ML minor, Report assignment1
Finn Thomas - 722051
Robin Tollenaar - 672023

Introduction

Machine learning is a branch of artificial intelligence that focuses on the development of algorithms allowing computer systems to learn and improve their performance on tasks by analyzing and adapting to data. It utilizes statistical methods for pattern recognition, enabling applications such as image analysis and natural language processing to make predictions and decisions based on learned patterns.

This report covers the training of two machine learning models, K-NN and Naive Bayes, on a dataset of fraudulent and non-fraudulent online orders, with the goal of predicting fraud in future orders. In K-Nearest Neighbors (K-NN), data points are classified or predicted based on the majority class or average value of their nearest neighbors in the feature space *(IBM).* The Naive Bayes model calculates the probability of a data point belonging to a particular class based on the conditional probabilities of its features *(Javatpoint)*.

Analysis through data exploration and preprocessing of the dataset is also covered.

Data exploration

The dataset is a batch of 1,000,000 online payment records, classified as either legitimate or fraudulent transactions. The dataset contains 7 features and 1 classifier:

- **distance_from_home**

The distance from the customers' home where the purchase was made. Metric is not described.

- **distance_from_last_transaction**

Either the distance between the last transaction made at the retailer or the customers previous transaction. As there are no null values for this column it is assumed to be the former, or some data processing has been done to remove zero values where it’s the customers’ first order. Metric is assumed Euclidean distance, but again is not described.

- **ratio_to_median_purchase_price**

Ratio metric of the transaction’s price versus the median purchase price for that retailer.

- **repeat_retailer**

Whether or not the customer has purchased from that retailer before. Metric is binary.

- **used_chip**

Whether or not the chip of the card was used when payment was processed. Metric is binary.

- **used_pin_number**

Whether or not the pin number was entered when payment was processed. Metric is binary.

- **online_order**

Whether or not the order was placed online. Metric is binary.

- **fraud**

The classifier, whether or not the order was fraudulent. Metric is binary.

ML minor, Report assignment1
Finn Thomas - 722051
Robin Tollenaar - 672023

To start, we will generate some histograms on the first 3 features to analyse their distribution. This will help with understanding the spread of the data and what preprocessing steps might be necessary to normalise the data.

