TVS CREDIT

e.p.i.c 6

ENRICH | PERFORM | INNOVATE | CHALLENGE

Adityasingh031807 & IIT Mandi

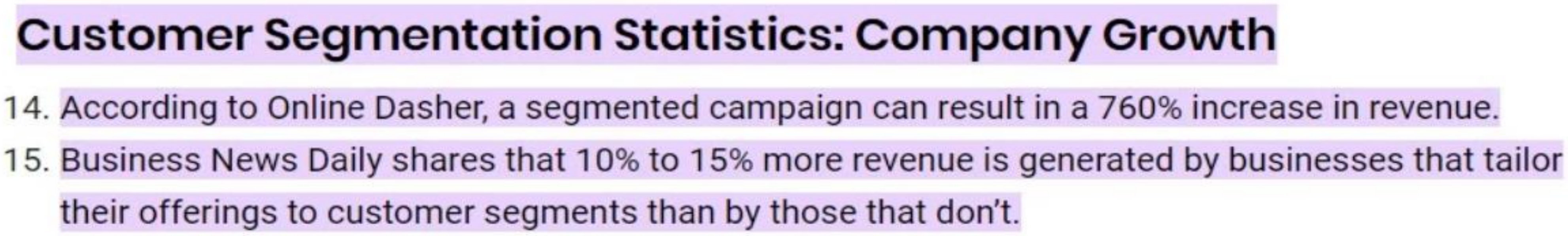
Aditya Singh | Pranav Shirbhate

Classification: Public

🏔 Problem Statement Tackled

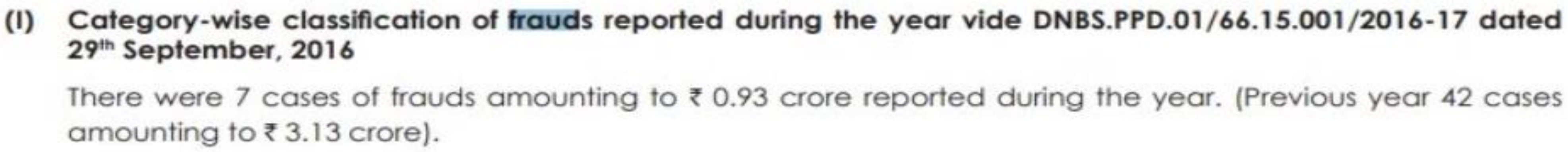
"How can TVS Credit leverage machine learning models to optimize customer segmentation, improve loan defaulter detection, enhance cashflow analysis, and detect fraudulent transactions, while also identifying opportunities for effective cross-selling, in order to maximize customer engagement and minimize financial risk?"

🤝 Succession/transition plans



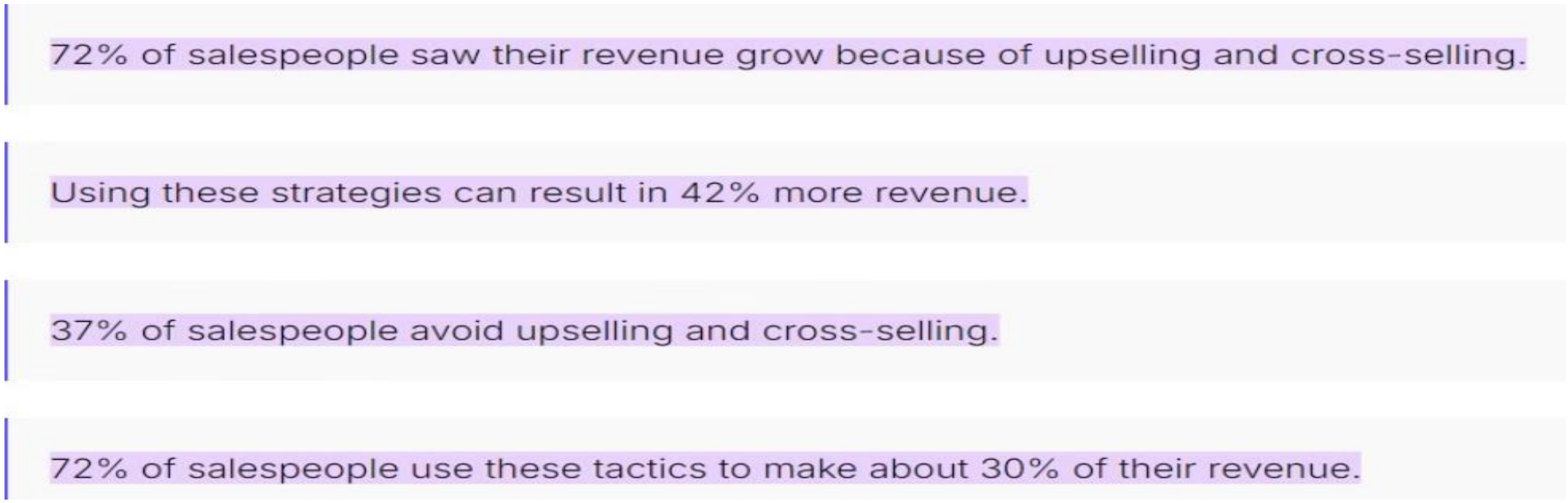
Customer Segmentation Statistics: Company Growth

14. According to Online Dasher, a segmented campaign can result in a 760% increase in revenue.
15. Business News Daily shares that 10% to 15% more revenue is generated by businesses that tailor their offerings to customer segments than by those that don't.



(I) Category-wise classification of frauds reported during the year vide DNBS.PPD.01/66.15.001/2016-17 dated 29th September, 2016

There were 7 cases of frauds amounting to ₹ 0.93 crore reported during the year. (Previous year 42 cases amounting to ₹ 3.13 crore).



