**Homework 8**

**Fraud Problem**

**1a) Describe of the project**

Based on the scenario, the goal of this data science project is to build a **machine learning system** that can accurately detect **fraudulent credit card transactions**, helping the company reduce financial losses. The project starts with **raw transaction data** that includes information about the transaction time, amount, merchant, and customer demographics. Using **data science techniques**, the data will first be **cleaned**, **processed**, and **enhanced by creating new features (feature engineering)** that make fraud patterns easier to detect.

After preparing the data, several **machine learning models** will be **trained** and **evaluated** to predict the probability of a transaction being fraudulent. Because fraud is rare compared to normal transactions, special techniques will be used to handle **class imbalance** and make sure the models don't just favor the majority class. The **best-performing model** will then be selected, and the **classification threshold** will be **optimized based on business costs**, not just standard accuracy metrics. This **threshold tuning** will help balance the need to catch fraud while minimizing unnecessary costs like card reissuance and customer churn.

The project will also include a **cost-sensitive analysis** that estimates the financial impact of missed frauds and false alarms. Additionally, the project will generate **insights** about when, where, and under what conditions fraud is most likely to happen, helping the company strengthen its **fraud prevention strategies**. Overall, the project will not only create a **predictive model** but also deliver **actionable business insights** to reduce risk, save money, and improve the customer experience.

**1b) i) Dataset Description**

The **Fraud Detection** dataset is a **large-scale, publicly available** collection specifically created to study fraudulent transaction patterns in financial systems. It contains detailed structured records of credit card transactions, enriched with customer information such as demographics, address, employment data, transaction metadata, merchant details, and fraud labels. The dataset is well-suited for research and application in fraud detection, anomaly detection, and risk modeling.

1. **Accessible**: The dataset is openly available on Kaggle and can be accessed with a free Kaggle account via Python scripts, Kaggle API, or direct download.
2. **Open License**: Released under a **CC0: Public Domain** license, allowing unrestricted use, modification, and redistribution for academic and commercial purposes.
3. **Machine Processable**: The dataset is **structured in CSV format** and fully compatible with standard data processing libraries like Pandas.

Link – [Kaggle Dataset](https://www.kaggle.com/datasets/kartik2112/fraud-detection/data)

**About the dataset**

The dataset used in this project is a **simulated credit card transaction dataset** containing both legitimate and fraudulent transactions recorded between **January 1, 2019**, and **December 31, 2020**. It represents the activities of approximately **1,000 customers** making purchases across a pool of **800 merchants**. Due to the **sensitive nature of financial data and customer privacy concerns**, real-world transaction records were not used. Instead, the data was generated using the **Sparkov Data Generation tool**, available on GitHub and developed by Brandon Harris.

The simulator works by creating predefined lists of merchants, customers, and transaction categories, using the **Python "faker" library** to generate realistic but entirely synthetic customer and merchant information. For each customer profile such as adult females aged 25–50 from rural areas parameters like transaction frequency, day-of-week distributions, and amount distributions (mean and standard deviation) are configured. Based on these parameters, transactions are generated to realistically mimic customer behavior over the two-year period.

To create a more diverse and comprehensive dataset, outputs from multiple simulation profiles were combined into a standardized format. Portions of the simulator code were reviewed to better understand the generation process. The final dataset offers a realistic representation of credit card activity while ensuring **no real personal or financial information** is exposed, thus maintaining **full compliance with data privacy requirements**.

While the real-world scenario involves a massive volume of data approximately **100 million customers**, each with an average of **30,000 transactions** such large-scale datasets are not publicly available due to **privacy concerns**, **sensitive financial information**, and **proprietary restrictions**. Therefore, this study has been conducted using a **simulated dataset** that captures transactions from **1,000 customers** across **800 merchants**, covering the period from **January 2019 to December 2020**, totaling over **1.8 million transactions**. Although this dataset represents only a fraction of the full population, it was generated using behavioral profiles and transaction distributions designed to reflect realistic consumer activity. In practice, even if access to the complete 100 million customer dataset were available, any analysis would still likely be carried out on **sample subsets** due to computational constraints. Hence, this dataset serves as a **representative sample of the broader population**, enabling the development and evaluation of fraud detection techniques that can scale to larger environments.

**Data analysis**

**Dataset Overview**

* The dataset contains **1852394 rows** and **24 columns** after merging and cleaning.
* It is a **structured, machine-processable dataset** stored in **CSV format**.

