Banking Transaction (Oman Context) Classifier using ML & NLP

Submitted by: Vivan Desai

Summer Internship – July 2025

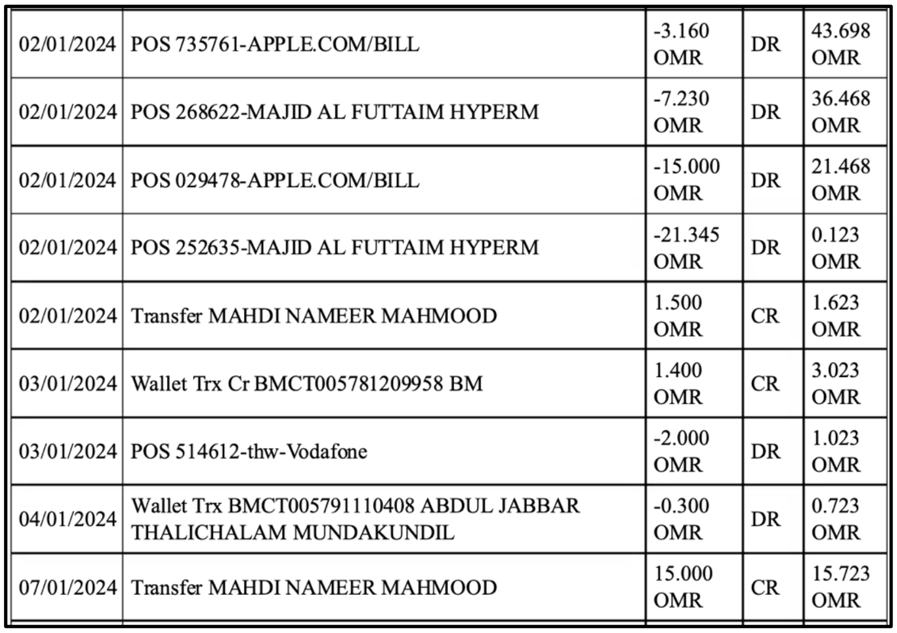
1. Introduction

In today's digital banking era, understanding customer spending behavior is essential for both financial institutions and users. In Oman, as well as many other countries, commercial banks classify transaction records using **Merchant Category Codes (MMC's)**. Although, this method does serve a function of generalizing spending habits, they also often fail to reflect the true nature of a transaction in way that is meaningful to customers and financial institutions. For instance, every customer who has used a commercial bank for spending has seen certain ambiguous transactions labelled as "Other" or even misclassified which rises a problem in misleading the customer in their own spending habits and giving financial institutions inaccurate data.

As financial services rapidly move to an online platform, commercial banks must be able to evolve and adapt to ensure customer retention and loyalty, which is extremely important in the competitive banking industry. One of the foundations, lies in improving **Personal Finance Management (PFM)**. Allowing users to view, understand their financial activity in an insightful manner. Furthermore, enhanced features associated with **PFM** will allow commercial banks to also gain insight in spending behaviors in Oman.

This project proposes a **Supervised machine learning-based transaction classification model** tailored specifically for Omani banking data. We explore different methods including **Support Vector Machine (SVM), Gradient Boosting** and **Multi-Layer Perceptron's (MLP)**, to classify banking transactions. The aim is to automatically assign categories with banking transactions. By improving the granularity and precision of current methods to interpret transactions, the proposed solution provides:

- Users to understand one owns financial activity, create budgets, and set saving goals,
- And for financial institutions to firstly, generate more accurate and insightful data and enhance customer engagement in our digitally transforming world.



02/01/2024	POS 735761-APPLE.COM/BILL	-3.160 OMR	DR	43.698 OMR
02/01/2024	POS 268622-MAJID AL FUTTAIM HYPERM	-7.230 OMR	DR	36.468 OMR
02/01/2024	POS 029478-APPLE.COM/BILL	-15.000 OMR	DR	21.468 OMR
02/01/2024	POS 252635-MAJID AL FUTTAIM HYPERM	-21.345 OMR	DR	0.123 OMR
02/01/2024	Transfer MAHDI NAMEER MAHMOOD	1.500 OMR	CR	1.623 OMR
03/01/2024	Wallet Trx Cr BMCT005781209958 BM	1.400 OMR	CR	3.023 OMR
03/01/2024	POS 514612-thw-Vodafone	-2.000 OMR	DR	1.023 OMR
04/01/2024	Wallet Trx BMCT005791110408 ABDUL JABBAR THALICHALAM MUNDAKUNDIL	-0.300 OMR	DR	0.723 OMR
07/01/2024	Transfer MAHDI NAMEER MAHMOOD	15.000 OMR	CR	15.723 OMR

Figure 1, Sample Bank Muscat Statement

Figure 1 illustrates the typical structure of banking transactions as seen in digital statements or mobile apps. These descriptions are often noisy, unstructured, and ambiguous including abbreviations, vendor codes, and inconsistent formatting. For example, descriptions like “POS 268633-MAJID AL FUTTAIM HYPERM” do not clearly indicate their category (e.g., groceries, utilities, etc.) without context. This ambiguity makes it difficult for users to understand their financial behavior and for banks to accurately categorize expenses.

This project addresses the challenge by introducing a machine learning-based classification model that transforms such raw transaction descriptions into meaningful categories, improving both personal finance management and institutional analytics.