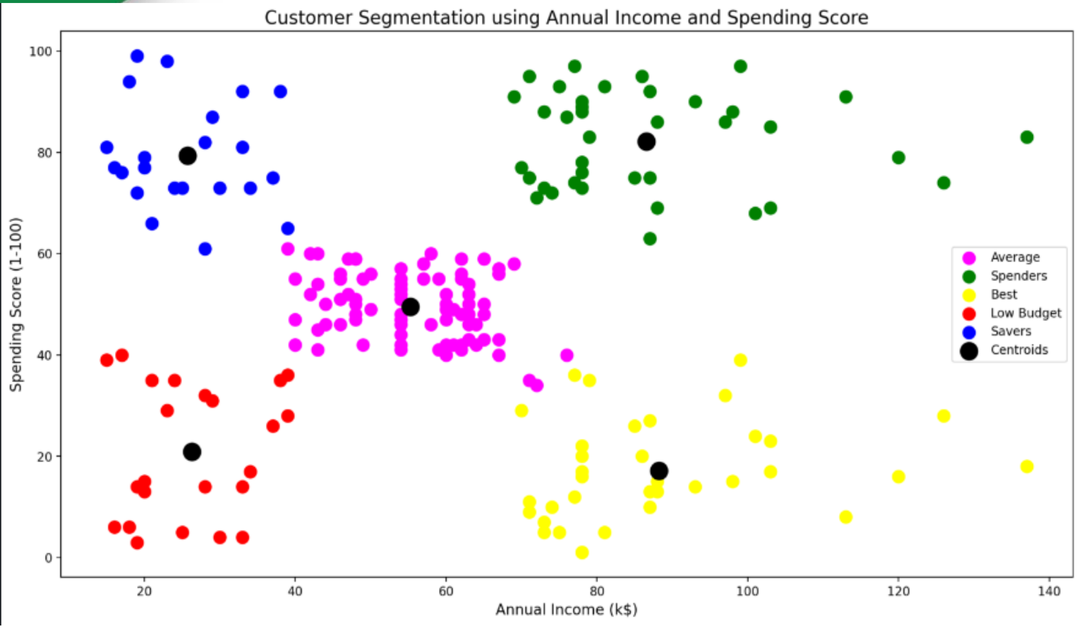
> 72% of salespeople saw their revenue grow because of upselling and cross-selling.

> Using these strategies can result in 42% more revenue.

> 37% of salespeople avoid upselling and cross-selling.

> 72% of salespeople use these tactics to make about 30% of their revenue.

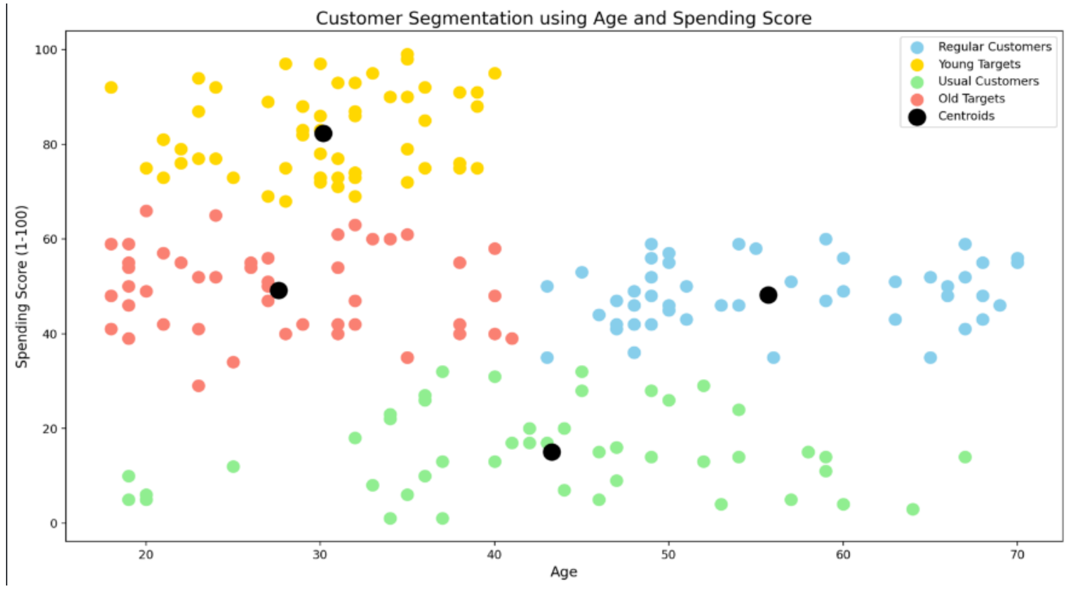
Our Solution (Model Explanation)