A screenshot of a computer

AI-generated content may be incorrect.

**Salary Feature Engineering**

* The original dataset did not include salary information.
* An external **Bureau of Labor Statistics (BLS) API** was used to map **estimated salary** values based on **customer profession**.
* The **estimated salary** was successfully added to the dataset for each customer.

A screenshot of a computer

AI-generated content may be incorrect.

Added salary based on profession and location using Api

A screenshot of a computer

AI-generated content may be incorrect.

**Customer Information Availability**

* For each customer, detailed information is available from their **card application**, including:
  + **Address** (street, city, state, zip)
  + **Demographic information** (gender, date of birth)
  + **Employment information** (job title)
  + **Estimated salary** (added using BLS API)

A screenshot of a computer

AI-generated content may be incorrect.

**Transaction History**

* Each customer has a record of **all their transactions** over the **past 4 years**.
* Transaction details include:
  + **Transaction date and time**
  + **Amount charged**
  + **Merchant/vendor name**
* A sample of a customer's **past 20 transactions** was inspected to verify completeness.

A screenshot of a computer

AI-generated content may be incorrect.

**Fraud Detection Label**

* For approximately less than **1% of customers**, there is a **fraud flag** indicating that their credit card was **reissued** due to fraudulent use of their prior card.

A green circle with a red line

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**A total of 9,651 fraudulent transactions occurred across 999 different customers. Among them, 976 customers experienced at least one fraud event and subsequently had their credit cards reissued, while 23 customers continued using their original cards.**

**External Data Usage**

* There is **no need to buy additional data** such as census information.
* The dataset already includes **city population (city\_pop)** for each customer's zip code, providing sufficient demographic context.

**Empirical Analysis of Feature Impact on Fraud Detection(What Variables I’m using)**

Based on exploratory data analysis (EDA) and model performance testing, several variables in the dataset were found to have a **statistically meaningful and practically significant relationship** with the target variable, is\_fraud. These relationships are not speculative; they are supported by clear trends in the data, observed fraud rates across feature values, and consistent feature importance scores across multiple machine learning models.

**Confirmed Strong Influences:**

* **Transaction Amount (amt)**:  
  Fraudulent transactions are **disproportionately concentrated at lower amounts**, especially under $50. This trend was consistently confirmed by plotting fraud rate vs. amount and by feature importance scores in tree-based models. Hence, amt is a **highly predictive variable**.
* **Merchant Category (category)**:  
  Certain categories particularly **shopping\_net**, **misc\_net**, and **grocery\_pos** showed **elevated fraud rates**. This was confirmed by grouping fraud rates by category and ranking their contribution to fraud detection in the model. Therefore, merchant type is a **key risk indicator**.
* **Transaction Time (trans\_date\_trans\_time)**:  
  When transformed into **hour-of-day**, transactions flagged as fraud were found to **peak between 9 PM and 3 AM**. This confirms a behavioral pattern where fraud is more likely to occur during hours when both users and monitoring systems may be less active.
* **Geographic Distance (from lat, long, merch\_lat, merch\_long)**:  
  Fraudulent transactions often involved a **significant geographic gap** between the customer’s location and the merchant’s, suggesting card misuse. This relationship was confirmed by calculating distances and comparing average fraud rates across distance bins.
* **Demographics (gender, job, estimated\_salary, city\_pop)**:  
  These features showed **weak but consistent** differences in fraud rates when segmented. For example, fraud was slightly higher in **mid-income groups** and **urban populations** and varied slightly across occupations. However, their **individual predictive power is low**, and they gain value only when used in **combination with behavioral features**.

**Features with No Predictive Value:**

* **Identifiers (Unnamed: 0, cc\_num, trans\_num, customer\_name)**:  
  These variables serve indexing or personal identification purposes and show **no direct or indirect influence on fraud outcome**. They were confirmed to have **zero feature importance** and are excluded to avoid overfitting and privacy violations.

**1b) ii) Data Pre-processing Steps**

1. **Data Cleaning**:

* Check for and handle missing values across all features (e.g., address, job, transaction details).
* Remove any duplicate transactions if present.
* A screenshot of a computer code

  AI-generated content may be incorrect.
* A screenshot of a computer

  AI-generated content may be incorrect.

