Machine Learning Project

Detection of fraudulent transactions

Fraud Detection - before machine learning

- In the past, fraud detection relied upon rulesets to guide transaction assessment.
- However, fixed rules do not account for the ever-evolving methods to produce fraudulent transactions.
- Also, if rules are too strict in order to try to prevent most frauds, they might block many genuine transactions and impact customers' satisfaction

Fraud Detection - after machine learning

- Models can be deployed to function in an automated fashion, which speeds ups the process of fraud detection.
- Models can learn new patterns of fraudulent transactions very fast, and so they are promptly to keep up with new forms of fraud attempts.
- Automated functioning after deploying and readiness to adapt makes this approach very scalable.

About the project

The aim of this project is to build a machine learning model to predict fraudulent credit card transactions.

- The key metric is recall as it indicates the proportion of fraudulent transactions being identified.

If the tolerance for allowing transactions is too low, high recall is expected (that is, most fraudulent transaction will be captured by the model), but many legitimate transaction will be blocked.

If the tolerance is too high, recall will be lower, most legitimate transaction will be allowed, but so does some fraudulent.

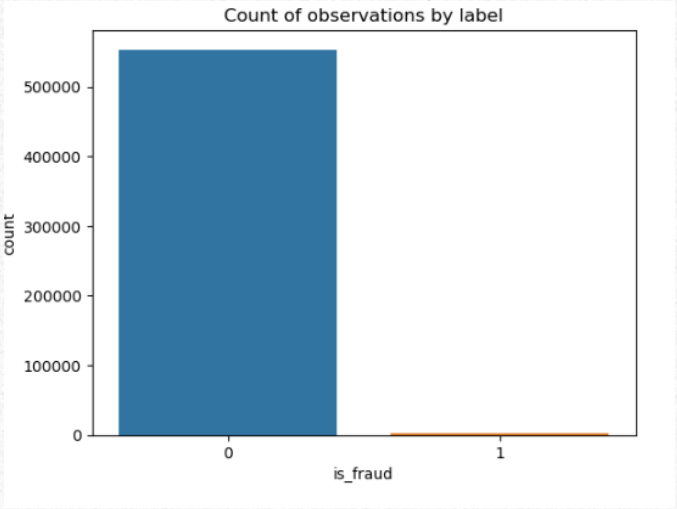
Dataset

Source	This Project utilized a public dataset available in Kaggle (https://www.kaggle.com/datasets/kelvinkelue/credit-card-fraud-prediction?resource=download)
Description	The original dataset had 22 atributes (columns) regarding trasactions and 555719 obsevations (rows)
Feature engineering	The columns "month", "weekday", "hour", "lat_dist", "long_dist" and "age" were added to the dataset during feature engineering.
Preprocessed data	After being preprocessed for modeling, the data kept 21 features: “amt”, “month”, “weekday”, “hour”, “lat_dist”, “long_dist”, “age”, merchant categories (dummified into 13 features), gender (dummified into 1 feature).

For more info:
https://www.kaggle.com/datasets/kelvinkelue/credit-card-fraud-prediction?resource=download

Exploratory Data Analysis

Unbalaced target variable: only 0.386% of the transactions were labeled as fraud (label "1")



