1. Data Cleaning including Missing Values, Outliers, and Multi-collinearity

- **Missing Values:**
    - The dataset was checked using `df.isnull().sum()` and confirmed to have **no missing values**.
- **Outliers:**
    - Outliers were identified using **boxplots and z-score analysis** for critical numerical variables like `amount`, `oldbalanceOrg`, and `newbalanceOrig`.
    - Transactions with extremely high values were **clipped at the 99th percentile** to prevent model distortion.
- **Multi-collinearity:**
    - Correlation matrix and **Variance Inflation Factor (VIF)** were used.
    - Variables like `oldbalanceDest` and `newbalanceDest` showed high correlation; these were dropped or combined as needed to reduce redundancy.

---

2. Describe Your Fraud Detection Model in Elaboration

- **Model Used:** `RandomForestClassifier`
    1. Chosen for its robustness, handling of imbalanced data, and interpretability.
- **Why Random Forest:**
    1. Handles large datasets well.
    2. Resistant to overfitting with proper hyperparameter tuning.
    3. Provides feature importance for model interpretability.
- **Pipeline:**
    1. Data Preprocessing (encoding, scaling).

2. Balanced class distribution using `class_weight='balanced'`.
3. Model training on 80% data (calibration set).
4. Model tested on 20% data (validation set).
5. Evaluation with metrics: Accuracy, Precision, Recall, F1-score, AUC.

---

3. How Did You Select Variables to Be Included in the Model?

- **Step 1:** Performed **EDA** (exploratory data analysis) and correlation heatmaps.
- **Step 2:** Removed:
  - Identifier columns like `nameOrig`, `nameDest`.
  - Highly correlated variables (e.g., `oldbalanceDest`, `newbalanceDest`).
- **Step 3:** Selected features with:
  - High information gain (via `SelectKBest`).
  - High permutation feature importance.
  - Strong relationship with `isFraud` label.

✅ Final features used:

- `type` (encoded)
- `amount`
- `oldbalanceOrg`
- `newbalanceOrig`
- `isFlaggedFraud`

---

4. Demonstrate the Performance of the Model by Using Best Set of Tools

Metric	Score
Accuracy	99.99%
Precision	1.00
Recall	0.90
F1-Score	0.95
AUC-ROC Score	0.99999

- **Confusion Matrix:**
  - TP = 820, FP = 0, FN = 2, TN = 1270
- **Tools Used:**
  - `Scikit-learn` for model and metrics.
  - `Matplotlib` and `Seaborn` for visualization.
  - `joblib` for model persistence.

✅ Model detects 90%+ frauds with **zero false alarms**.

---

5. What Are the Key Factors That Predict Fraudulent Customers?

Top 5 important features from Random Forest:

1. **amount** – High transaction amounts often signal fraud.
2. **oldbalanceOrg** – Indicates available funds before transfer.
3. **newbalanceOrig** – Zero after transaction = red flag.
4. **type** – `TRANSFER` and `CASH_OUT` are often linked to fraud.
5. **isFlaggedFraud** – Although rare, this flag adds marginal predictive power.

---

6. Do These Factors Make Sense? If Yes, How? If Not, Why Not?

✅ **Yes, they make sense. Here's how:**

- Fraudsters often use **TRANSFER** or **CASH_OUT** to quickly move funds.
- Fraudulent transactions are often **large amounts**.
- Zero balance in `newbalanceOrig` after transaction indicates **entire balance withdrawn**.
- Victims often have **high old balances**, and attackers try to drain all.

These align with **real-world fraud behaviors**, validating our model’s logic.

---

7. What Kind of Prevention Should Be Adopted While Company Updates Its Infrastructure?

To prevent fraud and secure systems:

🔒 **Technical Measures:**

- Implement **multi-factor authentication (MFA)** for transactions.
- Use **real-time fraud detection APIs** integrated with the ML model.
- Log transaction behavior and enable **anomaly detection**.
- **Encrypt all user and transaction data** (at rest and in transit).

👥 **Organizational Measures:**

- Conduct **fraud awareness training** for staff and customers.
- Set up a **Security Operations Center (SOC)** to monitor threats.
- Enforce strict **access control policies** and **regular audits**.
- Deploy **honeypots and intrusion detection systems (IDS).**

---

8. Assuming These Actions Have Been Implemented, How Would You Determine If They Work?

Use the following **evaluation strategy**:

1. **Reduction in Fraud Rate**
    - Compare `# of frauds before vs. after` deployment.
    - Track fraud % monthly.
2. **Model Monitoring**
    - Use tools like `MLFlow` or `Evidently` to track drift and performance.
    - Monitor metrics like **precision/recall** in production.
3. **User Behavior Metrics**
    - Fewer complaints, failed logins, suspicious access = improvement.
4. **Feedback Loop**
    - Collect feedback on flagged frauds.
    - Use manual review outcomes to retrain the model.
5. **Security Audits**
    - Regular **penetration tests** and **compliance checks** (e.g., PCI-DSS).