1. **Feature Engineering**:

* Create **new features** such as transaction time patterns.
* Estimate **customer age** from date of birth.
* Use city population (city\_pop) to add **location demographic strength**.
* Include **estimated salary** based on job titles.

A screenshot of a table

AI-generated content may be incorrect.

1. **Fraud Label Adjustment**:

* Review customers whose cards were reissued but final investigation showed **no fraud**.
* Adjust fraud labels if ground truth shows false positives.

A screenshot of a computer

AI-generated content may be incorrect.

After reviewing the dataset, a final\_investigation table was created to verify whether the customers whose cards were reissued had experienced confirmed fraud. The table contained 976 customers, and all customers had a fraud\_confirmed value of 1, indicating that fraud was indeed validated in every case. As a result, there were no customers with fraud\_confirmed = 0, meaning no false positives were identified. Consequently, no adjustments to the is\_fraud labels were necessary.

1. **Scaling and Normalization**:

* Normalize continuous variables like transaction amount (amt), latitude/longitude.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Imbalance Handling**:

* Since fraud is very rare (~0.5% of transactions), apply techniques such as:
  + **Oversampling** fraud examples (SMOTE)
  + **Undersampling** non-fraud examples
  + **Cost-sensitive learning** focusing more on fraud detection.

A screenshot of a computer program

AI-generated content may be incorrect.

**1b) iii) Proposed Analysis Steps**

1. **Fraud Detection Modeling:**

* Build a **supervised classification model** (e.g., Logistic Regression, Random Forest, XGBoost) to predict fraud probability.
* Emphasize **precision**, **recall**, and **F1-score**, rather than overall accuracy, to minimize financial loss from missed frauds.

**A graph with a line

AI-generated content may be incorrect.**

Based on the evaluation results, **Logistic Regression** failed to separate fraudulent from non-fraudulent transactions under extreme class imbalance conditions, achieving an AUC of approximately 0.50. **Random Forest and XGBoost** both demonstrated strong predictive performance, with Random Forest achieving an AUC of 0.975 and XGBoost achieving an AUC of 0.996. Overall, XGBoost performed best, offering a good balance between precision and recall, making it the most suitable model for minimizing financial loss due to undetected frauds.

1. **Cost-Sensitive Evaluation:**

* Incorporate business cost into evaluation:
  + Average loss per undetected fraud = **$3,000**.
  + Reissue cost per customer = **$50**.
  + Customer loss rate when reissuing unnecessarily = **0.5% cancellation**.
* Build **custom evaluation metrics** combining financial impact.

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Assumptions**

**XGBoost classification model** to predict fraud on dataset.

After prediction:

* You calculated the **Confusion Matrix**, which tells:
  + **True Positives (TP)**: Correctly detected frauds.
  + **True Negatives (TN)**: Correctly detected non-frauds.
  + **False Positives (FP)**: Wrongly flagged non-frauds as fraud (unnecessary reissue).
  + **False Negatives (FN)**: Missed frauds (fraud was not caught).

Then, **converted these confusion matrix numbers into business financial costs** using given cost assumptions.

**Cost Rules (Given in Scenario)**

|  |  |
| --- | --- |
| **Situation** | **Cost** |
| Missed Fraud (False Negative) | $3,000 loss per fraud |
| Reissue Cost (for TP + FP) | $50 per reissue |
| Customer Cancellation (for FP) | 0.5% of falsely reissued customers cancel, costing $3,000 each |

**Results**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| False Negatives (missed frauds) | 536 |
| False Positives (unnecessary reissues) | 1,488 |
| Fraud Loss | $1,608,000 |
| Reissue Cost | $192,350 |
| Customer Cancellation Cost | $22,320 |
| Total Estimated Financial Cost | **$1,822,670** |

**Interpretation**

|  |  |
| --- | --- |
| **Area** | **Impact** |
| Fraud Loss | **Most expensive** part each missed fraud costs $3k. Missing 536 frauds costs $1.6M! |
| Reissue Cost | Moderate. Every reissued card (whether fraud or not) costs $50. |
| Cancellation Loss | Smaller but still dangerous. 0.5% of unnecessary reissues lead to losing customers, costing $3k each. |

After building the **XGBoost** classification model for fraud detection, a detailed **cost-sensitive evaluation** was conducted to understand the real **financial impact** of model predictions. A **confusion matrix** was created to classify the results into **true positives** (correctly detected frauds), **true negatives** (correctly detected non-frauds), **false positives** (non-fraud cases incorrectly flagged as frauds leading to unnecessary card reissuances), and **false negatives** (missed fraud cases that could lead to major financial loss). This classification allowed us to directly tie prediction errors to **business costs**, based on the problem scenario provided.