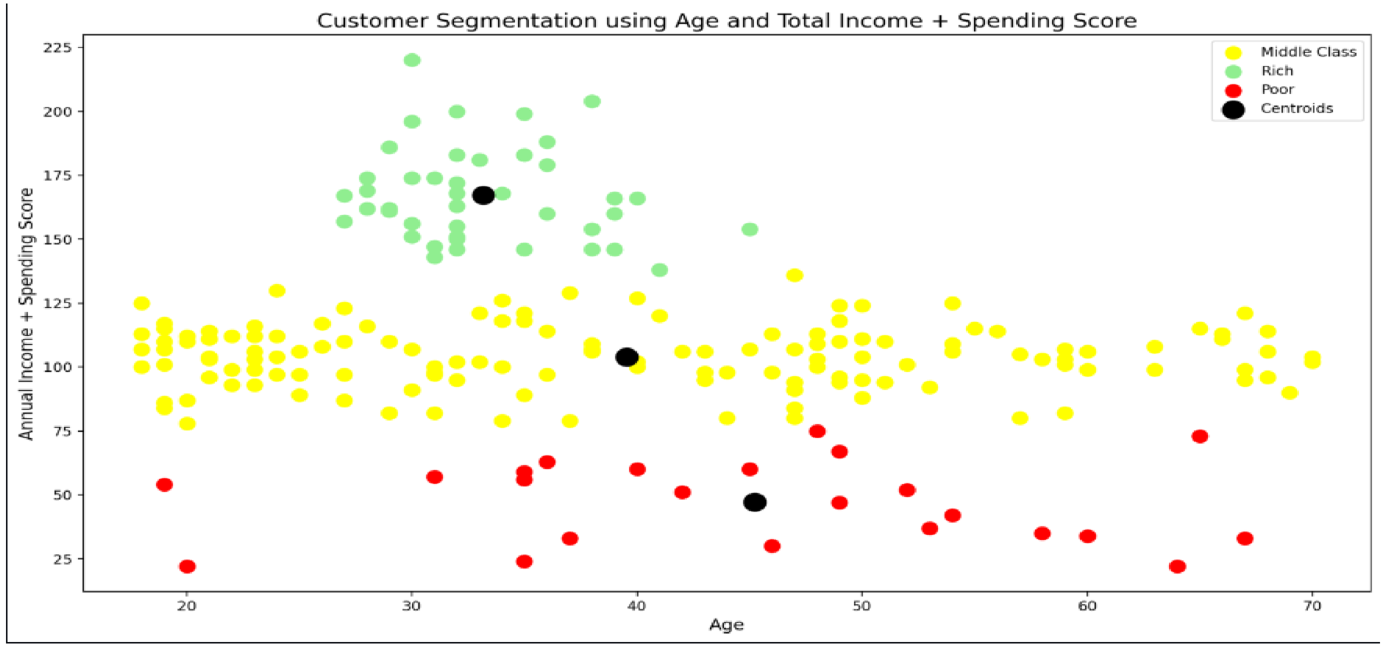

Customer Segmentation using Annual Income and Spending Score

Customer Segmentation Model :

Dataset contains the following variables:

- **Gender**: Can be predicted using name or accessed directly.
- **Age**: Calculated from the date of birth.
- **Annual Income**: Indicates earning capacity, which correlates with spending potential.
- **Spending Score**: A measure of spending habits based on the number and value of transactions.


Customer Segmentation using Age and Spending Score


Customer Segmentation using Age and Total Income + Spending Score

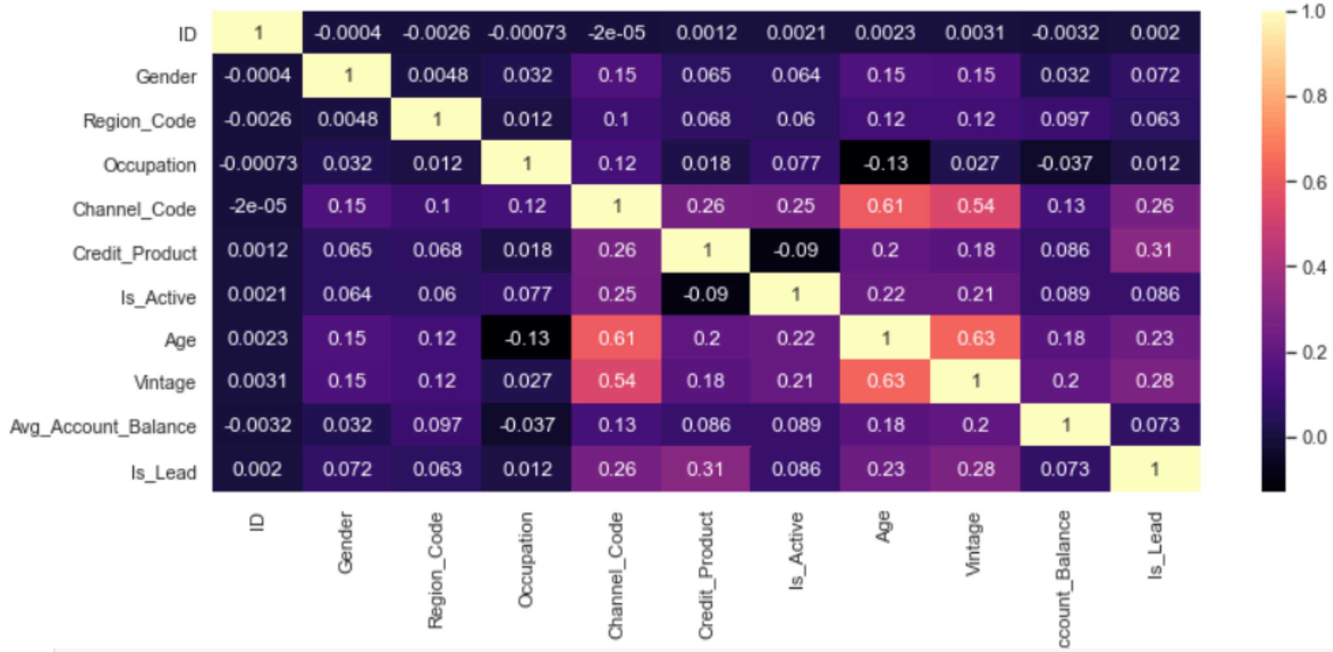
K-Means Clustering

Fig 1: Using variables: **Annual Income** and **Spending Score**, we are able to classify customers as **Average, Spenders, Best, Low budget, Saver, Centroids**