Overtime, I believe this approach can support a feedback loop; users will be prompted and able to assign ambiguous transactions not associated to a specific category, which in turn help develop a dynamic lexicon that will continuously improve the systems precision and build a greater master data set of merchants. This model doesn’t just solve a technical problem, rather it sets the foundation for a smarter, user focused and digitalized financial world in Oman.

2. Previous Studies

This project draws inspiration from prior research on bank transaction classification, particularly the study by **Abdel-Hamid et al. (2020)** titled *"Automatic Classification of Bank Transactions Using Natural Language Processing."* Their work explored the use of lexicon-based features, word and character n-grams, and machine learning models such as Support Vector Machines (SVMs) and Gradient Boosting to classify short, noisy transaction descriptions into predefined categories.

The study demonstrated that combining domain-specific lexicons with traditional NLP techniques significantly improved model performance - particularly for financial texts where structured data is limited. Their methodology of incrementally adding features and evaluating the impact of each provided a solid framework, which this project adopts and builds upon.

While their study was conducted on European financial data, this project adapts the approach to the **Omani banking context**, incorporating localized merchant names and category definitions to enhance relevance and performance.

3. Data Generation

To develop a controlled and balanced dataset suitable for supervised classification, I generated 30,000 synthetic banking transactions using the Faker library in Python. Each transaction mimics the typical structure seen in real-world bank statement logs. The dataset includes the following fields:

- **Description** - noisy short-text with random spacing, abbreviations, and vendor-like patterns simulating real transaction narratives
- **Amount** - a numeric value indicating the size of the transaction, either negative (expense) or positive (income)
- **Date** - a timestamp reflecting when the transaction occurred
- **Category** - the target label used for supervised learning

This dataset was not randomly generated - it was carefully designed to mirror actual spending patterns in Oman, informed by both public market insights and my firsthand experience during my internship at Bank Muscat.

3.1. Banking Transaction Design

The **category structure** is arguably the most important feature, as it directly reflects how users interpret and manage their personal finances. To make this system relevant and useful in an Omani context, I designed 17 specific categories grounded in:

- **Cultural behavior** (e.g., frequent café visits in Oman)
- **Institutional data** (e.g., government payments, utility providers)
- **Public lifestyle data** (e.g., average household grocery spending)

Some examples of tailored categories include:

- **Dining and Cafes** - critical in Oman where coffee culture is prevalent; many merchants include the word "café" or "coffee"
- **Charity and Donations** - reflecting frequent religious donations, especially around events like Ramadan or Eid
- **Groceries**, **Transport**, **Utilities**, **Shopping**, **Education**, **Health**, **Entertainment**, **Rent**, and **Salary** - essential for mapping common financial activity patterns across middle-income Omani households

Working within the Retail Banking department at Bank Muscat exposed me to how financial data is structured internally, and more importantly, how customers behave when interacting with bank statements and mobile apps. This helped me identify the following:

- Common **merchant types** and **categories** users care about most
- The difference between **transaction value ranges** across income brackets
- Patterns in **salary timing** and subsequent spikes in spending
- The prevalence of **ambiguous transaction descriptions** and how they confuse users

Additionally, public data from Oman's National Centre for Statistics & Information (NCSI) and consumer surveys indicate:

- Average household grocery expenditure is **OMR 150–200/month**
- Utility bills (electricity, water) range from **OMR 25–60**
- Salary disbursements typically occur **towards the end of the month**

These insights were directly translated into **synthetic data rules**:

- **Amount and Direction**: Each category has a hand-curated transaction range. For example:
  - Dining and Cafes: (OMR 1 - 300)
  - Groceries and Supermarkets: (OMR 5 - 200)
  - Loan Payments: (OMR 500 - 200,000)

    Transactions were marked as **positive** (income) or **negative** (expense) based on predefined income-related categories (e.g., Salary) or matching keywords (e.g., "transfer from", "salary").

- **Date Distribution**:
  Based on salary timing, I introduced a **weighted distribution** so that transaction dates are more frequent towards the **1st to 10th of each month -** simulating post-paycheck spending behavior.

This structured simulation process ensures that the training data for the classifier reflects **realistic consumer financial behavior** in Oman, making the machine learning outputs more interpretable and deployable.

4. Preprocessing
To make sure the data follows a solid structure for further training, the data had to go through several steps of preprocessing.

- **Date Column**
  Using the Python Pandas library, I make sure that the values in the **date** column of the data set were converted from strings to Pandas datetime object. Through this, Panadas can manipulate the values and make them useful in our next phase for feature engineering.

- **Cleaning Description and Tokenization**
  To make the data ready for the next phase of feature engineering, I applied my cleaning function that:

  - Removes all extra white spaces,
  - Removes all non-alphabetic characters (symbols, numbers, and punctuation),
  - Converts the text to lowercase.

After cleaning the description, I performed the start of the **tokenization** process by splitting the description into small units called **tokens**. Here, **tokens** are essentially meaningful words, and will be helpful in the next step of any NLP (Natural Language Processing) pipeline in tasks such as:

- Generating **unigrams** and **bigrams** for lexicon-based feature engineering,
- Making **TF-IDF** word and character n-gram features,
- Inputting into machine learning model for classification.

5. Feature Engineering

After preprocessing the dataset into a structured and tokenized form, the following step is to extract meaningful features that will help the model classify the transactions into their manageable segments. Our main task is text classification; hence the goal is to curate features which captures semantic, lexical, and contextual patterns within these transaction descriptions. The final features will used as input variables into the supervised machine learning model.