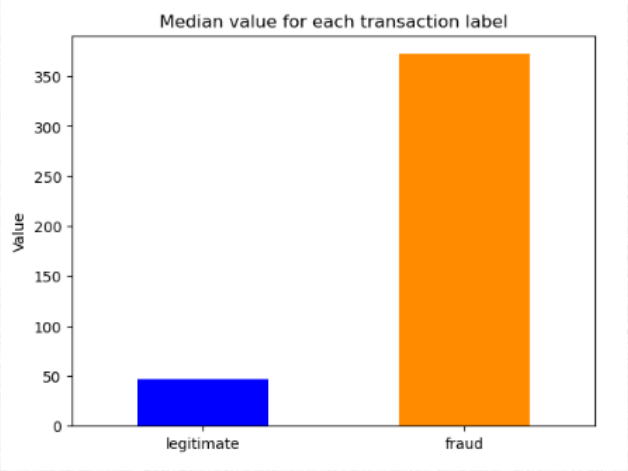
Legitimate

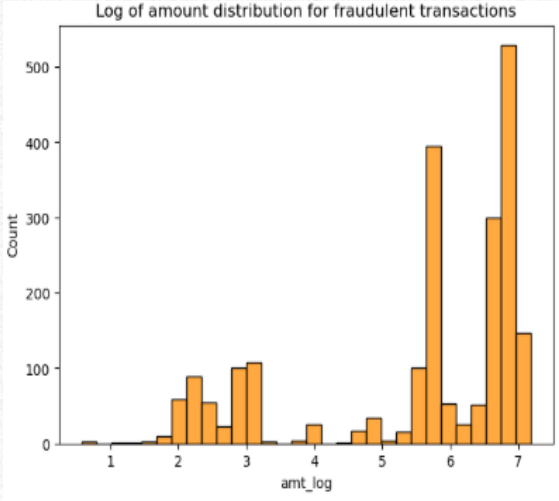
Fraudulent

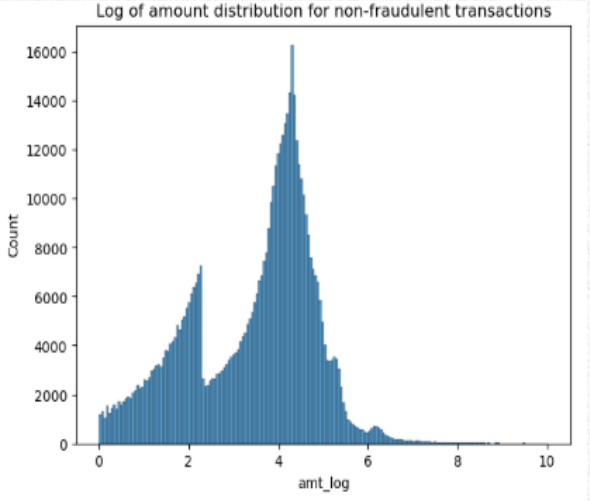
Exploratory Data Analysis

Transaction amount and fraudulent transactions: positive correlation, different median values and distribution for legitimate and fraudulent values.

R = 0.18. Median values: fraudulent = 371.94; non-fraudulent = 47.91.