According to the scenario, every undetected fraud case (**false negative**) results in an average **financial loss** of **$3,000**. Every card reissued, regardless of whether the fraud is confirmed, costs the company **$50** per customer. Additionally, if a customer’s card is wrongly reissued and no fraud occurred, there is a **0.5%** chance that the customer will cancel their card, resulting in an additional loss of **$3,000** per customer cancellation. Using these **business cost assumptions**, I computed the total **financial loss** from the model’s predictions.

The model results showed **536 false negatives** (missed frauds) and **1,488 false positives** (unnecessary card reissues). The **financial impact** was calculated as follows: the loss from missed frauds was **$1,608,000** (536 × $3,000), the cost of reissuing cards for all flagged customers (both **true positives** and **false positives**) was **$192,350**, and the estimated cost from customer cancellations due to **false positives** was **$22,320**. When summed together, the total estimated **financial cost** associated with the model’s current predictions amounted to approximately **$1,822,670**.

Analyzing these results reveals that the majority of the **financial loss** was driven by **missed frauds**, not by card reissuance or customer cancellations. Missing fraudulent transactions proved to be significantly more costly than mistakenly flagging legitimate customers. While **card reissue** and **cancellation costs** do impact finances, they are comparatively small relative to the severe loss caused by each undetected fraudulent transaction.

This detailed **cost-sensitive evaluation** clearly demonstrates that, in the context of fraud detection, maximizing **recall** (the ability to detect as many fraud cases as possible) is far more important than maximizing overall **accuracy** or **precision**. Although issuing replacement cards for **false positives** incurs some cost, it is relatively minor compared to the **financial damage** caused by missing fraud. Therefore, the business goal should not be simply to optimize model **accuracy**, but rather to **minimize the total financial cost**, which requires a careful trade-off between catching more frauds and tolerating a manageable number of **false alarms**.

Based on the evaluation, it is evident that the default **decision threshold** of **0.5** used by the **XGBoost** model is not optimal for minimizing **financial loss**. Even though **XGBoost** performed well compared to other models, the high number of **false negatives** still leads to unacceptable levels of **business risk**. A next logical step would be to tune the **decision threshold** by lowering it below **0.5**, thereby increasing **recall** and catching more fraud cases. Although this would slightly increase the number of **false positives** and **card reissues**, the overall **financial impact** would likely improve substantially by preventing large fraud losses.

In conclusion, the evaluation highlights the critical importance of incorporating real **business costs** into **model evaluation** and decision-making. Rather than optimizing for standard **performance metrics** alone, a **cost-sensitive framework** enables smarter and more financially sound fraud detection strategies, helping businesses **minimize overall losses** effectively.

1. **False Positive Management:**

* Design thresholds to **minimize unnecessary card reissuance**, since false alarms cause reissue cost and potential customer churn.

A screenshot of a computer

AI-generated content may be incorrect.

After completing the cost-sensitive evaluation at the default threshold of 0.5, it was observed that the total estimated financial cost was excessively high due to a significant number of **false negatives** (missed frauds) and **false positives** (unnecessary reissues). To address this, a **threshold tuning** procedure was performed, where the classification threshold was systematically varied between 0.1 and 0.9 in steps of 0.01. For each threshold, a confusion matrix was calculated, and the total financial cost was computed by summing the loss from undetected frauds, the cost of card reissuance, and the cost resulting from customer cancellations.

The tuning process revealed that the optimal decision threshold that minimizes the total financial cost is approximately **0.15**. At this threshold, the minimum estimated total financial cost was approximately **$1,213,365**, which is significantly lower than the cost observed at the standard 0.5 threshold. By lowering the threshold from 0.5 to 0.15, the model becomes more sensitive to detecting frauds, thus increasing **recall** and reducing the number of missed frauds. Although this adjustment slightly increases the number of **false positives** (and therefore the cost of unnecessary card reissues), the overall financial loss is substantially reduced, demonstrating the effectiveness of using cost-sensitive threshold optimization in fraud detection.