5.1. Lexicon Features

A lexicon refers to a collection of **unigrams** (single words) and **bigrams** (pairs of consecutive words), which are frequently occurring and associated with a specific transaction category (e.g., "Lulu hypermarket" for Groceries).

Using a dictionary mapping system, I created the collection of lexicons. Grouping the cleaned transaction description by its categories and counted the frequency of each **unigram** and **bigram** in each group. To ensure only useful **unigrams** and **bigrams** are used, I filtered them by keeping the number of tokens that only appeared a minimum number of times, removing any noise.

The lexicons were then converted into a numerical feature vector by counting how many **unigrams** and **bigrams** (word and phrases) from each category's lexicon appeared in the transaction description. For example, a transaction containing the phrase "fuel station muscat" might match 2 **unigrams** and 1 **bigram** within the category of fuel and transport. Through this approach, we provide the model with specific lexical signature for each category in other words a group of words which most represent a certain category. Therefore, serves as a critical feature that provides the supervised model to make meaningful patterns.

5.2. Amount and Date

Alongside the lexical patterns within the transaction descriptions, numeric and time features such as the transaction amount and date can provide patterns for classification. For instance, certain categories for transactions will have a range of the amount in the transaction and can help the model further reveal patterns. The two features extracted from the amount column are the following:

- **Absolute Value of Amount**
  This feature captures the magnitude of the amount regardless of if it is money coming in or leaving, and its key reason is that it can be useful for determining categories that are associated with smaller or larger transactions.

- **Sign of Amount**
  A binary feature indicating whether the banking transaction is income **(1)** or expense **(0)**. This was easily engineered by whether the amount is negative or positive. This gives the model a cue if the banking transaction is a form of income which will narrow down the possible categories the transaction is associated with.

Next, the feature extract from the date column is:

- **Is End of the Month**
  Similarly, it is a binary feature indicating whether the transaction occurred towards the end of the month (specifically days 26 or later). This feature is based on the observation that most individuals receive their pay/salary towards the end of the month and may stimulate major payments such as rent or loan payments. And can also trigger, more frequent spending in the early days of a new month.

5.3. Word N-grams

While the lexicon-based features mentioned above are curated by manual and selective decisions, **word n-grams** allow to automatically select patterns from a text. These are used frequently in natural language processing (NLP) tasks.

Word n-grams is essentially a contiguous sequence of *n* words in a text. For example, the description "receipt shell oman petrol" would have:

- Unigrams (*n* = 1): "receipt", "shell", "oman", "petrol"
- Bigrams (*n* = 2): "receipt shell", "shell oman", "oman petrol"
- Trigrams (*n* = 3): "receipt shell oman", "shell oman petrol"

These features can help to identify common phrases which are associated with different categories. Using a **TF-IDF vectorization** which is in the scikit learn Python library, the word n-grams are extracted. The term **TF-IDF** stands for:

- **Term Frequency (TF)** – how often a term appears in a document
- **Inverse Document Frequency (IDF)** – how rare that term is across all documents

The result of using this via scikit learn is a matrix where each row represents a transaction, and each column is an n-gram feature. In the case of this model, the **TF-IDF** is extracting word n-grams from *n* = 1 to 4, to capture both high level semantics and specific transaction phrases.

5.4. Character N-grams

Though word n-grams can capture a sequence of words and identify phrases, it struggles with unstructured text which has typos, abbreviations or uncommon spacing. However, all of these are common in banking descriptions. Hence, character n-grams serves as a powerful tool as it works on a smaller level analyzing characters.

Character n-grams are a sequence of *n* characters extracted from a string. For example, the word "transfer":

- 3 Grams (*n* = 3): 'tra', 'ran', 'ans', 'nsf', 'sfe', 'fer'

Like the word n-grams process, using scikit learn a matrix is produced where each row represents a transaction with specific char n-grams in each column.

5.5. Feature Selection

A key step especially in this scenario is to identify which features are meaningful for an accurate classification model, and which features are not providing enough information. Selecting important features can reduce overfitting and improve generalization, and this becomes crucial when dealing with synthetic data. To assess the relevance of features I employed a **model-based feature importance** approach specifically the **gradient boosting classifier**, which is well suited as it can understand the non-linear relationships within the dataset.

The descriptive features that include **lexicon features**, **word n-grams** and **char n-grams**, are extremely high dimensional and noisy as they were engineered through the banking transaction descriptions, therefore it would be computationally heavy to individually inspect each feature. Rather, I focused on the 2 features **is_end_of _month** and **abs_amount** because both these features were designed manually during the data generation, hence may pose some variability to the performance of classification. This assumption rests on the fact that:

- Many transactions from different categories share overlapping amount ranges
- The date logic was purely random with some bias towards income categories

After training the **gradient boosting model**, the feature importance scores were the following:

- **abs_amount**: 0.9952
- **is_end_of _month**: 0.0048

These results illustrate that the **abs_amount** feature holds almost all the predictive power among the numerical features analuzed. On the other hand, the **is_end_of_month** feature has little to zero importance, which confirms the earlier assumption made about the feature. This low score can be attributed to the randomized nature of the dates within the transaction data. Hence, I retained **abs_amount** and decided to remove the **is_end_month** to ensure performance is improved.

6. Model Training

The next step after feature engineering is the supervised learning phase. The objective was to train models capable of precisely classifying banking transactions into meaningful segments, based on numerical and text-based inputs.

