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

[Author] : Yassine Sfaihi

Combating Fraud with Machine Learning and Resampling Techniques for Imbalanced Data

Introduction

The goal of this project is to detect fraud transactions using machine learning algorithms. The code includes the implementation of 4 oversampling techniques (**SMOTE, ADASYN, SMOTEENN** and **ENN**) and compares their results on a set of models.

Tools and Libraries Used

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Sklearn
- Imblearn

Data Preprocessing

The data was preprocessed and cleaned to prepare it for the analysis. This involved performing exploratory data analysis (EDA), handling missing values, and transforming the features as necessary.

Oversampling Techniques

The following imblearn techniques were used to address the class imbalance problem in the data:

1. **SMOTE** stands for Synthetic Minority Over-sampling Technique. It is an oversampling method used in imbalanced learning problems to balance the class distribution by generating synthetic samples for the minority class. SMOTE creates these synthetic samples by interpolating between existing minority samples and adding them to the data set.
2. **SMOTEENN**: A combination of the Synthetic Minority Over-sampling Technique (SMOTE) and the Edited Nearest Neighbours (ENN) algorithm. It oversamples the minority class and cleans the synthetic data generated by SMOTE by removing the noisy instances.

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

3. **ADASYN**: Adaptive Synthetic (ADASYN) oversampling method. This method generates synthetic samples to balance the class distribution.
4. **EditedNearestNeighbours**: A method for removing samples from the majority class. The method removes instances whose nearest neighbors are all of the same class.

Models

The following models were used in this analysis:

1. Logistic Regression
2. Decision Tree Classifier
3. Gaussian Naive Bayes
4. Random Forest Classifier
5. Extra Trees Classifier

Results

The results of the analysis can be found in the Precision-Recall curves for each model, both with and without using the oversampling techniques. The evaluation results for each model, both with and without oversampling, are also provided for different metrics.