This approach emphasizes that, rather than relying on traditional classification metrics such as **accuracy** alone, it is crucial to align the model’s behavior with business goals through **cost-sensitive evaluation**. In fraud detection, minimizing financial loss is far more important than maximizing conventional accuracy or minimizing **false positives** in isolation. The threshold tuning technique ensures that false alarms (unnecessary reissuances) are tolerated only to the extent that they contribute to a greater reduction in total financial risk, striking an optimal balance between fraud detection and operational costs.

In conclusion, fine-tuning the classification threshold based on **business-driven costs** is a critical step in designing effective fraud detection systems. By adjusting the model’s sensitivity to fraud probabilities, companies can drastically lower their financial exposure while maintaining an acceptable customer experience. This reinforces the importance of building cost-aware machine learning solutions in real-world financial applications.

1. **Trend Analysis:**

* Analyze fraud transaction patterns by:
  + Time of day
  + Merchant category
  + Transaction amount ranges
  + Geographical regions

**Time of Day**

**A graph with a line graph

AI-generated content may be incorrect.**

The analysis of fraud transaction patterns across different times of the day revealed a clear trend. Fraudulent activities were observed to be heavily concentrated during late-night and early-morning hours, specifically between **9 PM and 3 AM**. The histogram showed a distinct spike in the number of fraud transactions occurring during these off-peak hours, with **relatively lower** activity during the daytime. This trend suggests that fraudsters may intentionally target times when banks, merchants, and customers are **less vigilant**, and when real-time monitoring or manual interventions are limited. Such insights highlight the need for strengthening fraud detection systems with automated alerts during late-night periods, ensuring that even off-peak transaction windows are adequately protected.

**Merchant category**

A graph with blue squares

AI-generated content may be incorrect.

The analysis of fraud transaction patterns across different merchant categories revealed that fraudulent activities were most heavily concentrated in a few specific sectors. The top two categories where fraud was most frequent were **Grocery Point of Sale (grocery\_pos)** and **Online Shopping Networks (shopping\_net)**, both recording similarly high fraud counts. Other notable categories included **Miscellaneous Online Network Purchases (misc\_net)**, **Retail Shopping Point of Sale (shopping\_pos)**, and **Gas Stations and Transportation Services (gas\_transport)**. These results suggest that fraudsters tend to target categories where transactions are typically frequent, rapid, and amounts can vary widely, allowing fraudulent activities to blend into normal customer behavior patterns. Furthermore, the high occurrence of fraud in online shopping platforms highlights the vulnerability of e-commerce systems to fraudulent attacks, emphasizing the need for robust fraud detection mechanisms and multi-factor authentication protocols. Targeted monitoring of transactions in these high-risk merchant categories could substantially enhance early detection and prevention of fraud.

**Transaction amount ranges**

A white grid with black text

AI-generated content may be incorrect.

The analysis of fraud transactions across different transaction amount ranges revealed a strong concentration of fraudulent activities within the lowest monetary band. Specifically, most fraud cases occurred in the **$0–$50** range, with very few fraudulent transactions observed in higher amount brackets such as **$50–$100**, **$100–$500**, **$500–$1000**, **$1000–$5000**, or above **$5000**. This trend suggests that fraudsters often prefer executing low-value transactions, likely to avoid immediate detection by automated systems or customer notifications. Small-dollar fraud attempts can often fly under the radar of banks' alert thresholds or may not trigger customer suspicion, allowing fraudsters to either test stolen card information or to execute multiple small fraudulent purchases over time. Recognizing this behavior pattern is crucial, as it underscores the need for fraud detection models to pay close attention even to low-value transactions, which might traditionally be considered low-risk in conventional banking systems.

**Geographical regions**

**A map of the united states

AI-generated content may be incorrect.**

The geographical analysis of fraud transactions, visualized using a scatter plot of latitude versus longitude, revealed that fraudulent activities are predominantly clustered within certain regions. The majority of fraud incidents appear to be concentrated in areas corresponding to densely populated and economically active parts of the United States, particularly in the eastern and central regions. There is a noticeable aggregation of fraud points in locations that likely represent major metropolitan hubs, suggesting that fraudsters may prefer operating in regions with high transaction volumes, where fraudulent activities can blend more easily with legitimate customer behavior. The geographical spread also highlights a few isolated instances of fraud in less populated areas, but these are considerably rare compared to the clusters seen in urban zones. These insights emphasize the importance of incorporating regional transaction patterns and location-based risk scoring into fraud detection models to improve the accuracy of identifying suspicious activities.