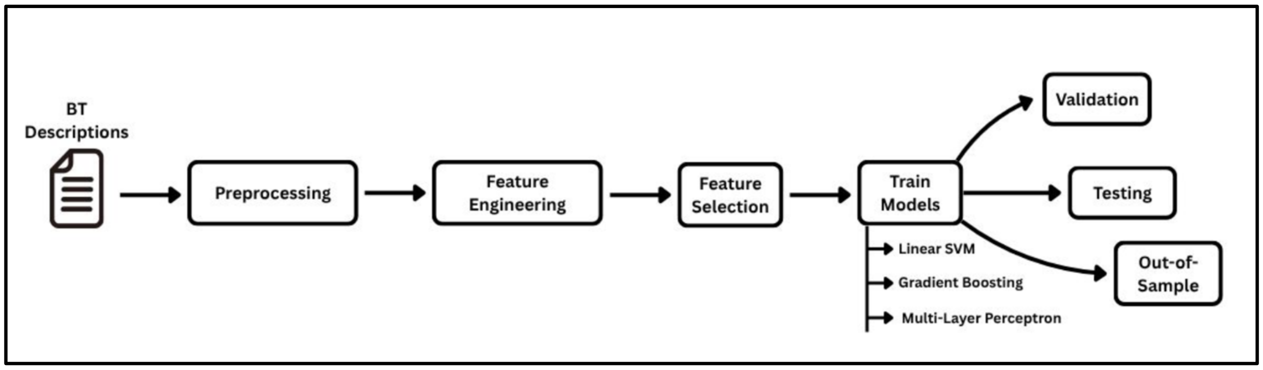


Figure 2, Workflow of Process

6.1. Train-Validation-Test Split

To ensure proper evaluation of the model I split the data into 3 parts in a 70% allocated for training, 20% for validation and 10% for the testing set.

- **Training Set**: Used to train the models
- **Validation Set**: To tune the models hyperparameters and to assess any changes that need to be made to the data
- **Testing Set**: Used for final performance of the model

6.2. Model Selection

The three classification algorithms chosen were the following:

- **Linear Support Vector Machine (SVM)**
  The use of an SVM is particularly suited for high dimensional data especially with sparse features like word and char n-grams. Therefore, for the dataset used in this scenario an SVM will work very well in generalizing patterns (Ghosh et al.)

- **Gradient Boosting Classifier (GB)**
  Gradient boosting builds an ensemble of decision trees in a sequential way, allowing to capture non-linear relationships. When it creates sequential decision trees it minimizes the loss function which is the mean squared error using gradient descent. My implementation of GB is using a small learning rate which is computationally heavy however, it minimizes the chances of overfitting the data (Kashyap).

- **Multi-Layer Perceptron (MLP)**
  This is a type of neural network capable of learning complex decision boundaries. It works by taking an input layer consisting of neurons which are essentially the features. The hidden layers are where it starts to recognize patterns. Finally, the output layer uses SoftMax which is tailored towards multi-class classification, where probabilities will be generated for each transaction and what category it's most associated with. MLP allows to find hidden patterns that are not so subtle and works well with a dataset like the banking transactions (Zhu et al.).

6.3. Final Feature Set

All three classification models were trained on the same feature set for fair comparisons, this feature set includes:

- **Lexicon Features**: the count of category specific unigrams and bigrams within the transaction descriptions
- **TF-IDF Word N-Grams**: from unigrams to 4-grams
- **TF-IDF Char N-Grams**: getting 3 to 5 characters
- **abs_amount**: scaled numeric feature indicating the magnitude of the transaction

7. Evaluation and Testing

To test the model's robustness and ability to capture non-linear relationships, I assessed their performance across 3 sets:

- **Validation Set**: from the training split
- **Testing Set**: the 20% of the synthetic data from the train-test split
- **Out-of-Sample Set**: unseen data generated with merchant category mapping

Unlike the validation and test sets, which are generated from the same distribution as the training data, the out-of-sample set introduces realistic complexity. It contains unfamiliar and familiar merchant names and category combinations, mimicking how real-world financial transactions appear when deployed in production. This is critical for assessing how well the model handles ambiguity, new vocabulary, and unstructured data.

Strong performance on the out-of-sample set shows that the model has not simply memorized patterns but has learned to generalize - an essential quality if this classifier is to be used in a real banking environment, personal finance tool, or future SaaS product

The metrics used are the ones included within the scikit learn library's classification report, this includes:

- **Accuracy**: the overall correctness of the model
- **Precision**: how many predicted categories were correct
- **Recall**: the ratio of correctly predicted categories and the all the predictions in that category
- **F1 Score**: the balance of precision and recall

The **precision** metric is by far the most important when it comes to personal finance management (PFM). In the event where suppose an application is created, the user is most concerned with accuracy. If the PFM tool incorrectly associates transactions with wrong categories, users will quickly lose trust in the tool. Therefore, paying close attention to **precision** we observe how many classified transactions were correct (Ta et al.).