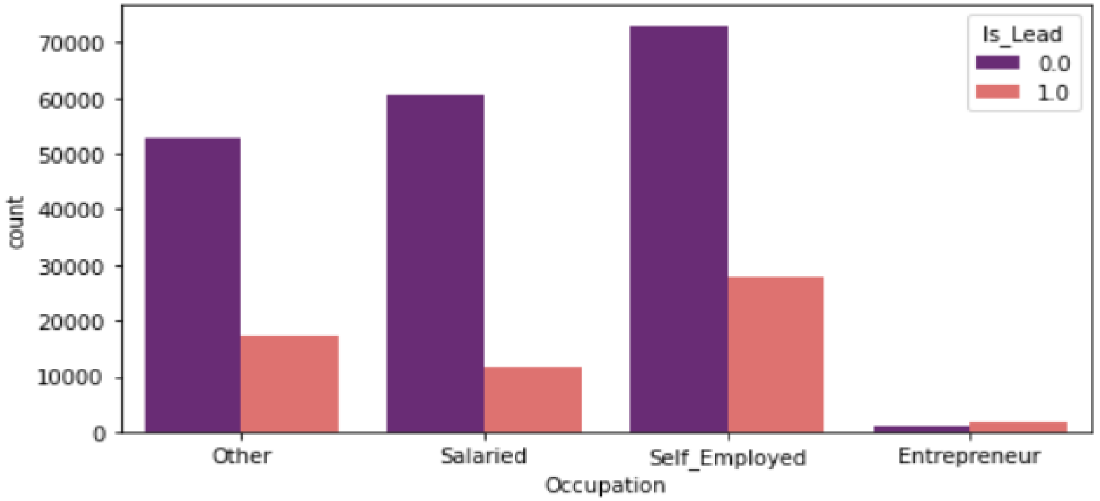
Fig 2: Using variables: **Age** and **Spending Score**, we are able to classify customers as **Regular Customers, Young Targets, Usual Customers, Old Targets, Centroids**

Fig 3: Using variables: **Age** and **Total Income + Spending Score**, we are able to classify customers as **Higher Segment, Middle Segment, Lower Segments, Centroids**

Cross Sell Model



	ID	Gender	Region_Code	Occupation	Channel_Code	Credit_Product	Is_Active	Age	Vintage	ccount_Balance	Is_Lead
ID	1	-0.0004	-0.0026	-0.00073	-2e-05	0.0012	0.0021	0.0023	0.0031	-0.0032	0.002
Gender	-0.0004	1	0.0048	0.032	0.15	0.065	0.064	0.15	0.15	0.032	0.072
Region_Code	-0.0026	0.0048	1	0.012	0.1	0.068	0.06	0.12	0.12	0.097	0.063
Occupation	-0.00073	0.032	0.012	1	0.12	0.018	0.077	-0.13	0.027	-0.037	0.012
Channel_Code	-2e-05	0.15	0.1	0.12	1	0.26	0.25	0.61	0.54	0.13	0.26
Credit_Product	0.0012	0.065	0.068	0.018	0.26	1	-0.09	0.2	0.18	0.086	0.31
Is_Active	0.0021	0.064	0.06	0.077	0.25	-0.09	1	0.22	0.21	0.089	0.086
Age	0.0023	0.15	0.12	-0.13	0.61	0.2	0.22	1	0.63	0.18	0.23
Vintage	0.0031	0.15	0.12	0.027	0.54	0.18	0.21	0.63	1	0.2	0.28
Avg_Account_Balance	-0.0032	0.032	0.097	-0.037	0.13	0.086	0.089	0.18	0.2	1	0.073
Is_Lead	0.002	0.072	0.063	0.012	0.26	0.31	0.086	0.23	0.28	0.073	1



StandardScaler was identified as the most effective scaler. The EDA process can be streamlined using Pandas-Profiling. For modeling, **LightGBM (LGBM)** is used, with parameter tuning achieved through RandomizedSearchCV or GridSearchCV. Hyperparameter tuning optimizes model parameters to enhance accuracy and involves methods like Bayesian Optimization, Evolutionary Algorithms, and Random Search. LightGBM is favored for its efficiency in gradient boosting tasks. Model performance is assessed with AUC-ROC, and a 10-fold validation is employed to average the ROC score.

Data Variables :

Variable	Description
ID	Unique Identifier for a row
Gender	Gender of the Customer
Age	Age of the Customer (in Years)
Region_Code	Code of the Region for the customers
Occupation	Occupation Type for the customer
Channel_Code	Acquisition Channel Code for the Customer (Encoded)
Vintage	Vintage for the Customer (In Months)
Credit_Product	If the Customer has any active credit product (Home loan,Personal loan, Credit Card etc.)
Avg_Account_Balance	Average Account Balance for the Customer in last 12 Months
Is_Active	If the Customer is Active in last 3 Months
Is_Lead(Target)	If the Customer is interested for the Credit Card , 0 : Customer is not interested , 1 : Customer is interested

TVS CREDIT needs to identify customers likely to show higher intent for a recommended credit schemes. To achieve this, consider analyzing customer details (such as gender, age, and region) and their relationship with the bank (including Channel_Code, Vintage, and Avg_Asset_Value). This analysis will help them to find **intentful** customers on scale of 0 to 1.

Credit Score with UPI Cashflow

1. Payment History

1.1 **Statistical Formulae**

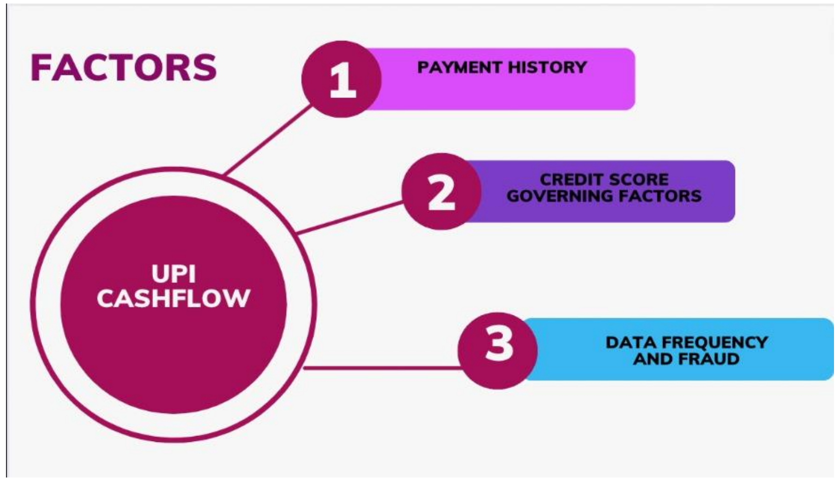
- Consider the **daily average income** of a vegetable vendor in India as metadata.