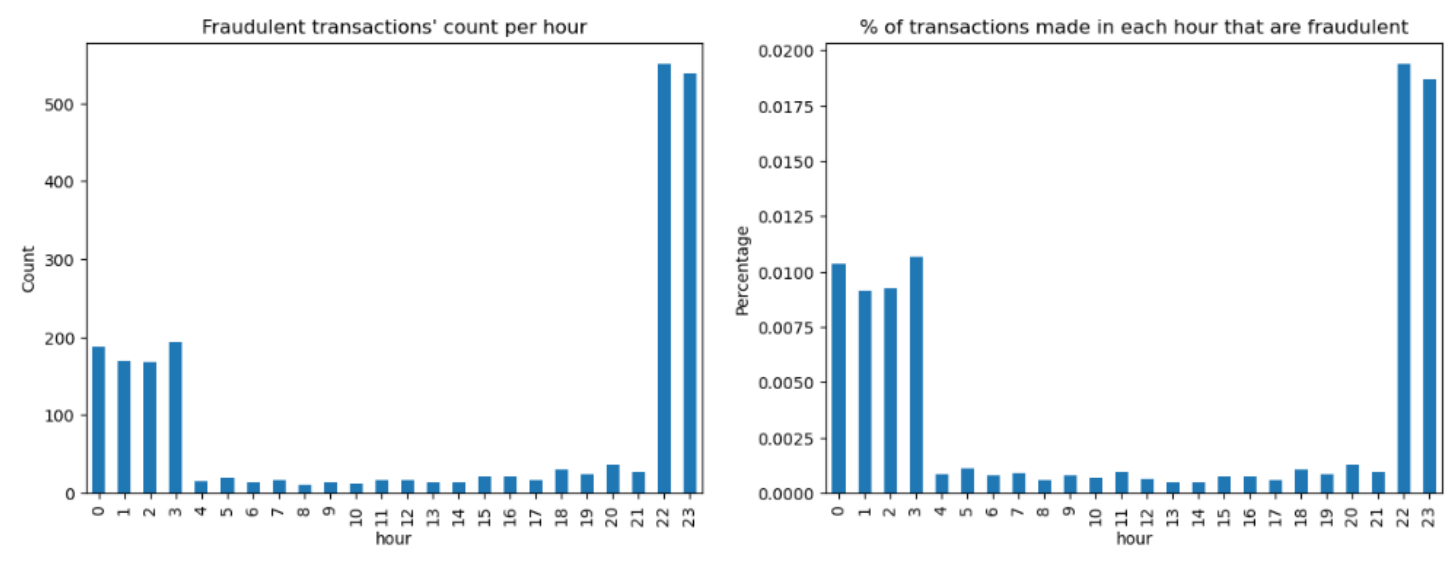






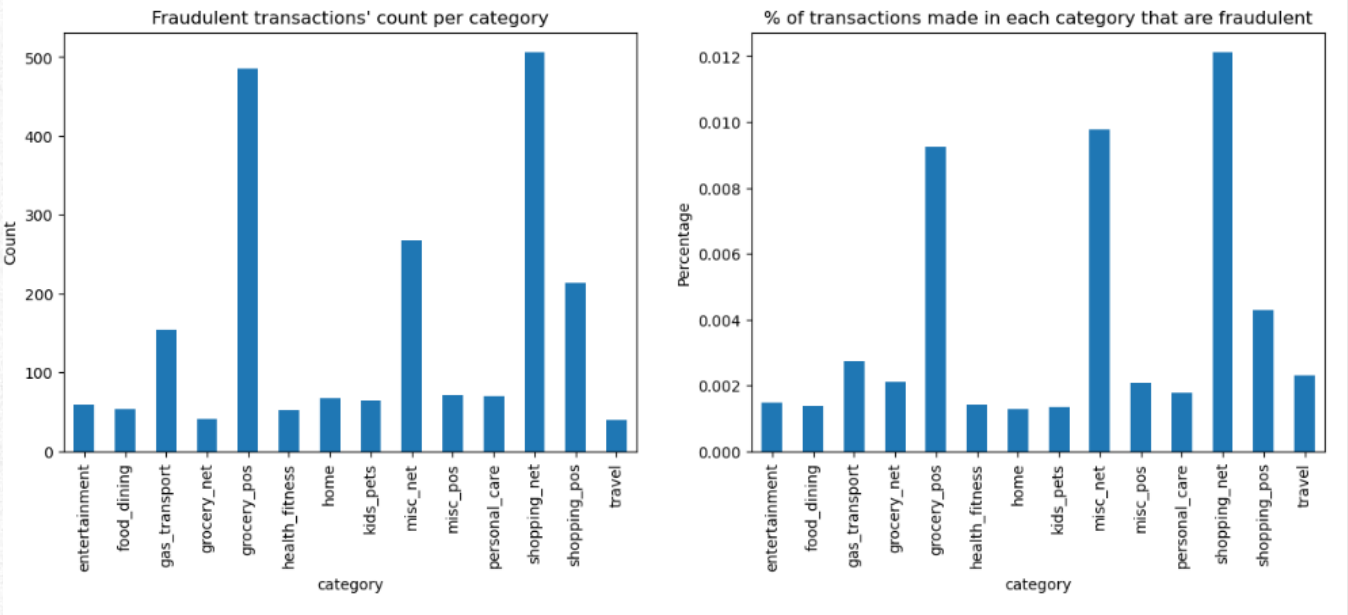
Exploratory Data Analysis

Hour and fraudulent transactions: most fraudulent transactions happened between 22:00 and 3:00.