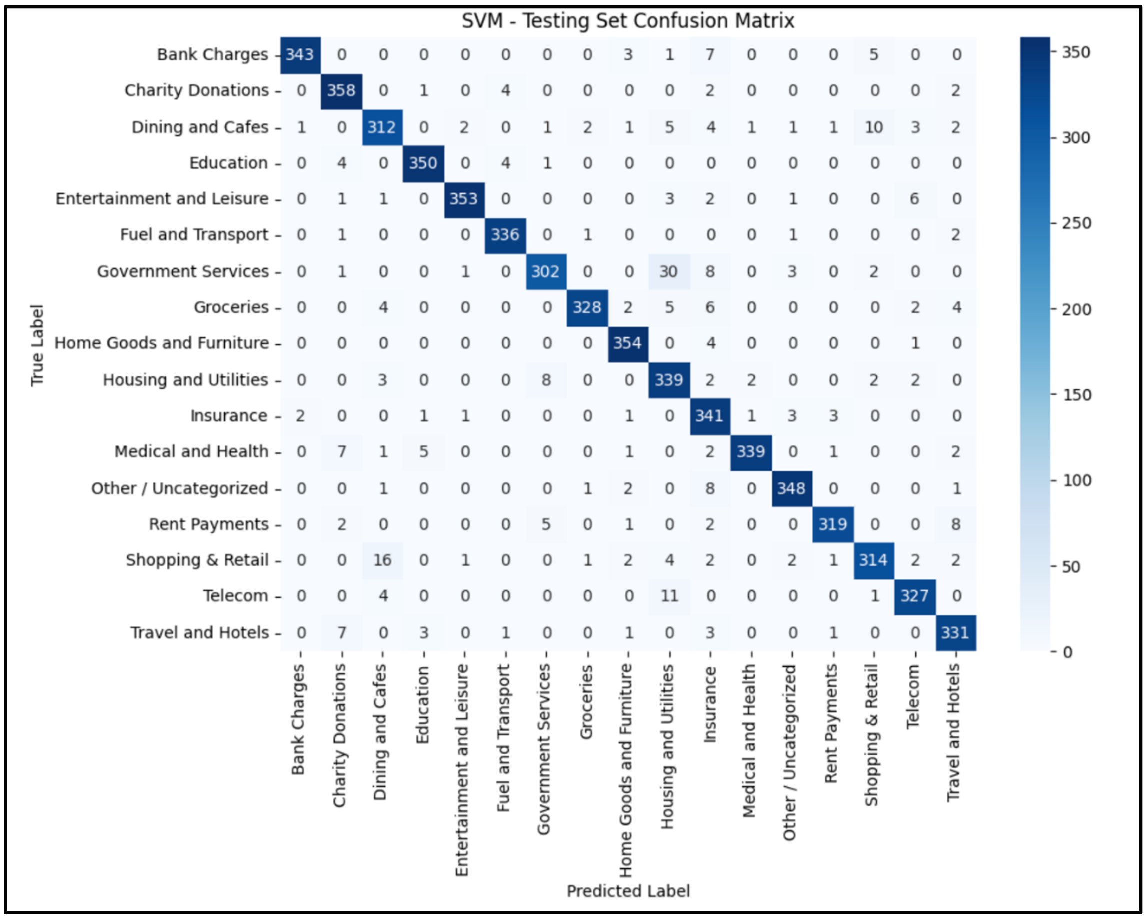
7.1. Support Vector Machine (SVM)

- **Validation Performance:**
  On the validation set, SVM achieved an accuracy of **95.3%**, corresponding to **2860 correct predictions** and **140 incorrect predictions** out of 3,000 transactions.

  - **Strengths**
    SVM demonstrated strong generalization even on a relatively small validation set. Categories like **Bank Charges**, **Medical and Health**, **Fuel and Transport**, and **Entertainment and Leisure** achieved F1-scores close to or above 0.97, showing that SVM performs exceptionally well when lexical features are distinctive and unambiguous.

  - **Weaknesses**
    The category **Housing and Utilities** had the lowest precision (0.87), which, despite a strong recall (0.96), indicates misclassifications where other classes were predicted incorrectly as Housing. This is likely due to overlapping merchant names or weak lexical distinction.

- **Testing Performance:**
  The SVM achieved an accuracy of **94.9%** on the full test set of 6,000 transactions, with **5694 correct** and **306 incorrect** predictions.



True Label \ Predicted Label	Bank Charges	Charity Donations	Dining and Cafes	Education	Entertainment and Leisure	Fuel and Transport	Government Services	Groceries	Home Goods and Furniture	Housing and Utilities	Insurance	Medical and Health	Other / Uncategorized	Rent Payments	Shopping & Retail	Telecom	Travel and Hotels
Bank Charges	343	0	0	0	0	0	0	0	3	1	7	0	0	0	5	0	0
Charity Donations	0	358	0	1	0	4	0	0	0	0	2	0	0	0	0	0	2
Dining and Cafes	1	0	312	0	2	0	1	2	1	5	4	1	1	1	10	3	2
Education	0	4	0	350	0	4	1	0	0	0	0	0	0	0	0	0	0
Entertainment and Leisure	0	1	1	0	353	0	0	0	0	3	2	0	1	0	0	6	0
Fuel and Transport	0	1	0	0	0	336	0	1	0	0	0	0	1	0	0	0	2
Government Services	0	1	0	0	1	0	302	0	0	30	8	0	3	0	2	0	0
Groceries	0	0	4	0	0	0	0	328	2	5	6	0	0	0	0	2	4
Home Goods and Furniture	0	0	0	0	0	0	0	0	354	0	4	0	0	0	0	1	0
Housing and Utilities	0	0	3	0	0	0	8	0	0	339	2	2	0	0	2	2	0
Insurance	2	0	0	1	1	0	0	0	1	0	341	1	3	3	0	0	0
Medical and Health	0	7	1	5	0	0	0	0	1	0	2	339	0	1	0	0	2
Other / Uncategorized	0	0	1	0	0	0	0	1	2	0	8	0	348	0	0	0	1
Rent Payments	0	2	0	0	0	0	5	0	1	0	2	0	0	319	0	0	8
Shopping & Retail	0	0	16	0	1	0	0	1	2	4	2	0	2	1	314	2	2
Telecom	0	0	4	0	0	0	0	0	0	11	0	0	0	0	1	327	0
Travel and Hotels	0	7	0	3	0	1	0	0	1	0	3	0	0	1	0	0	331

Figure 3, SVM Testing Set Confusion Matrix

  - **Strengths**
    Performance remained consistent across almost all categories. Categories such as **Home Goods and Furniture**, **Medical and Health**, and **Fuel and Transport** maintained very high precision and recall, confirming the model's strength in identifying non-linear decision boundaries.