**1c) Workflow**

**A diagram of a company

AI-generated content may be incorrect.**

**Figure 1c**

A purple rectangle with a white border

AI-generated content may be incorrect. - Data Input/OutputA yellow rectangle with a yellow border

AI-generated content may be incorrect. - FunctionsA green and yellow wavy shape

AI-generated content may be incorrect.- Parameters(optional)

From the figure 1c, the fraud detection workflow begins with **raw credit card transaction** **data,** containing attributes such as transaction amount, merchant, customer details, and fraud label indicators. The first function in the pipeline is **clean\_data(),** which removes **duplicate records** (parameter: keep='first'), handles **missing values** (parameter: missing\_strategy='drop'), and corrects column data types such as converting timestamps to datetime format (parameter: type\_casting). The output of this function is a clean and consistent dataset ready for feature generation.

The cleaned data is passed to the **feature\_engineering()** function, where additional informative features are created. This includes extracting the transaction hour from timestamps (parameter: timestamp\_column='trans\_date\_trans\_time'), estimating customer age from **date of birth** (parameter: dob\_column='dob'), **mapping estimated salaries from job titles** (parameter: job\_salary\_mapping dictionary), and categorizing transaction amounts into defined bins (parameters: bins and labels). This function outputs an **enhanced dataset** with richer features for downstream modeling.

The **enhanced dataset** is then passed into **scale\_normalize\_data(),** where continuous numerical variables such as transaction amount (amt), latitude, and longitude are scaled using either **Min-Max scaling or Standardization** (parameter: scaling\_method='min-max'). The resulting scaled and normalized data ensures uniformity across feature ranges.

Since fraud represents a small fraction of total transactions, imbalance handling is applied through the **balance\_classes()** function, which uses techniques like **SMOTE oversampling and random undersampling** (parameter: sampling\_strategy='auto'). This results in a balanced dataset, where the fraud and non-fraud classes are more equally represented.

The balanced data is then divided into training and testing sets using the split\_dataset() function, typically with a **70:30** ratio (**holdout method**) (parameter: random\_state=42). The training data is used in train\_models(), where algorithms like **Logistic Regression, Random Forest, and XGBoost** are trained using optimized hyperparameters (parameters: model\_type, n\_estimators, max\_depth).

Predictions generated by the models are evaluated through the **evaluate\_model()** function, which compares predicted labels to **true labels** (parameter: y\_true) using evaluation metrics such as **precision, recall, F1-score, AUC, and a confusion matrix**.

The evaluation scores from all models are then passed to the **select\_best\_model()** function, where the model with the **highest metric (typically F1-score or AUC)** is selected. The selected best trained model is then subjected to **threshold\_tuning\_and\_cost\_sensitive\_evaluation**(), where different classification thresholds are tested (parameter: threshold\_range = np.arange(0.1, 0.9, 0.01)) to minimize total financial loss, factoring in fraud losses, card reissuance costs, and customer churn penalties.

Finally, the optimized model performance and the corresponding financial cost analysis are summarized, and fraud trend patterns are analyzed by the **fraud\_trend\_analysis()** function, exploring trends across transaction time, merchant categories, transaction amounts, and geographical regions. The workflow concludes with visualizations and interpretability reports based on detected fraud patterns.

**1d. Expected Outputs of the Project**

The project is expected to deliver a fully trained and validated machine learning model capable of accurately identifying fraudulent credit card transactions. By leveraging algorithms such as **Logistic Regression, Random Forest, and XGBoost**, the system will be able to differentiate between legitimate and suspicious activities with a strong focus on maximizing **recall, precision, and F1-score**. This will enable the company to significantly reduce the financial losses associated with undetected fraud, which can be extremely costly if fraudulent transactions are not intercepted promptly.

In addition to building the **fraud detection model**, the project will generate an optimized **classification threshold** based on cost-sensitive evaluation. Rather than relying on a default 0.5 threshold, the system will recommend a specific **cutoff point** that strategically balances the need to detect fraud early while minimizing the operational costs of **unnecessary card reissuances** and the risk of alienating genuine customers. This optimization ensures that fraud prevention efforts are both financially and operationally sustainable.

As another major output, the project will provide comprehensive insights into fraud patterns uncovered during **analysis**. These patterns will highlight when fraudulent activities are most likely to occur (for example, late at night or early in the morning), which merchant categories are most often associated with **fraudulent transactions**, the typical transaction amounts involved in fraud cases, and the **geographic regions** where fraud tends to **cluster**. Such insights will empower the business to tailor its monitoring systems more intelligently and prioritize resources toward the highest-risk areas.

