Documentation

This documentation provides a detailed explanation of the steps involved in the credit card fraud detection project.

1. Data Loading and Examination

In this phase, the dataset used for the project is loaded. The dataset is loaded using the **pandas** library, a widely-used library in the Python programming language. The dataset is named "Banksim.csv".

Firstly, the first 5 rows of the dataset are displayed to quickly examine its contents, providing an overview of the variables and sample observations.

	category	age	gender	merchant	amount	fraud
0	es_transportation	4	M	M348934600	4.55	0
1	es_transportation	2	M	M348934600	39.68	0
2	es_transportation	4	F	M1823072687	26.89	0
3	es_transportation	3	M	M348934600	17.25	0
4	es_transportation	5	M	M348934600	35.72	0

Figure (1): Examination of the first 5 rows of the dataset

Subsequently, the **info()** function is used to obtain general information about the dataset. This includes the types of columns, presence of null values, and memory usage.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594643 entries, 0 to 594642
Data columns (total 6 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   category  594643 non-null  object
 1   age       594643 non-null  object
 2   gender    594643 non-null  object
 3   merchant  594643 non-null  object
 4   amount    594643 non-null  float64
 5   fraud     594643 non-null  int64
dtypes: float64(1), int64(1), object(4)
memory usage: 27.2+ MB
```

Figure (2): Examination of general information about the dataset

This step is crucial for understanding the dataset and identifying its important features, which will aid in planning further operations on the dataset.

2. Comparison of Fraud and Non-Fraud Data

In the third stage of the project, two separate dataframes are created for fraudulent and non-fraudulent transactions. The primary purpose is to examine the differences between these two datasets and identify factors affecting fraudulent activities.

Firstly, a dataframe containing non-fraudulent cases (df_non_fraud) and another containing fraudulent cases (df_fraud) are created. This step facilitates further analysis and visualization to understand how fraudulent examples differ from other examples.

Subsequently, the Seaborn library is used to compare fraudulent and non-fraudulent orders. By comparing two selected columns representing different situations, a graph showing the difference between the number of fraudulent transactions and normal purchase transactions is obtained.

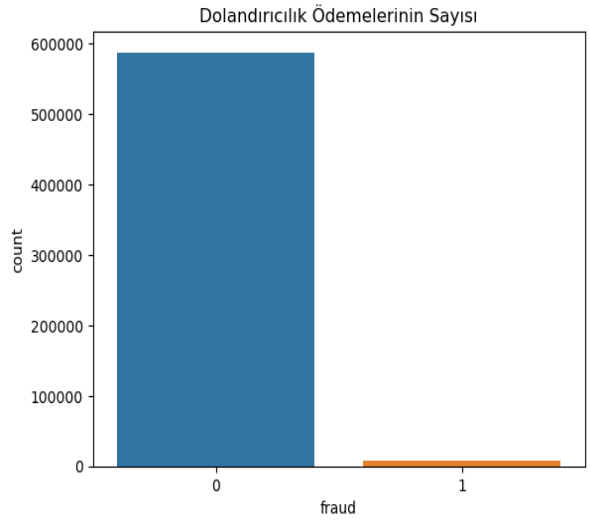

Figure (3): Graphical representation of the number of fraudulent events