  - **Weaknesses**
    Slight underperformance was again seen in **Government Services** and **Shopping & Retail** which can be seen in figure 3. While not drastic (F1: 0.91), this suggests that some

category boundaries may still be difficult to separate based on merchant descriptions alone.

- **Out-of-Sample Performance:**
  On unseen data resembling real-world structure, the SVM achieved an impressive **91.1% accuracy**, correctly predicting **9110 out of 10000 samples**.

  - **Strengths**
    The SVM performed exceptionally well in **Medical and Health** (F1: 0.97), **Education**, **Insurance**, **Groceries**, and **Travel and Hotels**. This are the categories categories with strong lexical identity and minimal ambiguity

  - **Weaknesses**
    Performance dropped in **Dining and Cafes** (F1: 0.66) and **Housing and Utilities** (F1: 0.78). These results reflect the difficulty of classifying categories with broad or overlapping merchant name structures, especially when training is based on synthetic merchant data.

7.2. Gradient Boosting (GB)

- **Validation Performance:**
  On the validation set, the GB model achieved an accuracy of **94.4%**, with **2833 correct predictions** and **167 incorrect** out of 3,000 samples.

  - **Strengths**
    GB performed exceptionally well in classifying **Medical and Health**, **Telecom**, **Entertainment and Leisure**, and **Rent Payments**. These categories had high precision and recall values, indicating the model's strength in capturing consistent lexical patterns. Telecom in particular benefits from its narrow merchant base in Oman, making it easy to learn.

  - **Weaknesses**
    Performance for **Dining and Cafes** dropped slightly with an F1-score of 0.88. This

category is harder to classify due to its broad and often ambiguous merchant names, which can overlap with unrelated businesses.

- **Testing Performance:**
  The GB model achieved an accuracy of **94.9%** on the test set (6,000 samples), with **5697 correct predictions** and **303 incorrect**.

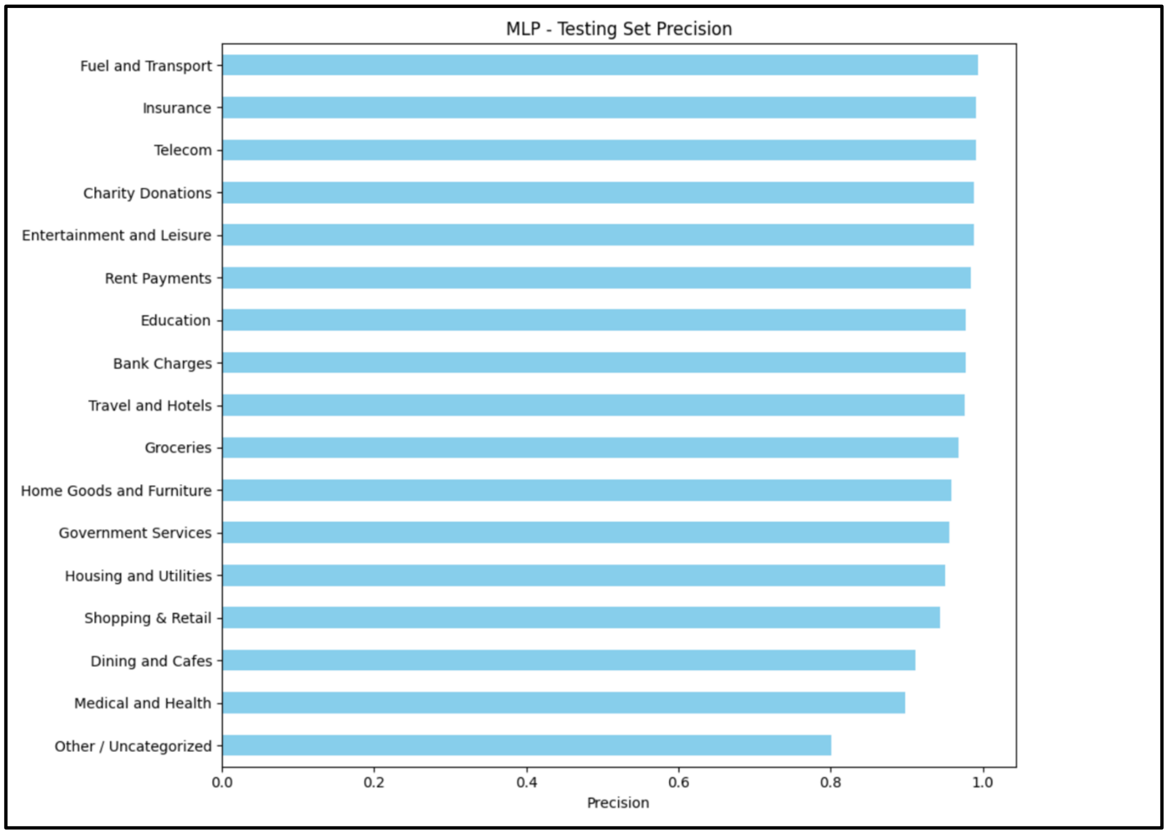


Figure 4, MLP Testing Set Precision Chart

  - **Strengths**
    Categories like **Fuel and Transport** (Precision: 0.99), **Medical and Health**, and **Rent Payments** continued to show high classification strength. This is consistent with validation, reinforcing GB's ability to capture patterns across both small and large datasets.

