- **Yassine Sfaihi**
- www.linkedin.com/in/yassinesfaihi
- www.github.com/yassinesfaihi
- www.kaggle.com/yassinesfaihi

[Author] : Yassine Sfaihi

Fraudulent transactions detection with Keras and TensorFlow: Hyperparameter Tuning and Weight Regularization for Improved Model Performance

INTRODUCTION

This notebook performs several operations on a credit card dataset, including data analysis, data cleaning, training a binary classification model, and hyperparameter tuning.

The **tools used** in this notebook include:

- Pandas
- Seaborn
- Matplotlib
- Numpy
- Tensorflow
- Keras
- Scikit-learn

Data Analysis

The first part of this notebook is dedicated to analyzing the credit card dataset. The following operations are performed:

1. Reading the dataset from a csv file using Pandas.
2. Printing an overview of the data.
3. Printing the data types of columns.
4. Printing the descriptive statistics of the dataset.
5. Counting and calculating the percentage of the two classes in the target variable.
6. Plotting the count of each class in the target variable.
7. Plotting the distribution of the "Amount" column for each class.
8. Plotting the box plot of the "Amount" column for each class.
9. Plotting histograms for all columns.
10. Checking for missing values in the dataset.
11. Removing outliers from the dataset.
12. Checking if there are any NaN or infinite values in the dataset.

Model Training and Evaluation

The second part of this notebook is dedicated to training a binary classification model and evaluating its performance. The following steps are performed:

1. Splitting the data into training and validation data.

- **Yassine Sfaihi**
- www.linkedin.com/in/yassinesfaihi
- www.github.com/yassinesfaihi
- www.kaggle.com/yassinesfaihi

2. Normalizing the training and validation data.
3. Defining a 2-layer neural network model using Keras.
4. Training the model using the Adam optimizer and binary crossentropy loss.
5. Evaluating the model on the validation data and calculating precision, recall, and F1-score.
6. Visualizing the model loss and confusion matrix.

Hyperparameter Tuning

The third part of this notebook is dedicated to hyperparameter tuning. The following steps are performed:

1. Defining the grid search parameters.
2. Wrapping the neural network model using KerasClassifier from scikit-learn.
3. Performing the grid search using the GridSearchCV class from scikit-learn.
4. Finding the best hyperparameters for the model.
5. Printing the best hyperparameters.

Model Performance Visualization

The final part of this notebook is dedicated to visualizing the performance of the trained model. The following steps are performed:

1. Computing class weights for the training targets.
2. Compiling the model using the Adam optimizer and binary crossentropy loss.
3. Training the model using the training data.
4. Evaluating the model on the validation data.
5. Visualizing the model performance using loss plots, confusion matrix, ROC curve, and precision-recall curve.

Summary

This project demonstrates the development and evaluation of a binary classification model for a credit card dataset. The model is built using Keras and TensorFlow, with hyperparameter tuning and weight regularization for improved performance. The model is evaluated using various performance metrics and visualizations, and the best hyperparameters are found using a grid search.