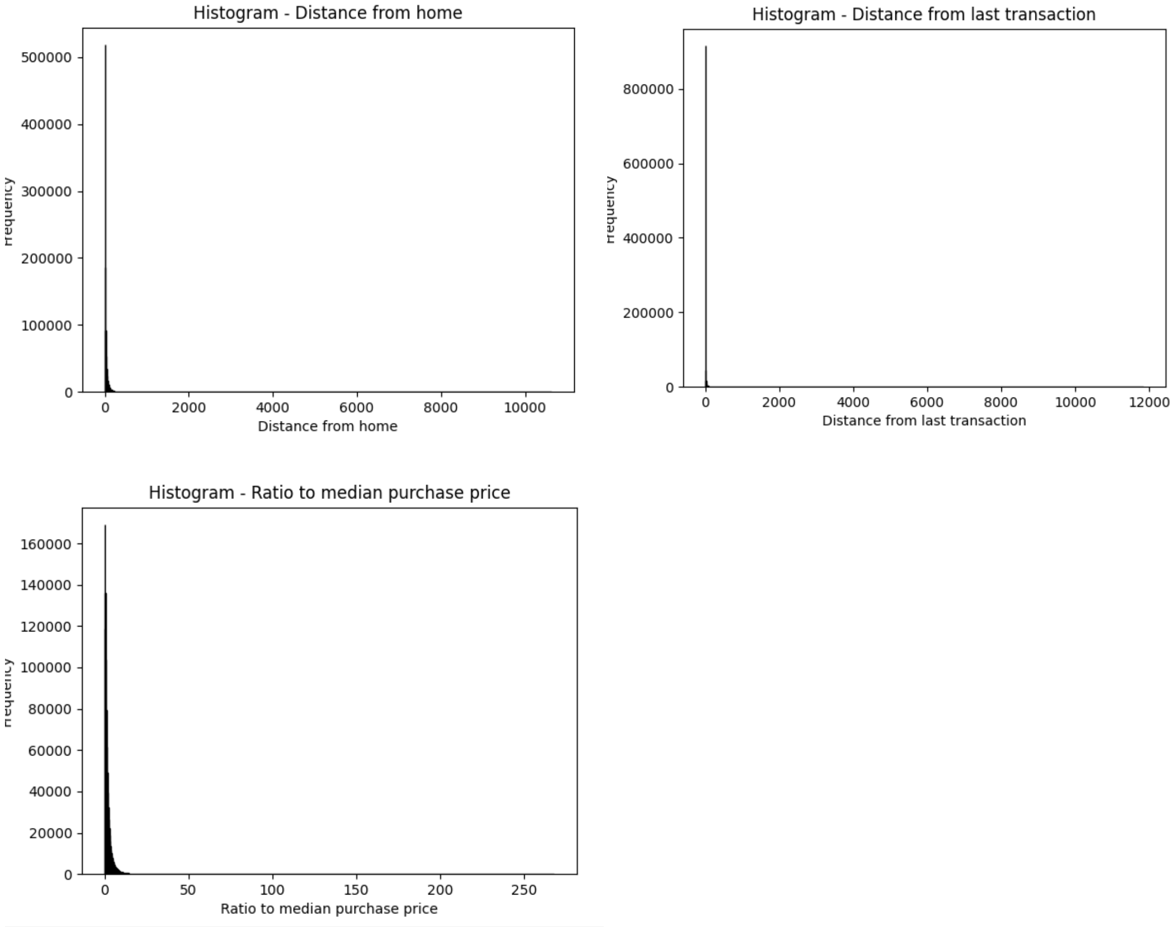

Figure 1: Histograms of the distance features

As we can see, the data is heavily right-skewed. This will need to be normalized later in the preprocessing stage. Due to the variance, it is also difficult to visualize the data. To bring it closer to a normal distribution, we can logarithmically transform the data. Logarithmically transforming a dataset that is heavily right-skewed can have several benefits in data analysis and modeling. By making the distribution more symmetrical and closer to a normal distribution, log-transformed data can be easier to visualize. It can help reveal patterns and relationships that might not be apparent in the original, skewed data. For example:

ML minor, Report assignment1
Finn Thomas - 722051
Robin Tollenaar - 672023

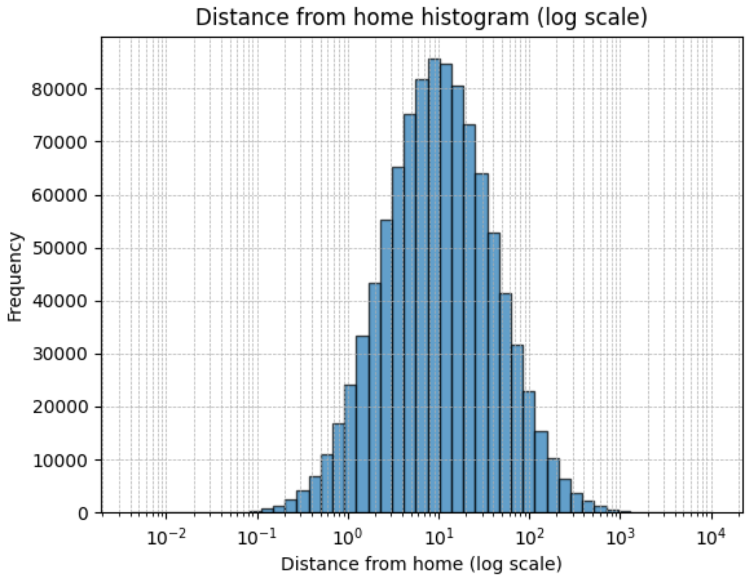

Figure 2: Histogram of the distance from home feature in log scale

Finally, we will use the pandas library correlations method to see if there are any correlations between the features and the fraud indicator. This will help show which features might indicate an order being fraudulent. Below is an excerpt from the matrix calculation:

	fraud
distance_from_home	0.187571
distance_from_last_transaction	0.091917
ratio_to_median_purchase_price	0.462305
repeat_retailer	-0.001357
used_chip	-0.060975
used_pin_number	-0.100293
online_order	0.191973
fraud	1.000000

Although none of the features are (by themselves) strongly correlated with an order being fraudulent (at least 0.6 or -0.6 would indicate a good correlation), we do see some potential indicators. A high distance from home value and an order being an online one have some weak correlation with it being fraud. A high ratio to median purchase price has a moderate correlation, indicating that fraudulent orders tend to be higher value than the median for that retailer.