    - **Weaknesses**
      Dining and Cafes remained a challenge, suggesting that GB may struggle with more nuanced, ambiguous merchant descriptions, especially when lexical signals are weak or shared across categories

- **Out-of-Sample Performance:**
  In the most realistic scenario on unseen data GB's accuracy dropped to **81.7%**, with **8173 correct predictions** out of the **10,000 samples**.

    - **Strengths**
      The model performed well in **Home Goods and Furniture** (F1: 0.98), **Travel and Hotels** (0.96), **Medical and Health** (0.97), and **Insurance** (0.96). These categories are usually easier to identify due to consistent naming and distinctive language patterns.

    - **Weaknesses**
      GB struggled significantly with **Telecom**, which despite performing well earlier, fell to a precision of 1.00 but a recall of only 0.04, indicating severe underprediction. **Dining and Cafes** also performed poorly (F1: 0.46), continuing the trend. This suggests that GB overfit some lexical cues in training and couldn't generalize to new, ambiguous merchants. It also reflects the challenge of synthetic data when it doesn't capture enough real-world diversity.

7.3. Multi-Layer Perceptron (MLP)

- **Validation Performance**
  The MLP model achieved an **accuracy of 95.8%** on the validation set, with **2873 correct predictions** and **127 incorrect** out of 3,000 samples.

    - **Strengths**
      The MLP excelled in several categories, such as **Rent Payments**, **Fuel and Transport**, and **Medical and Health** with most had precision values above 0.95. It also did well in **Insurance** (Precision: 1.00) and **Telecom** (0.98), showing its ability to form flexible, non-linear representations of the data.

- **Weaknesses**
  While overall performance was strong, the model struggled slightly with the **Other / Uncategorized** category. Despite a high recall of 0.98, the lower precision (0.80) suggests that the model sometimes misclassified other transactions as "Other," likely due to overlapping lexical patterns.

- **Testing Performance**
  On the 6,000-sample test set, MLP reached an **accuracy of 95.3%**, with **5719 correct predictions** and **281 incorrect**.

  - **Strengths**
    The model continued to perform well in previously strong categories, especially **Groceries**, **Travel and Hotels**, and **Rent Payments**, suggesting good generalization from the training set. It also held its own across all 17 categories with no major breakdowns in performance.

  - **Weaknesses**
    As with validation, **Other / Uncategorized** again had lower precision. This reaffirms the issue with fuzzy lexical overlap and reinforces the need for stronger disambiguation techniques or additional contextual features.

- **Out-of-Sample Performance:**
  MLP achieved **90.5% accuracy** on the out-of-sample dataset which consists of both seen and unseen merchants.

  - **Strengths**
    Despite the unseen nature of the data, MLP showed strong generalization. **Medical and Health**, **Rent Payments**, **Groceries**, **Insurance**, and **Travel and Hotels** all achieved precision scores above 0.94. The model's flexibility likely helped it deal with minor inconsistencies and noise.

  - **Weaknesses**
    Like other models, **Dining and Cafes** (F1: 0.68) proved challenging. Its poor recall (0.53) suggests that ambiguous or generic merchant names may have been misclassified

into other categories. **Housing and Utilities** also had a lower precision (0.68), although its recall was strong. This pattern shows that while the model could often catch the category, it also sometimes incorrectly applied it.

7.4. Overall Model Comparison and Performance Insights

After evaluating all three models - Support Vector Machine (SVM), Gradient Boosting (GB), and Multi-Layer Perceptron (MLP) - across the validation, test, and out-of-sample datasets, several clear patterns emerge:

Model	**Train-Test-Validitation Split**	**Validation Set Accuracy**	**Testing Set Accuracy**	**Out-of-Sample Set Accuracy**
SVM	70/20/10	95.3%	94.9%	91.1%
GB		94.4%	94.9%	81.7%
MLP		95.8%	95.3%	90.5%