To examine in which category fraudulent events occur more frequently, the number of categories in the dataframe containing fraudulent cases is calculated, and a Box Plot graph is created. This graph displays the distribution of fraudulent events by categories.

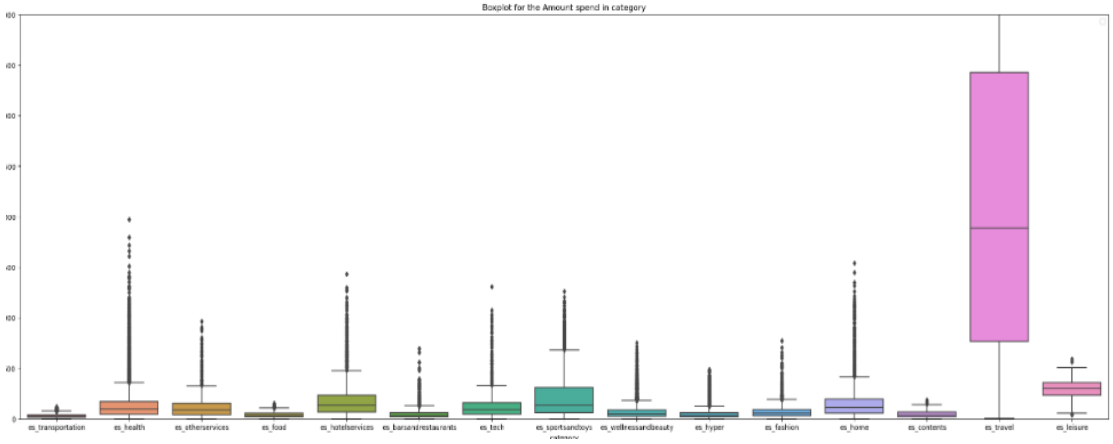

Figure (4): Graphical representation of which category fraudulent events occur more frequently

Above the category, the average amount and fraud percentage are displayed. According to the results, entertainment and travel are the most preferred categories for fraudsters. Fraudsters seem to have chosen categories where people spend more on average.

Below is the graph showing the amount of fraudulent transactions in which fraudulent cases occur more frequently.

Mean feature	amount	fraud
category		
es_barsandrestaurants	43.461014	0.018829
es_contents	44.547571	0.000000
es_fashion	65.666642	0.017973
es_food	37.070405	0.000000
es_health	135.621367	0.105126
es_home	165.670846	0.152064
es_hotelservices	205.614249	0.314220
es_hyper	45.970421	0.045917
es_leisure	288.911303	0.949900
es_otherservices	135.881524	0.250000
es_sportsandtoys	215.715280	0.495252
es_tech	120.947937	0.066667
es_transportation	26.958187	0.000000
es_travel	2250.409190	0.793956
es_wellnessandbeauty	65.511221	0.047594

Figure (5): Examining which category of fraud incidents occur more frequently

They generally appear to have made fewer transactions. Lower-value transactions may be easier for fraudsters to use without attracting attention and abusing the system. Therefore, fraudsters often prefer smaller transactions.

According to the results, entertainment and travel are the most preferred categories for fraudsters. Fraudsters seem to have chosen categories where people spend more on average.

The graph below examines the frequency of fraudulent events occurring at different amounts.

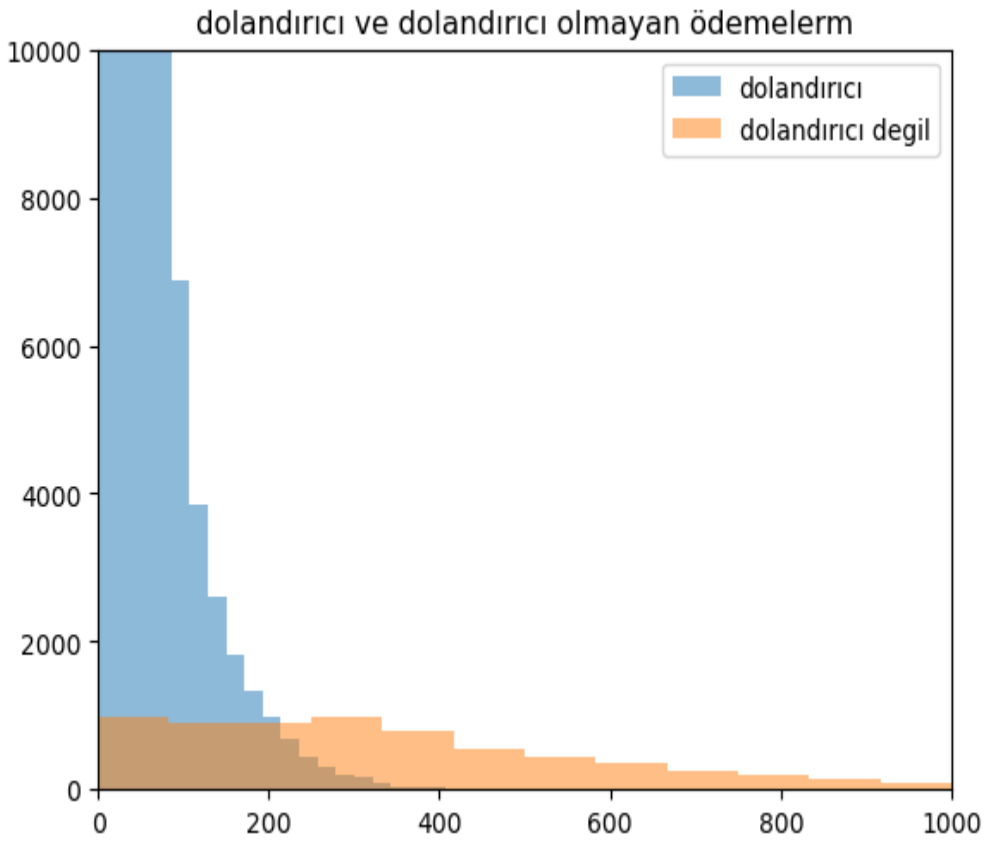

Figure (6): Graphical representation of the frequency of fraudulent events occurring at different amounts

3. Data Transformation and Determination of Independent/Dependent Variables

In this step, columns of object type are transformed into categorical to facilitate the data transformation process. This process aims to prepare the dataset for use with machine learning models. Subsequently, the independent variable (X) and dependent/target variable (y) are defined.

4. Balancing of Samples