•**Net Income** = AvgCredit - AvgDebit

- If **Net Income > Daily Avg Income**, it indicates a **positive sign** and a **good UPI cashflow quality**.
- If **Net Income > (Daily Avg Income + Penalties Amount & Frequency)**, it reflects a **GOOD UPI cashflow quality**.

1.2 **Analysis for Non-Salaried Folks**

- **Quarterly Analysis**: Calculate **average monthly income** for every quarter (3, 6, and 9 months).
- **Volatility Check**: Compare volatility with **banking metadata**.
  - If **quarterly avg income > vegetable vendor avg income**, it shows **good UPI cashflow**.
  - Also, check if there's deliberate tinkering to manipulate the cashflow.



2. Credit Score Governing Factors

2.1 **EMIs**

- **Factors**: Specific Date, Amount Debited, Penalties, Total Obligation Amount (90 Days, 3 Months, 6 Months), Avg Monthly Obligation (6 Months), Volatility of Amount.
  - If **debited on a specific date** and **amount debited < income**, then **positive** sign.
  - If **debited on a specific date** and **amount debited > income**, it's okay but **take care**.
  - If **no specific date** and **amount debited > income** with **multiple penalties**, the **UPI cashflow quality is bad**.

2.2 **Bounce Formula**

- **Total Reversal** + **Total Charge Amount** + **Weight of Bounces** > **Income**, then **Red**

Signal and Defaults.

2.3 **Financial Factors for Credit Score**

•**On-time repayment** indicates a **good credit score**. Late payments indicate a **bad credit score**.

•**Unsecured loans** (loans without collateral) lead to a **bad credit score**.

•Spending **beyond the credit limit** shows a **bad credit score**.

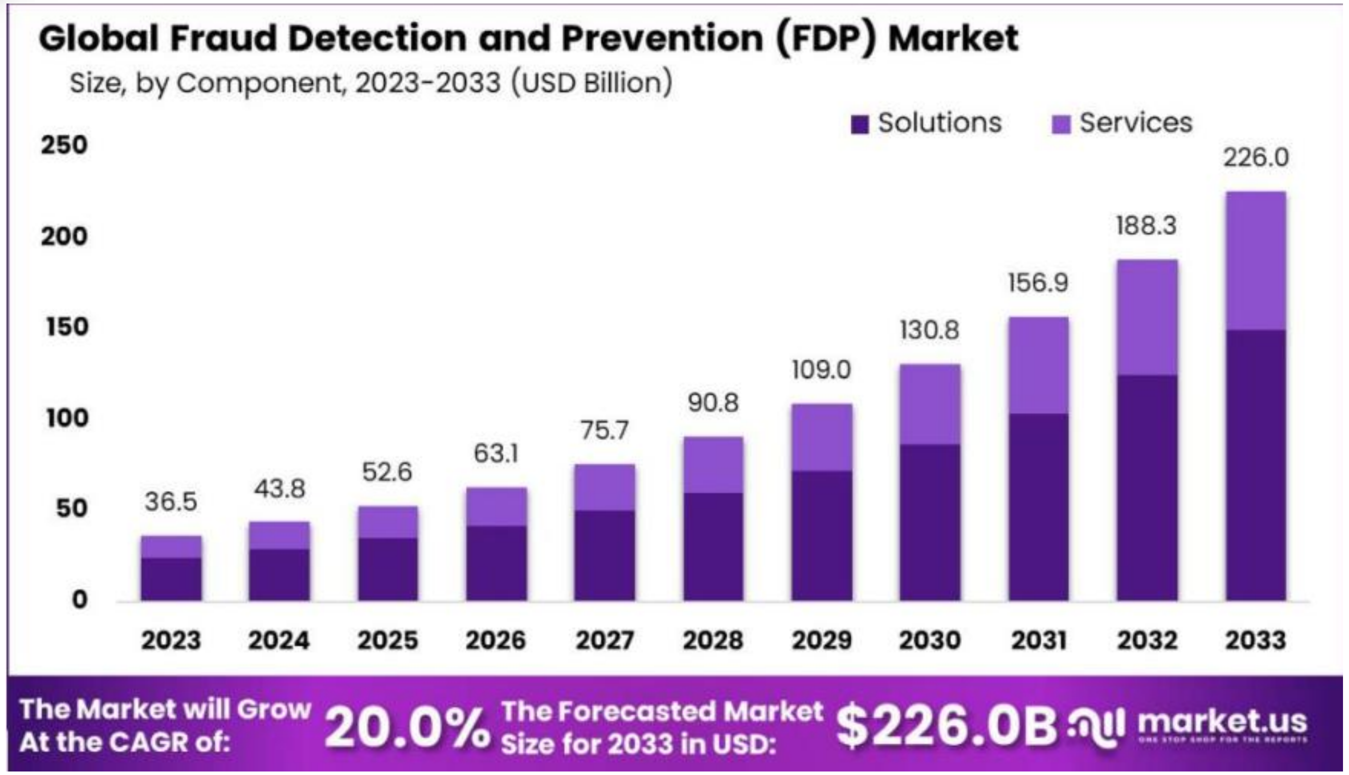
•A **longer credit history** improves the **credit score quality**.

•Association with defaulted accounts **negatively impacts credit score**.