Figure 5, Overall Performances on Models

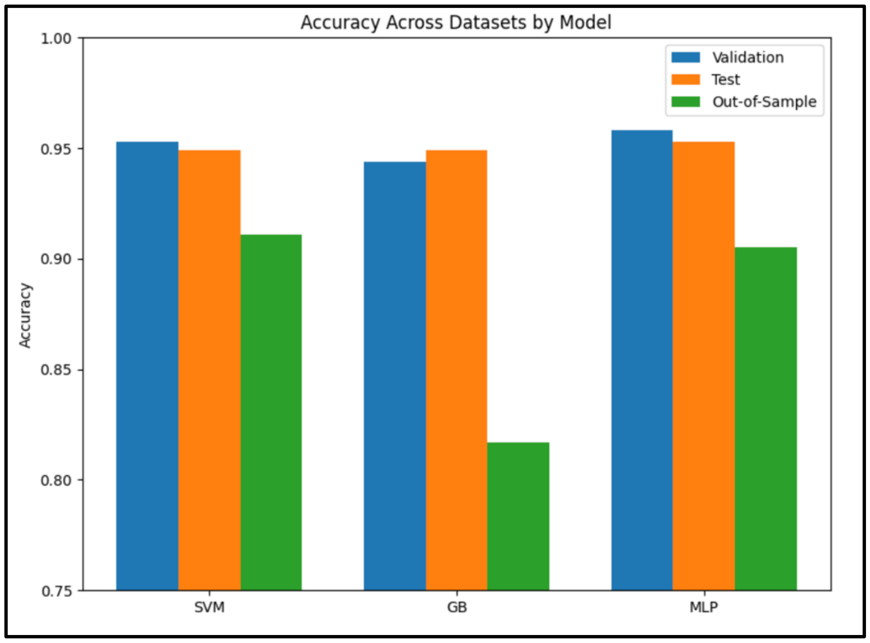


Figure 6, Accuracy Bar Chart Across Models

- **Most Accurate: MLP**
  While slightly behind SVM on out-of-sample accuracy, **MLP achieved the highest validation 95.8%** and **testing 95.3% accuracy**, making it the most consistently accurate across known data.

It was particularly strong in categories like **Medical and Health**, **Insurance**, and **Rent Payments**, likely due to its ability to learn nonlinear representations and adjust to subtle feature variations.

- **Best Generalization: SVM**
  The Support Vector Machine showed the highest accuracy on **unseen, out-of-sample data 91.1%**, meaning it generalized best despite being trained on synthetic data. Its stable performance across all three datasets demonstrates that SVM is effective in learning high-quality, transferable decision boundaries even with limited or noisy lexical overlap.

- **Highest Risk of Overfitting: GB**
  Despite competitive results on validation and test sets (94.4% and 94.9%), the **Gradient Boosting model dropped significantly to 81.7%** on the out-of-sample set and can be easily visualized in figure 6. This indicates **overfitting**, where the model learned the synthetic training data too closely and failed to generalize to real-world-like patterns. This may be attributed to GB's tendency to overfit small patterns, especially if the synthetic merchant names were too regular or repetitive.

8. Applications and Deployment Potential

The transaction classification model developed in this project offers not only strong academic value but also real-world applicability for financial institutions in Oman. Its consistent performance on unseen data highlights its potential as the foundation for a production-level system. However, I believe that using real world data is a crucial step. This process done by me has solely used synthetic data, and serves as a big constraint with biases, therefore real-world data will be a more accurate representation.

8.1. SaaS-Style Personal Finance Tool

One key direction for deployment is transforming a classifier like this into a **Software-as-a-Service (SaaS)** application that empowers users to understand their own financial activity. Such a solution could allow individuals to:

- Upload their raw bank statements (CSV or PDF)
- Automatically categorize their transactions
- Visualize monthly spending patterns

- Set budgets and financial goals based on actual behavior

This approach aligns with global trends in **Personal Finance Management (PFM)** tools and could help Omani banks offer more engaging, intelligent user experiences.

8.2. Developer-Facing API for Fintech Integration

Alternatively, the classification engine could be deployed as a **REST API**, allowing fintech developers and partner institutions to programmatically access its functionality. For instance, developers could send raw transaction descriptions and receive categorized outputs in real time.
This API-first strategy can:

- Support integration into mobile banking apps
- Enable third-party PFM startups to use the model
- Power internal bank analytics and dashboards

8.3. Internal Use by Banks

The model also has direct value within the bank's operations:

- **Enhancing transaction logs:** Improving the readability and insightfulness of customer statements
- **Risk assessment and credit scoring:** Understanding spending behavior helps build customer profiles
- **Automated reconciliation or reporting:** Categorized data is easier to aggregate and analyze

8.4. Feedback Loop for Continuous Improvement

To maintain and improve performance over time, a **user feedback mechanism** could be introduced. For example:

- Allowing users to correct misclassified transactions
- Updating the merchant lexicon dynamically
- Using real user data (anonymized) to fine-tune future models

This would enable the system to **learn continuously** and handle edge cases more effectively.

9. Bibliography

Ghosh, S., et al. “A Study on Support Vector Machine Based Linear and Non-Linear Pattern Classification.” *IEEE Xplore*, 1 Feb. 2019, pp. 24–28, https://doi.org/10.1109/ISS1.2019.8908018.

Kashyap, Piyush. “Gradient Boosting in Machine Learning: A Comprehensive Guide to Classification and Regression.” *Medium*, 7 Nov. 2024, medium.com/@piyushkashyap045/gradient-boosting-in-machine-learning-a-comprehensive-guide-to-classification-and-regression-08ce69b862a7.

“كل المؤشرات.” *Www.ncsi.gov.om*, www.ncsi.gov.om/Pages/AllIndicators.aspx.

Ta, Duc Tuyen, et al. *Specialized Text Classification: An Approach to Classifying Open Banking Transactions*. Oct. 2023, https://doi.org/10.1109/csit61576.2023.10324203. Accessed 24 June 2025.

Zhu, Xiaohang, et al. “Study on Classification Models Utilizing the Multilayer Perceptron Feature Selection and Ensemble Voting Classifier.” *Proceeding of the 2024 5th International Conference on Computer Science and Management Technology*, ACM, Oct. 2024, pp. 221–25, https://doi.org/10.1145/3708036.3708074. Accessed 21 July 2025.