ML minor, Report assignment1
Finn Thomas - 722051
Robin Tollenaar - 672023

Preprocessing

Preprocessing of the binary data is not necessary, as this already has a fair distribution for a model to work with. The other variables are quantitative, and vary from numbers smaller than 1 up to numbers bigger than 50. The data is standardized by removing the mean and scaling to unit variance.

z = (x - u) / s
Where u is the mean of the training samples, and s is the standard deviation of the training samples.

After this step, the normalized data and the binary data is combined again, and the data is now ready for use.

The models

A model needs data to train. This data has been explored and preprocessed as necessary. Now, this data is ready to be fed to a model. For the model to be properly assessed, it requires testing data. To get this, the original data must be split into training and testing data. This is done with a simple train_test_split function of sklearn. With a test size of 0.2, the model has enough data to learn, and the testing data is big enough to represent the data well.

K-nearest neighbor model

The sklearn library provides a KNeighborsClassifier class to use for this type of scenario. The initialization requires just a n_neighbors parameter, this indicates the number of neighbors to use by default for kneighbors queries. A higher n_neighbors would result in a more thorough query, but it also costs a lot more processing power and time. Then, the model can train by calling the fit method with the training data as parameters.

Naïve Bayes model

The initialization of the Naïve Bayes model is very similar to the k-nearest neighbor model. Sklearn also provides a class for this model, but doesn't require any parameters. The specific model used for this problem is the Gaussian Naïve Bayes, since the data this problem works with is only binary and quantitative. The model is trained with the fit method, which also uses the training data as input.

Testing

an explanation of how the models were tested on a new example and results of these tests

Testing a model's accuracy is important to knowing the success of the training. A high accuracy can show a good grasp of the data. A low accuracy may indicate a flaw in one of the previous steps. In order to test the models used, a simple testing method can be used. Both the models are capable of predicting values based on data. Using the testing data gained in the previous step, it's possible to get the predicted outcomes. If these outcomes are compared to the actual answer, an accuracy can be calculated. Sklearn computes the Accuracy score by counting the number of correct predictions and dividing that by the total number of samples. The parameters used are; a testing size of 20%, a k-score of 5. The score gained in this project is; 0.999 for k-nearest neighbor and 0.951 for naïve Bayes.

Conclusion

In conclusion, this report delves into machine learning models, specifically K-Nearest Neighbors (K-NN) and Naive Bayes, applied to the detection of online fraud. The primary aim was to predict fraudulent transactions accurately using these models.

To start, an in-depth dataset analysis was done, revealing right-skewed features in the distance metric features. Suggestions were made to improve the legibility of the data through logarithmic transformations. Additionally, correlation analysis hinted at potential indicators of fraud, albeit with moderate or weak correlations.

Preprocessing mainly involved standardizing quantitative data, ensuring consistency in our dataset. Model training ensued, where both K-NN and Naive Bayes were trained on the processed training data.

Finally, testing was conducted to assess model performance, with accuracy as the primary metric. K-NN achieved an impressive accuracy score of 0.999, while Naive Bayes scored 0.951, indicating their effectiveness in identifying fraudulent transactions.

In summary, our exploration showcased the potential of machine learning models in combating online fraud, providing valuable tools for enhancing security and mitigating financial loss.

Discussion

When it comes to problems like this, there are always a few ways to improve the quality of the model. With more time and data, the accuracy could improve more.

References

IBM. “What is the k-nearest neighbors algorithm?” *IBM*, 2023, https://www.ibm.com/topics/knn. Accessed 15 September 2023.

ML minor, Report assignment1
Finn Thomas - 722051
Robin Tollenaar - 672023

Javatpoint. “Naive Bayes Classifier in Machine Learning.” *Javatpoint*, 2021, https://www.javatpoint.com/machine-learning-naive-bayes-classifier. Accessed 15 September 2023.