•**True Positive** shows fewer bounces and **good UPI cashflow**

Fraud Detection Model – I

Plan Proposed Earlier



Implementation: To implement our idea, we gathered a dataset of financial transactions, including details like type, amount, time, source, and destination. After preprocessing and feature engineering, we trained a machine learning model using supervised learning methods such as logistic regression, decision trees, and **XGBoost**. The model was trained to detect fraudulent transactions based on historical patterns. We evaluated the model's performance using metrics like accuracy, precision, recall, and F1-score, with **XGBoost** achieving the best accuracy.

suspectRatio: Calculated as (oldbalanceorg + amount) / newba1anceDest . This ratio evaluates whether the transaction amount is proportionate to the destination account's balance. A suspectRatio greater than I could indicate suspicious or fraudulent activity

Dataset Column Descriptions

- **step**: Integer. Maps a unit of time in the real world. In this case, 1 step is 1 hour of time. Total steps: 744 (30 days simulation).
- **type**: String/categorical. Type of transaction: CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER.
- **amount**: Float. Amount of the transaction in local currency.
- **nameOrig**: String. Customer who initiated the transaction.
- **oldbalanceOrg**: Float. Initial balance before the transaction.
- **newbalanceOrig**: Float. New balance after the transaction.
- **nameDest**: String. Customer who is the recipient of the transaction.
- **oldbalanceDest**: Float. Initial balance of recipient before the transaction.
- **newbalanceDest**: Float. New balance of recipient after the transaction.
- **isFraud**: Boolean/binary. Determines if the transaction is fraudulent (encoded as 1) or valid (encoded as 0).
- **isFlaggedFraud**: Boolean/binary. Determines if the transaction is flagged as fraudulent (encoded as 1) or not flagged at all (encoded as 0). An observation is flagged if the transaction is fraudulent and it involved a transfer of over 200,000 in the local currency.

diffOrg: Calculated as oldbalanceorg - newbalanceorig + amount . This metric indicates the difference in the original balance after a transaction, taking into account the transaction amount. A negative diff0rg can suggest discrepancies and potential fraud.

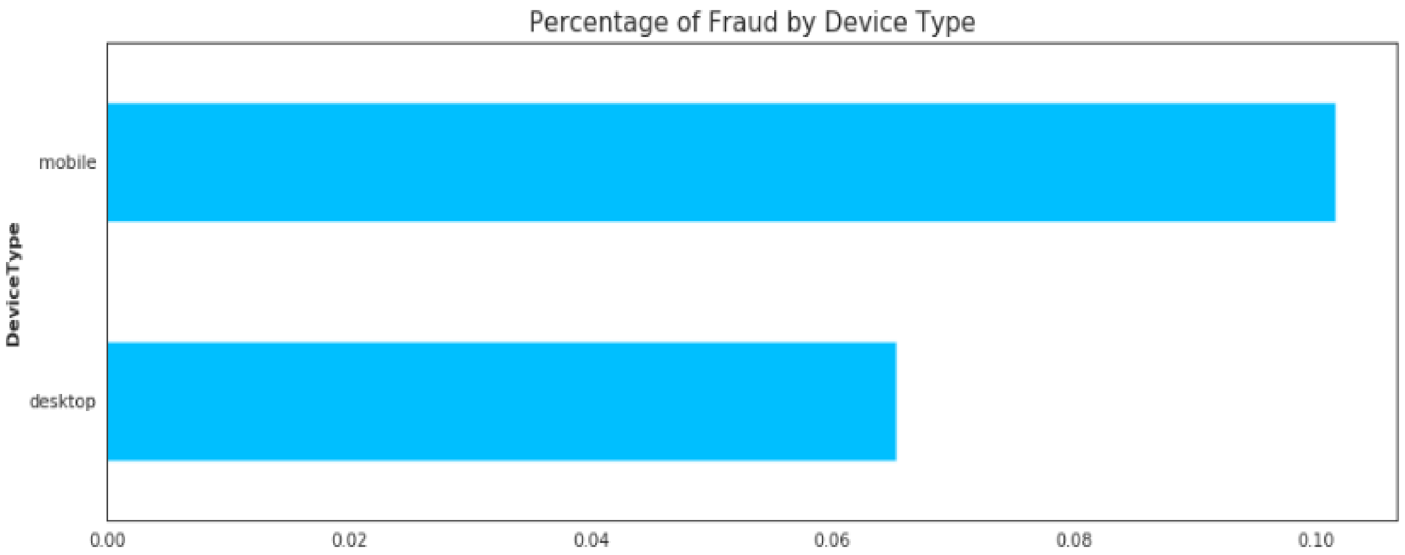
Final Result: Evaluation Metrics: We evaluated the performance of our models on the training, validation, and test datasets using various metrics. Here are the evaluation results:

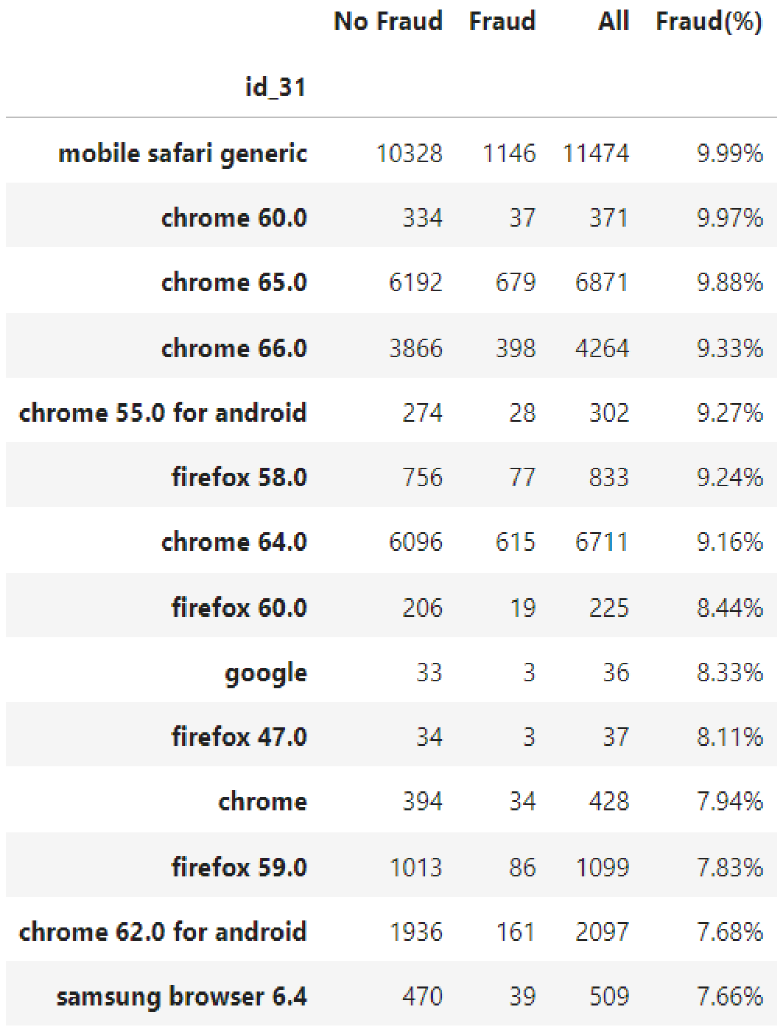
• For Model Trained Test Set: Accuracy. 0.9889 , Fl Score: 0.9890, ROC AUC Score: 0.9889

Fraud Detection Model – II

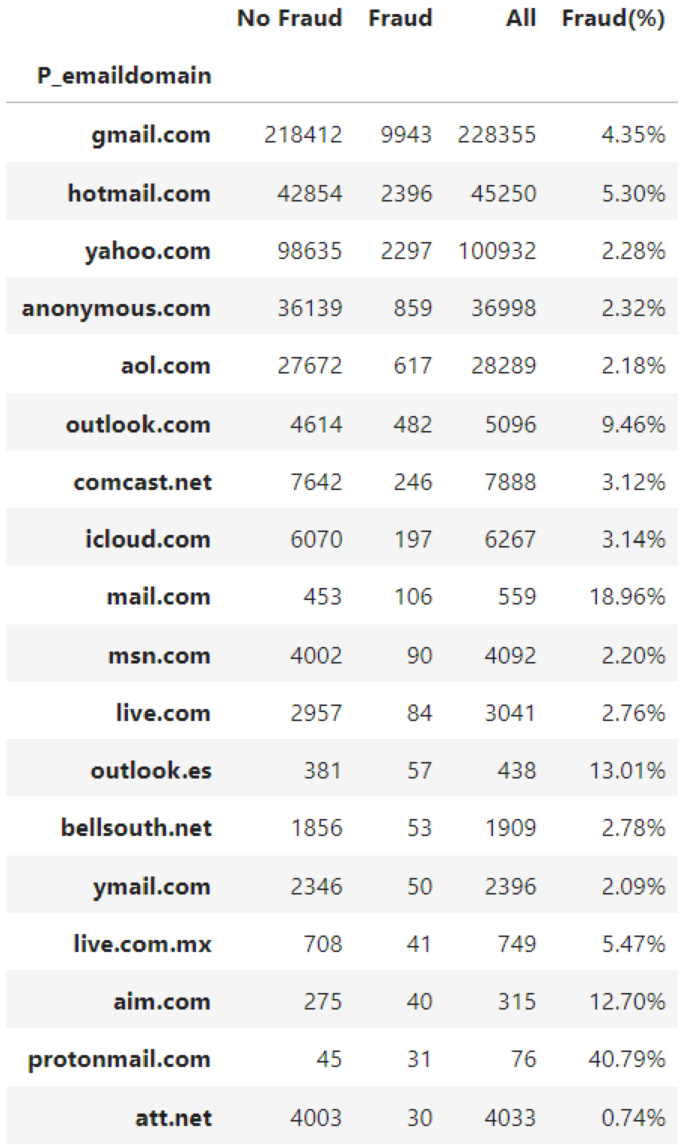
Additional Plan used..

EDA Analysis of the dataset -





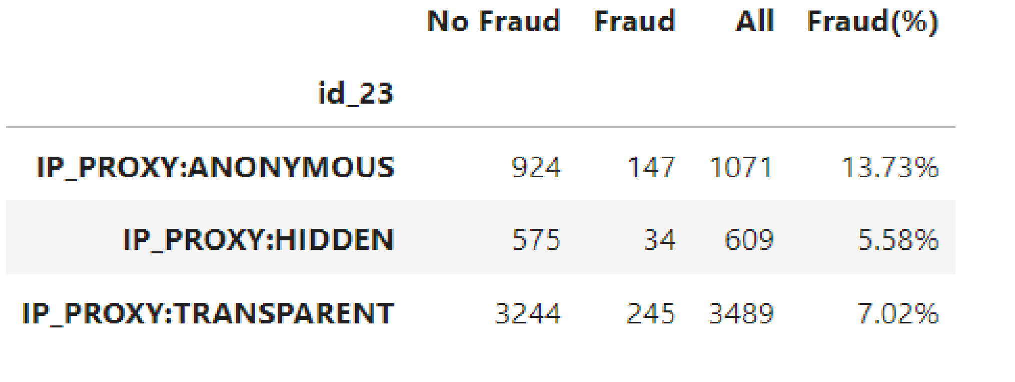
	No Fraud	Fraud	All	Fraud(%)
id_31				
mobile safari generic	10328	1146	11474	9.99%
chrome 60.0	334	37	371	9.97%
chrome 65.0	6192	679	6871	9.88%
chrome 66.0	3866	398	4264	9.33%
chrome 55.0 for android	274	28	302	9.27%
firefox 58.0	756	77	833	9.24%
chrome 64.0	6096	615	6711	9.16%
firefox 60.0	206	19	225	8.44%
google	33	3	36	8.33%
firefox 47.0	34	3	37	8.11%
chrome	394	34	428	7.94%
firefox 59.0	1013	86	1099	7.83%
chrome 62.0 for android	1936	161	2097	7.68%
samsung browser 6.4	470	39	509	7.66%



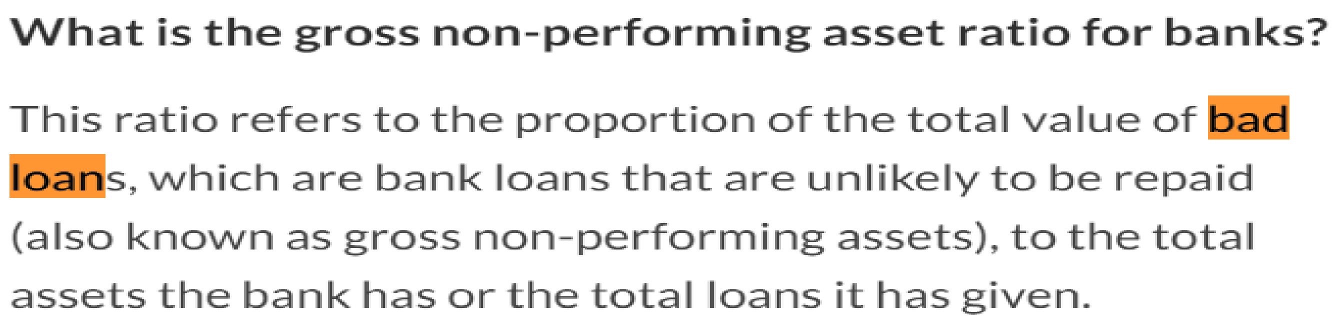
	No Fraud	Fraud	All	Fraud(%)
P_emaildomain				
gmail.com	218412	9943	228355	4.35%
hotmail.com	42854	2396	45250	5.30%
yahoo.com	98635	2297	100932	2.28%
anonymous.com	36139	859	36998	2.32%
aol.com	27672	617	28289	2.18%
outlook.com	4614	482	5096	9.46%
comcast.net	7642	246	7888	3.12%
icloud.com	6070	197	6267	3.14%
mail.com	453	106	559	18.96%
msn.com	4002	90	4092	2.20%
live.com	2957	84	3041	2.76%
outlook.es	381	57	438	13.01%
bellsouth.net	1856	53	1909	2.78%
ymail.com	2346	50	2396	2.09%
live.com.mx	708	41	749	5.47%
aim.com	275	40	315	12.70%
protonmail.com	45	31	76	40.79%
att.net	4003	30	4033	0.74%

Features and variables (Dataset info)

- **TransactionDT**: timedelta from a given reference datetime (not a timestamp)
- **TransactionAMT**: transaction payment in USD
- **ProductCD**: product code, the product for each transaction
- **card1-card6**: payment card information, such as card type
- **addr1-addr2**:: billing region and billing country
- **dist1-dist2**: distances between (not limited) billing address, mailing address, zip code, IP address, phone area, etc.
- **P_ and (R_)emaildomain**: purchaser and recipient email domain, some of the transactions do not require a recipient, and the corresponding Remaildomain is empty.
- **C1-C14**: counting, addresses and other things, actual meaning masked
- **D1-D15**: timedelta, such as days between previous transaction, etc
- **M1-M9**: match, such as names on card and address, etc
- **V1-V339**: Vesta engineered rich features, including ranking, counting, and other entity relations different Some of the V features have different proportions of missing, and their true meaning and treatment are still unknown.



	No Fraud	Fraud	All	Fraud(%)
id_23				
IP_PROXY:ANONYMOUS	924	147	1071	13.73%
IP_PROXY:HIDDEN	575	34	609	5.58%
IP_PROXY:TRANSPARENT	3244	245	3489	7.02%



What is the gross non-performing asset ratio for banks?

This ratio refers to the proportion of the total value of bad loans, which are bank loans that are unlikely to be repaid (also known as gross non-performing assets), to the total assets the bank has or the total loans it has given.

1. **Risk Management**: Accurate predictions help TVS Credit manage risk more effectively. By identifying potential bad loans early, they can take preventive measures such as adjusting interest rates, requiring additional collateral, or even rejecting high-risk applications.
2. **Financial Performance**: Good loan predictions can improve the company's financial health by reducing the number of defaults. This leads to better cash flow and profitability. Conversely, poor predictions can result in higher default rates, impacting the company's bottom line.

Good or Bad Loans

TVSCREDIT e.p.i.c 6 ENRICH | PERFORM | INNOVATE | CHALLENGE

Dataset's Features Overview For Good/Bad Loan

- **term**: The number of payments on the loan, where values are in months and can be either 36 or 60.
- **int_rate**: The interest rate on the loan.
- **sub_grade**: Assigned loan subgrade score based on borrower's credit history.
- **emp_length**: Borrower's employment length in years.
- **dti**: A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage, divided by the borrower's monthly income.
- **mths_since_recent_inq**: Months since most recent inquiry.
- **revol_util**: Revolving line utilization rate, or the amount of credit the borrower uses relative to all available revolving credit.
- **bc_util**: Ratio of total current balance to high credit/credit limit for all bankcard ac
- **num_op_rev_tl**: Number of open revolving accounts.

Observations -

We have a lot of features but we got these top 9 features using Logistic Regression with **SequentialFeatureSelector.**

Logistic Regression Accuracy of Logistic Regression: **81.2749%**

RandomForestClassifier Accuracy of RandomForestClassifier: 80.4781%

KNeighborsClassifier Accuracy of KNeighborsClassifier: 80.0797%

Annexure

Dataset links -

Customer Segmentation Dataset : https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python

Cross-sell Dataset : https://www.kaggle.com/code/prakharprasad/credit-card-lead-prediction

Fraud Detection I Dataset : https://www.kaggle.com/datasets/gopalmahadevan/fraud-detection-example

Fraud Detection II Dataset : https://www.kaggle.com/c/ieee-fraud-detection

Good or Bad Loans Dataset : https://www.kaggle.com/datasets/utkarshx27/lending-club-loan-dataset

Research links -

TVS Credit limited Annual Report: https://www.tvscredit.com/wp-content/uploads/2024/07/TVSCS-Annual-Report-2024.pdf

Cashflow Analysis: https://www.netsuite.com/portal/resource/articles/financial-management/cash-flow-analysis.shtml

IEEE CIS Fraud Detection: https://ieeexplore.ieee.org/document/9077872