CO 327 – MACHINE LEARNING

DELHI TECHNOLOGICAL UNIVERSITY

PROJECT REPORT

SUBMITTED TO:
DR. RUCHIKA MALHOTRA
Associate Professor and Associate Head
Department of Software Engineering

SUBMITTED BY:
NAME: ASHUTOSH GUPTA
ROLL NO: 2K16/SE/020
NAME: SIDHARTH BANSAL
ROLL NO: 2K16/SE/080

Credit Card Fraud Detection

Introduction

Throughout the financial sector, machine learning algorithms are being developed to detect fraudulent transactions. In this project, that is exactly what we are going to be doing as well. Using a dataset of nearly 28,500 credit card transactions and multiple unsupervised anomaly detection algorithms, we are going to identify transactions with a high probability of being credit card fraud. In this project, we will build and deploy the following two machine learning algorithms:

- Local Outlier Factor (LOF)
- Isolation Forest Algorithm

Furthermore, using metrics suchs as precision, recall, and F1-scores, we will investigate why the classification accuracy for these algorithms can be misleading.

In addition, we will explore the use of data visualization techniques common in data science, such as parameter histograms and correlation matrices, to gain a better understanding of the underlying distribution of data in our data set.

Local Outlier Factor (LOF)

The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood.

Isolation Forest Algorithm

The IsolationForest ‘isolates’ observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

Since recursive partitioning can be represented by a tree structure, the number of splittings required to isolate a sample is equivalent to the path length from the root node to the terminating node.

This path length averaged over a forest of such random trees, is a measure of normality and our decision function.

Random partitioning produces noticeably shorter paths for anomalies. Hence, when a forest of random trees collectively produces shorter path lengths for particular samples, they are highly likely to be anomalies.

Literature Review

The statistical approach includes normal, Kai, F, student-t, Poisson, alpha, gamma distributions, etc . Sometimes, more than one statistical distributions can fit the dataset. However, for most cases, this distribution is not known. Therefore, the statistical approach is not appropriate in our dataset. This has been discussed by M. Gupta, J. Gao, C.C. Aggarwal and J. Han, "Outlier Detection for Temporal Data: A Survey"

The main idea of the distance-based approach is to determine an outlier to its neighborhood by the Euclidean distance.The distance-based approach, including the k-nearest neighbor (KNN) method, is usually employed when the data does not fit any distribution, and a model generating mechanism is not required in this approach. This is discussed by V. Chandola, A. Banerjee, and V. Kumar, "Outlier Detection: A Survey,"

The reason we choose LOF over KNN is because LOF computes the relative density of each data point while kNN only calculates the sum of distances to each neighbor . So , LOF gives more accurate results than KNN.

And the reason we choose Isolation Forest is that Isolation Forest explicitly identifies anomalies instead of profiling normal data points. Isolation Forest, like any tree ensemble method, is built on the basis of decision trees. In these trees, partitions are created by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature. This is suggested in S. Wu, S. Wang, "Information-Theoretic Outlier Detection for Large-Scale Categorical Data", IEEE Trans. KDE, 25(3), pp.589, 2013

In principle, outliers are less frequent than regular observations and are different from them in terms of values (they lie further away from the regular observations in the feature space). That is why by using such random partitioning they should be identified closer to the root of the tree (shorter average path length, i.e., the number of edges an observation must pass in the tree going from the root to the terminal node), with fewer splits necessary.

Considering the characteristics of streaming data, an iForest-based anomaly detection framework is firstly proposed under the sliding windows framework and a new streaming data anomaly detection algorithm, namely iForestASD, is proposed. The experiments results on four real world data sets demonstrate that proposed method is efficient. This is the observation discussed by "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window"

By reading multiple research papers, we came to the conclusion that Local Outlier Factor and Isolation Forest Algorithm are the two algorithms which we should implement for the credit card fraud detection.

