REGex Software Services - DataScience - ML & DL Winter Internship/Training Program

Project Name: "Credit Card Fraud Detection using ML"

Problem Statement:

Assume that you are employed to help a credit card company to detect potential fraud cases so that the customers are ensured that they won't be charged for the items they did not purchase. You are given a dataset containing the transactions between people, the information that they are fraud or not, and you are asked to differentiate between them. This is the case we are going to deal with. Our ultimate intent is to tackle this situation by building classification models to classify and distinguish fraud transactions.

Steps Involved

1. Importing the required packages into our python environment.
2. Importing the data
3. Processing the data to our needs and Exploratory Data Analysis
4. Feature Selection and Data Split
5. Building six types of classification models
6. Evaluating the created classification models using the evaluation metrics

We are using python for this project because it is really effortless to make use of a bunch of methods, has an extensive amount of packages for machine learning, and can be learned easily. In recent days, the job market for python is seamlessly higher than any other programming language and companies like Netflix are using python for data science and many other applications. With that, let's dive into the coding part.

Dataset Used for: The data we are going to use is the Kaggle Credit Card Fraud Detection dataset (click here for the dataset). It contains features V1 to V28 which are the principal components obtained by PCA. We are going to neglect the time feature which is of no use to build the models. The remaining features are the 'Amount' feature that contains the total amount of money being transacted and the 'Class' feature that contains whether the transaction is a fraud case or not.

Context

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

Content

The datasets contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-senstive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Libraries used for : For this project, our primary packages are going to be Pandas to work with data, NumPy to work with arrays, scikit-learn for data split, building and evaluating the classification models, and finally the xgboost package for the xgboost classifier model algorithm.

Pandas : pandas is a **Python** package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in **Python**.

Numpy : NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Scikit-learn : Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency. It has minimal dependencies and is distributed under the simplified BSD license, encouraging its use in both academic and commercial settings

Xgboost : XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks.