1. Data cleaning including missing values, outliers and multi-collinearity.

- ⇨ The dataset contains **no missing values or duplicates**.
- ⇨ Outliers were identified using **boxplots**, but **not removed** because they carry important signals for fraud detection.
- ⇨ For **multicollinearity**, a heatmap and Variance Inflation Factor (VIF) were used.
- ⇨ High VIF values were found for oldbalanceOrg, newbalanceOrig, oldbalanceDest, and newbalanceDest.
- ⇨ To reduce multicollinearity, new features deltaOrig (oldbalanceOrg - newbalanceOrig) and deltaDest (newbalanceDest - oldbalanceDest) were created.
- ⇨ The original correlated columns were then **dropped** from the dataframe to improve model stability.

2. Describe your fraud detection model in elaboration.

- ⇨ The fraud detection model is designed to identify fraudulent financial transactions within a large-scale fintech dataset. The model uses a combination of transactional and derived features, such as transaction type, amount, and balance changes (deltaOrig, deltaDest), to capture behavioral patterns indicative of fraud.
- ⇨ A tree-based ensemble method like XGBoost or Random Forest is employed due to its ability to handle complex, non-linear relationships and robustness against outliers. The model is trained on a highly imbalanced dataset, where fraudulent transactions are rare, so class imbalance is addressed via techniques like class weighting or sampling. Yet Random Forest(best f1-score) takes too much time to be trained.
- ⇨ Feature engineering plays a key role by transforming raw balance data into difference features that reduce multicollinearity and improve predictive power. The model's performance is evaluated using precision, recall, F1-score, and ROC-AUC metrics to ensure a balanced detection of frauds while minimizing false alarms.
- ⇨ Finally, ANN is performed to optimize model generalization, Which was the best. However except Random Forest, All the models have very less precision (which is not important in this case), but have been able to capture the Fradulence properly.

3. How did you select variables to be included in the model?

- ⇨ To select variables, I performed feature engineering and multicollinearity analysis:

- Created new features capturing transaction behavior, such as flag_balance_mismatch (flagging zero old balance with positive amount), amount_gt_old (amount greater than old balance), and time-related hour from step.
- Derived balance difference features deltaOrig = oldbalanceOrg - newbalanceOrig and deltaDest = newbalanceDest - oldbalanceDest to capture balance changes during transactions.
- Added net_balance_change as the difference between origin and destination balance changes to summarize net fund movement.
- Used Variance Inflation Factor (VIF) to detect multicollinearity; found high VIF in the original balance columns, so dropped oldbalanceOrg, newbalanceOrig, oldbalanceDest, and newbalanceDest to reduce redundancy and improve model stability.

4. Demonstrate the performance of the model by using best set of tools.

I have used Random Forest Clf, XgBoost Clf, lightGBM, Catboost and finally Artificial Neural Network.

1. `Random Forest` --> precision= 0.97 `**` recall= 0.79 `**` ROC-AUC Score= 95%
2. `XgBoost(RandomizedSearchCV)` --> recall = 0.97 `**` F1 score= 0.12 `**` ROC-AUC score = 0.97
3. `LightGBM` --> recall = 0.93 `**` F1 score = 0.12 `**` ROC-AUC score = 0.99
4. `Catboost` --> recall = 0.96 `**` F1 score = 0.25 `**` ROC-AUC score = 0.99
5. `ANN` ---> recall = 0.99 `**` F1 score = 0.02 `**` ROC-AUC score = 0.99

- XgBoost RandomizedSearchCV (`Best Params: {'n_estimators': 1000, 'max_depth': 8, 'learning_rate': 0.01})`

5. What are the key factors that predict fraudulent customer?

- Transaction type — certain types (e.g., “TRANSFER” or “CASH_OUT”)
- many high-value transactions in a short time frame.
- Balance inconsistencies
- Old balances is greater then Transactions amounts.

6. Do these factors make sense? If yes, How? If not, How not?

- Fraudsters often target higher-value transactions because the gain is bigger. Yet, here in this specific dataset the majority of both fraud and non-fraud transactions are clustered at lower amounts.
- Fraudsters might focus on accounts with large balances because the potential gain is higher.

- No fraud transactions are sent to merchants (M).
- All fraud cases happen when money is sent to a customer (C).
- New or inactive accounts — accounts that suddenly become active with large transfers are high risk.
- Fraudsters might use zero-balance accounts to mask their identity or exploit system loopholes.

7. What kind of prevention should be adopted while company update its infrastructure?

- I. Implement systems capable of instantly detect suspicious pattern and fraudulent activities.
- II. Integrate robust ML models that continuously learn from new fraud patterns and adapt to evolving threats and regularly update the model.
- III . Set dynamic thresholds and limits for transactions based on risk profiles, and notify customers of unusual activities.
- IV. Develope a powerful API for bacjend against cyber-attacks.

8. Assuming these actions have been implemented, how would you determine if they work?

- Monitor key metrics like fraud detection **precision**, **recall**, **F1-score**, and **false positive rate** over time.
- Measure reduction in **customer complaints** related to false alarms.Also Collect feedback from customers for qualitative insights.
- Use A/B testing deployments to evaluate impact on a subset before full rollout.