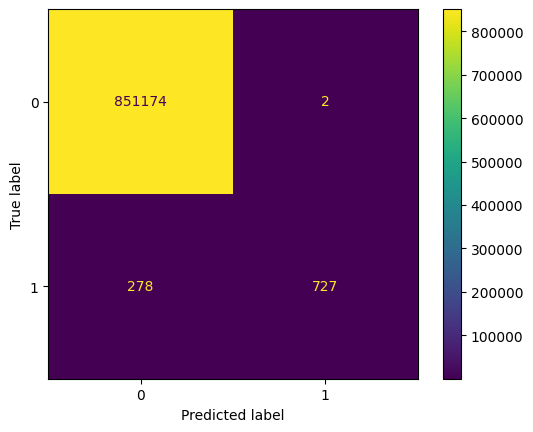
*Fig 1: Top 20 Cities with the Most Transactions*



*Fig 2: Categories with Highest Transaction Counts for Each Month*



*Fig 3: Number of Transactions by Category in Each Season*



*Fig 4: Confusion Matrix for the Random Forest Model*

In this project, three individuals were assigned to work on three different use cases, each focusing on a specific area of analysis: Regression, Classification, and Clustering.

Regression was performed by Aibarna Singh Basnet, Classification was performed by Kin Hang Chan and Clustering was performed by Rohit Sharma.