Exploratory Data Analysis

Category and fraudulent transactions: "grocery_pos, "misc_net", "shopping_net", and "shopping_por" categories seems to be overrepresented when it comes to fraud.



Baseline models

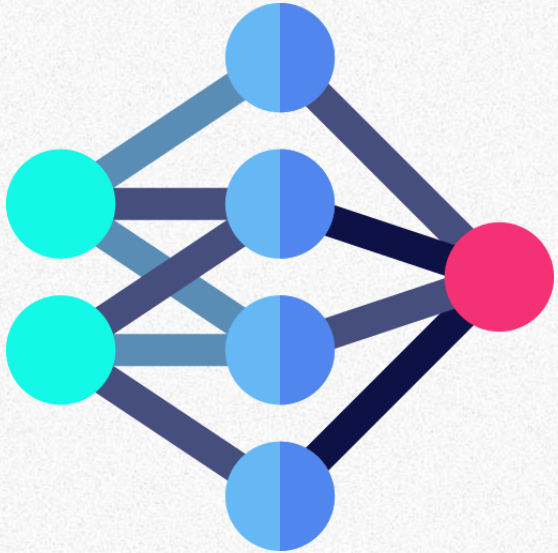
- Two model were built to serve as baseline: a logistic regression and a decision tree model.

Logistic Regression	Accuracy = 88.8% / Precision = 2.5% / Recall = 74.9% / F1 = 4.9%
Decision Tree	Accuracy = 99.8% / Precision = 78.1% / Recall = 73.2% / F1 = 75.5%

- Based on these metrics and the fact that non-fraudulent transactions represent more than 99% of the data, the goal set was to build a model able to predict correctly at least 99% of the time and identify 90% or more of fraud attempts.

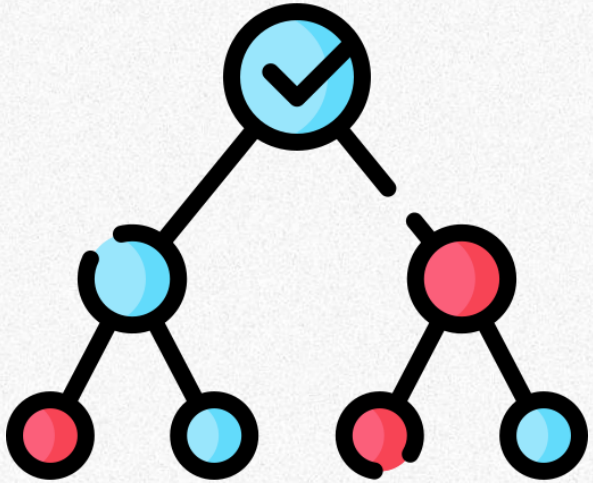
Modeling – Neural Network Models

- Four sequential neural network models were built and evaluated using a validation dataset.
- The model that performed the best was a deep neural network (more than 800 nodes).
- None of the models were able to reach the proposed goal. The best metrics on the validation set were accuracy = 99.5% ; precision = 44.8%; recall = 81.1%.



