CREDIT CARD FRAUD DETECTION

- FEAUTURE ENGINEERING
- MODEL TRAINING
- EVALUATION

FEAUTURE ENGINEERING FOR CREDIT CARD FRAUD DETECTION

- The article discusses the importance of feature engineering in credit card fraud detection. Feature engineering involves selecting and transforming relevant data to improve the accuracy of machine learning models. The author argues that feature engineering is crucial in detecting credit card fraud because it allows for the identification of patterns and anomalies in transaction data.

- The article highlights several techniques used in feature engineering, including normalization, scaling, and dimensionality reduction. Normalization involves transforming data to a common scale to eliminate

bias, while scaling involves adjusting the range of values to improve model performance. Dimensionality reduction techniques such as principal component analysis (PCA) are used to reduce the number of features in a dataset without losing important information.

- The author also emphasizes the importance of domain knowledge in feature engineering. Domain knowledge refers to understanding the specific characteristics and patterns of credit card transactions that are indicative of fraud. This knowledge can be used to create new features or modify existing ones to improve model accuracy.

- Finally, the article discusses the challenges associated with feature engineering, including time constraints and overfitting. Overfitting occurs when a model is too complex and performs well on training data but poorly on new data. The author suggests using cross-validation techniques to prevent overfitting and ensure model accuracy.

- Overall, the article highlights the importance of feature engineering in credit card fraud detection and provides insights into various techniques used in this process.

- A fraudster will try to abuse the card as much as possible in a **short period** before the card is detected and suspended.

MODEL TRAINING

- Certainly! When it comes to training a credit card fraud detection model with feature engineering in Python, there are several important steps to consider. Feature engineering involves selecting and transforming relevant features to improve the model's performance. Here's a high-level overview of the process:
- **Data Collection**: Start by gathering a comprehensive dataset that includes credit card transaction data. This data should contain information about transactions, including transaction amount, timestamp, merchant information, and whether the transaction is fraudulent or not.

- **Data Preprocessing**:
- **Data Cleaning**: Remove any missing or inconsistent data.
- **Feature Selection**: Choose the most relevant features for fraud detection. This may include transaction amount, location, time, etc.
- **Label Encoding/One-Hot Encoding**: Convert categorical variables into a numerical format if needed.
- **Normalization/Scaling**: Scale numerical features to have a standard range.
- **Feature Engineering**:
- **Creating New Features**: Generate new features that may be informative for fraud detection. For example, you can calculate the time since the last transaction, average transaction amount, or the distance

between the transaction location and the cardholder's home address.

- **Feature Transformation**: Apply mathematical transformations like logarithms or square roots to make the data more suitable for modeling.
- **Data Splitting**: Divide the data into training, validation, and testing sets to evaluate model performance effectively.
- **Model Selection**:
- Choose an appropriate machine learning algorithm for fraud detection. Common choices include logistic regression, decision trees, random forests, or neural networks.
- Consider ensemble methods and anomaly detection algorithms as well.

- **Model Training**:
- Train your selected model on the training data.
- Tune hyperparameters to optimize the model's performance.
- **Model Evaluation**:
- Evaluate the model on the validation set using metrics like accuracy, precision, recall, F1-score, and the receiver operating characteristic (ROC) curve.
- Adjust the model as needed based on the evaluation results.
- **Testing and Deployment**:
- Test the final model on the testing dataset to assess its real-world performance.Once satisfied with the

results, deploy the model for live fraud detection.

- **Continuous Monitoring**:

Continuously monitor the model's performance and update it as necessary to adapt to new fraud patterns.

- For the feature engineering part, Python offers a wide range of libraries and tools like NumPy, Pandas, and Scikit-Learn to help you with data manipulation and model training. You can also explore advanced techniques like dimensionality reduction (e.g., PCA) and feature selection algorithms to further enhance your model's performance.
- Remember that credit card fraud detection is a critical application, and

it's essential to stay updated with the latest fraud patterns and adapt your model accordingly.

Feature Engineering:

- **Time-Related Features**:
- Extract relevant information from the timestamp, such as hour of the day, day of the week, or time since the previous transaction.
- **Transaction Amount Features**:
- Create statistics like the mean, median, and standard deviation of transaction amounts for a given card.
- **Geographical Features**:
- Calculate the distance between the transaction location and the cardholder's address.

Card-Related Features:

- Generate features related to the card, such as the number of previous transactions for the card in a specific time window.

Behavioral Features:

- Create features that capture the cardholder's spending behavior, such as the frequency of transactions.

Aggregated Features:

- Compute rolling averages, sums, or other aggregations over time windows.

PCA (Principal Component Analysis):

- Apply dimensionality reduction techniques like PCA to capture the most significant variations in the data.

EVALUATION

When evaluating your credit card fraud detection model, you'll want to use appropriate evaluation metrics:

Confusion Matrix:

True Positives (TP): The number of correctly predicted fraudulent transactions.

True Negatives (TN): The number of correctly predicted legitimate transactions.

False Positives (FP): The number of legitimate transactions incorrectly classified as fraudulent.

False Negatives (FN): The number of fraudulent transactions incorrectly classified as legitimate.

Accuracy: The ratio of correctly predicted transactions (both fraudulent and legitimate) to the total number of transactions. While accuracy is essential, it may not be the most informative metric for imbalanced datasets.

Precision:

The proportion of true positive predictions among all positive predictions (fraudulent transactions). It measures the model's ability to avoid false alarms.

Recall (Sensitivity or True Positive Rate): The ratio of true positive predictions to the total number of actual positive cases (fraudulent transactions). It measures the model's ability to detect fraud.

F1-Score:The harmonic mean of precision and recall. It balances the trade-off between precision and recall.

ROC Curve and AUC:
The Receiver Operating Characteristic (ROC) curve visualizes the model's performance across different threshold values. The Area Under the Curve (AUC) summarizes the ROC curve's performance.
Specificity:
The proportion of true negative predictions among all actual negative cases. It complements recall by measuring the model's ability to avoid false positives.
Balanced Accuracy:
An accuracy metric that considers class imbalance, providing a more balanced view of the model's performance.In practice, selecting the most suitable evaluation metrics depends on the specific goals of your credit card fraud detection system.

DATASET

0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.4039
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.145
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.261
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.232
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.8034
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.033
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.045
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.3245
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.5703
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.4517
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.221
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.7076
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.683
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.982
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.2218
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.4325
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.575
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.0254
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.406

THANK YOU