As seen earlier, there is a considerable difference in the number of fraudulent and non-fraudulent events. Fraud data will be imbalanced, which can be understood both from the graph and the number of samples. Oversampling or undersampling techniques can be applied to balance the dataset. Oversampling involves increasing the number of samples from the minority class by generating samples from the minority class. Undersampling involves reducing the number of samples in the majority class by randomly selecting points, thereby equalizing it with the minority class. Both processes have some risks: Oversampling will create duplicate or similar data points, which may not always be helpful in fraud detection because fraudulent transactions may vary. Undersampling will cause us to lose data points and therefore information. In this project, an oversampling technique called SMOTE (Synthetic Minority Over-sampling Technique) has been applied to balance the data. SMOTE uses neighboring examples to create new data points from the minority class, so the generated examples are similar, although not exact copies.

5. Implementation of Machine Learning Algorithms

Various machine learning algorithms have been applied to the balanced dataset. These algorithms include Artificial Neural Network, Gaussian Naive Bayes, K-Nearest Neighbors (KNN), and XGBoost. These algorithms are models based on different principles used for fraud detection. Performance metrics such as accuracy, confusion matrix, and classification report have been examined using training and test datasets.

6.Findings

The numerical results of the implemented algorithms are shown in the table below:

Algorithm	Accuracy	Precision	Recall	F1 Score
ANN	0.974	0.972	-	-
GNB	0.92	0.92	0.92	0.92
KNN	0.99	0.99	0.99	0.99
XGBoost	0.99	0.99	0.99	0.99

Table (1): Numerical Results of the Algorithms

The Artificial Neural Network (ANN) model has performed well with the decrease in loss and increase in accuracy during training. The model achieved 97.25% accuracy during training and 97.42% accuracy on the validation set.

The Gaussian Naive Bayes model has an F1 score of 0.92 for class 0 and 0.92 for class 1. Precision and recall values are 0.91, 0.93, and 0.93, 0.91, respectively. The confusion matrix indicates that the model tends to miss some examples in class 1.

The K-Nearest Neighbors model has shown a highly successful performance with high accuracy rate (99%) and low error rates. Precision and recall values are high, and the rate of missing examples in class 1 is low.

Based on these results, it can be said that the K-Nearest Neighbors model performs better than the other two models.

The XGBoost (eXtreme Gradient Boosting) model has exhibited high performance by achieving 0.99 precision and 0.98 recall for class 0, and 0.98 precision and 0.99 recall for class 1. The F1 scores are 0.99, and the accuracy rate is determined as 99%. These results indicate that XGBoost is an effective model for fraud detection.