Implementation

```
#Importing Necessary Libraries
import sys
import numpy
import pandas
import matplotlib
import seaborn
import scipy

# import the necessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset from the csv file using pandas
data = pd.read_csv('creditcard.csv')

# Determine number of fraud cases in dataset
Fraud = data[data['Class'] == 1]
Valid = data[data['Class'] == 0]
outlier_fraction = len(Fraud)/float(len(Valid))
print(outlier_fraction)
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))

# Correlation matrix
corrmat = data.corr()
```

```
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corrmat, vmax = .8, square = True)
plt.show()

# Get all the columns from the dataFrame
columns = data.columns.tolist()

# Filter the columns to remove data we do not want
columns = [c for c in columns if c not in ["Class"]]

# Store the variable we'll be predicting on
target = "Class"

X = data[columns]
Y = data[target]

from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor

# define random states
state = 1

# define outlier detection tools to be compared
classifiers = {
    "Isolation Forest": IsolationForest(max_samples=len(X),
                                        contamination=outlier_fraction,
                                        random_state=state),
    "Local Outlier Factor": LocalOutlierFactor(
        n_neighbors=20,
        contamination=outlier_fraction)}
# Fit the model
plt.figure(figsize=(9, 7))
n_outliers = len(Fraud)


for i, (clf_name, clf) in enumerate(classifiers.items()):

    # fit the data and tag outliers
    if clf_name == "Local Outlier Factor":
        y_pred = clf.fit_predict(X)
        scores_pred = clf.negative_outlier_factor_
    else:
        clf.fit(X)
        scores_pred = clf.decision_function(X)
        y_pred = clf.predict(X)

    # Reshape the prediction values to 0 for valid, 1 for fraud.
```

```
    y_pred[y_pred == 1] = 0
    y_pred[y_pred == -1] = 1

    n_errors = (y_pred != Y).sum()

    # Run classification metrics
    print('{}: {}'.format(clf_name, n_errors))
    print(accuracy_score(Y, y_pred))
    print(classification_report(Y, y_pred))
```

OUTPUT:

```
0.00172341024198
Fraud Cases: 49
Valid Transactions: 28432
Local Outlier Factor: 97
0.9965942207085425
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.02      0.02      0.02        49

avg / total       1.00      1.00      1.00     28481

Isolation Forest: 71
0.99750711000316
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.28      0.29      0.28        49

avg / total       1.00      1.00      1.00     28481
```

Result: We have successfully implement Credit Card Fraud Detection. The LOF algorithm has higher accuracy of 97% then Isolation Algorithm which has accuracy of 71%. But the F-measure of Isolation algorithm is greater than LOF.

Conclusion: We discussed many approaches, and many algorithms like K Means Clusturing, K-Nearest Neighbours, etc. We implemented LOF and Isolation Forest Algorithm. We came to the conclusion that isolation forest algorithm gives the most accurate results for Credit Card Fraud Detection. We also came to know that the accuracy is a false measure for imbalanced data. F-measure is a fair estimate for checking the validity of the model.

References:

- S. Wu, S. Wang, “Information-Theoretic Outlier Detection for Large-Scale Categorical Data”, IEEE Trans. KDE, 25(3), pp.589, 2013
- M. Gupta, J. Gao, C.C. Aggarwal and J. Han, “Outlier Detection for Temporal Data: A Survey”, IEEE TKDE, 26(9), pp. 2250-2267, 2014.
- C.C. Aggarwal, P.S. Yu, “Outlier Detection for High Dimensional Data”; ACM SIGMOD, pp. 37-46, 2001.
- V. Chandola, A. Banerjee, and V. Kumar, "Outlier Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1-58, 2009.
- Mathew X. Ma , Henry Y.T. Ngan and Wei Liu; Department of Mathematics, Hong Kong Baptist University, Hong Kong, Communication Research Group, Department of Electronic & Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K.
- https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e
- https://www.ijraset.com/fileserve.php?FID=12493
- G. Hulten, L. Spencer, and P. Domingos. (2001). Mining time changing data streams. In Proceedings of the 7th ACM SIGKDD. He, Z., Xu, X., Deng, S. , (2003). Discovering cluster-based local outliers.
- Pattern Recognition Letters 24(9-10), 1641-1650. Kavitha, C. (2012). Massive stream data processing to attain anomaly Intrusion Prevention Devices, Circuits and Systems, In 2012 ICDCS, 15-16 March, 572 -575.
- K. Yamanishi and J. Takeuchi. (2006). A unifying framework for detecting outliers and change points from non-stationary time series data. Knowledge and Data Enginnering, 18(4): 482-492.
- K. Yamanishi, J.I. Takeuchi, G.Williams, and P.Milne. (2000). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In Proceeding of the sixth ACM SIGKDD International conference on Knowledge discovery and data mining. 320-324.