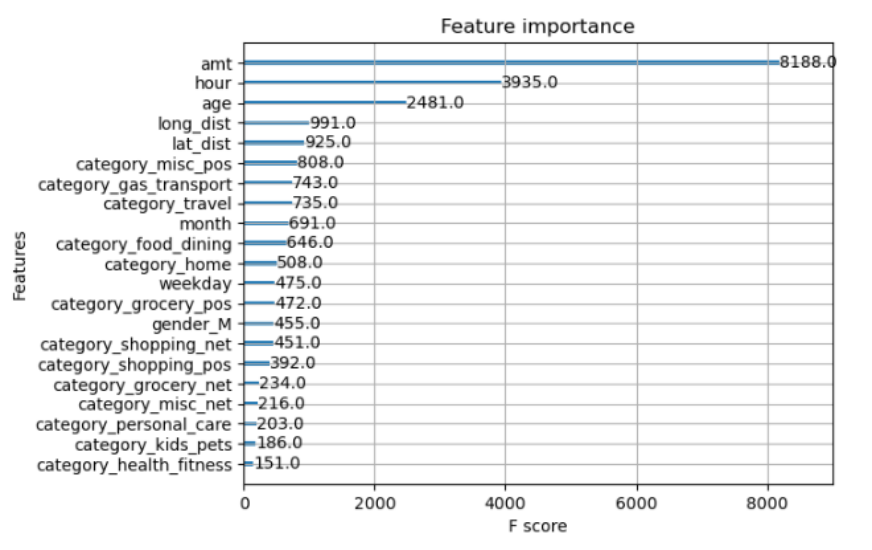
Modeling – XGBoost models

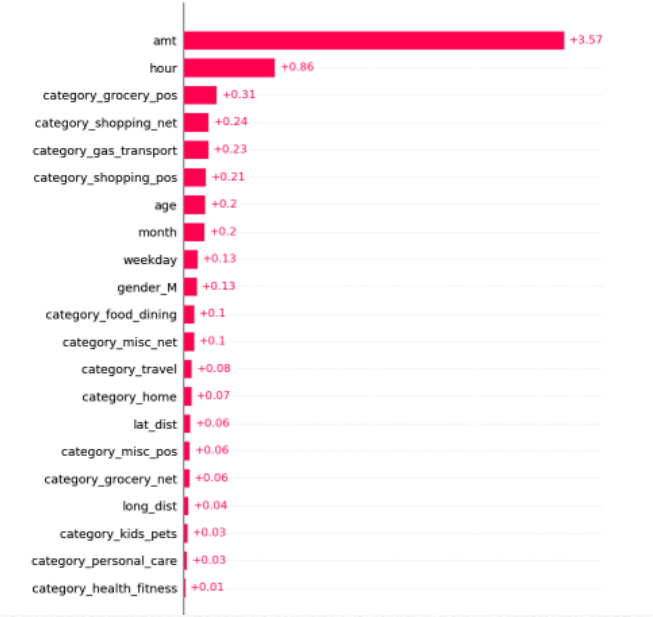
- The validation set was used to find optimal paramets for a XGBoost Model.
- The best paramets found were max_depth = 5 lr = 0.01.
- A XGBoost classifier model was build and scored at the test set. Its metrics achieved the proposed goal: accuracy = 99.1%; precision = 29.0%; recall = 92.1%; f1 = 44.2%.



Modeling – Model explanation

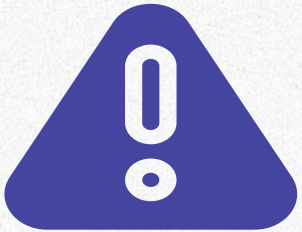
- Predictions of the best model were explained using XGBoost innate feature importance and SHAP.
- Results show that transaction amount and hour of transaction were the most important features.





Challenges

- Deciding which features to keep for modeling.
- Memory crashes due to high memory usage.
- Setting the goal without a business context.
- Taking the decision of which model to keep at the end as no model performed best at all metrics.



Conclusions

The results achieved by the final XGBoost model have to be considered within the context of a company. Even though 92% recall seems to be great at first glance, the business can have an even higher expectation and further exploration of models could be done by parameter tuning, trying different kinds of neural networks, and so on.

In case this model is satisfactory, some final business recommendations would be:

- Deploy the model for company everyday use. If performance is as expected, the model should be able to prevent about 92% of fraud attempts in an automated way.

- Test the suggested model in the real world and verify if it can achieve the 99% accuracy and 90% recall that were set as the initial goal. If performed it less than expected, further exploration of models can and should be performed.

- Continuously refine and improve the model to be used based on new data collected after its implementation.

Thanks!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**