The project will also generate detailed financial cost estimation reports that quantify the monetary impact of various fraud detection strategies. These reports will help business stakeholders understand the trade-offs between more aggressive fraud detection (which may inconvenience some customers) and more lenient approaches (which could allow more fraud losses). Armed with this information, the company can make informed decisions that maximize profitability and minimize unnecessary customer churn.

Finally, the project will culminate in a set of actionable **business recommendations**. These recommendations will not only suggest technical adjustments such as adjusting detection thresholds or retraining models periodically but also operational improvements, such as **refining customer communication** strategies when fraud is suspected and designing proactive engagement programs to retain customers who are falsely flagged. **Overall**, the outcomes of the project will strengthen the company’s fraud prevention ecosystem, improve operational efficiency, enhance customer satisfaction, and directly contribute to reducing financial losses and protecting the company’s reputation.

**Conclusion**

**Data Overview and How It Was Used in the Project**

The dataset provided a **comprehensive, structured view** of customer transactions and personal attributes over a four-year period. Here's how each component was leveraged during analysis and modeling:

1. **Customer Profile Information**

“For each customer, there is detailed information from their card application about their address, salary and employment, and demographic.”

I extracted and utilized the following:

* **Address-related data** (street, city, state, zip, lat/long) was used to assess geographic behavior and calculate distance between customer and merchant often a red flag in fraud.
* **Employment and job title** were used to map **estimated salary**, using a public API (e.g., Bureau of Labor Statistics) to enrich the dataset where salary was missing.
* **Demographic indicators** such as gender, city population, and inferred age (from dob) were used for profiling and feature engineering.

1. **Transaction Records**

“For each customer, there is a record of all their transactions (date, charges, and vendor) for the last 4 years.”

* Parsed **transaction timestamps** to derive temporal features like **hour, day of week, and seasonal trends**.
* Analyzed **merchant categories** and **amounts** to find fraud-prone patterns.
* Filtered and visualized customer-level transaction histories to identify **spending behavior** over time.

1. **Reissued Cards After Fraud**

“For 1% of customers, there is a flag that their credit card was reissued because of fraudulent use of their prior card.”

* Identified **customers with reissued cards** by tracking fraud flags (is\_fraud = 1) and repeated card numbers or new transaction patterns.
* Analyzed the impact of **false positives**, where cards were reissued but no actual fraud occurred.
* Quantified costs associated with **card reissuance** and **customer churn** for misclassified cases.

1. **External Data Options (e.g., Census)**

“You can buy additional data, like census data for any zip code.”

In our case:

* We did **not need to purchase external data**, as the dataset already included **city population**, **zip codes**, and **estimated salary**.
* This allowed us to simulate real-world customer profiling without relying on additional data sources.

**Cost Analysis**

In this fraud detection scenario, the financial implications are significant. When a fraudulent transaction goes undetected, the company incurs an average loss of **$3,000 per case**. On the other hand, proactively reissuing a customer's credit card after suspected fraud incurs a **cost of $50 per reissue**. However, this preventive measure has its own risk if the card is reissued unnecessarily (i.e., in the case of a false positive), **0.5% of customers tend to cancel their cards**, potentially leading to long-term customer churn and revenue loss. These cost factors directly informed my model evaluation and threshold tuning strategies to balance precision (to avoid missed frauds) with minimizing false positives (to reduce unnecessary reissues and churn).

**Overcoming the Challenges**

To handle the challenge of **data volume** where real-world systems involve over **100 million customers** with an average of **30,000 transactions each**, I worked with a well-structured, representative **sample dataset** of ~1.85 million transactions across 1,000 customers. This allowed for scalable modeling and testing while preserving realistic transaction patterns. For the second challenge, where **some credit cards were reissued unnecessarily** (false positives), I used a **cost-sensitive evaluation framework**. This approach factored in the financial impact of missed frauds ($3,000 loss) versus unnecessary reissuance ($50 cost and 0.5% customer churn). I further applied **threshold tuning** and focused on **precision and recall** minimizing such errors, ensuring that cards were only reissued when the probability of fraud was truly high. Together, these solutions enabled effective, scalable fraud detection without compromising on customer trust or operational efficiency.