

राजीव गांधी पेट्रोलियम प्रौद्योगिकी संस्थान
(संसद के अधिनियम के अधीनस्थापित राष्ट्रीय महत्व का एक संस्थान)
Rajiv Gandhi Institute of Petroleum Technology
(An Institute of National Importance established under an Act of Parliament)
Jais, Amethi – 229304, UP, India. Website: www.rgipt.ac.in



BTP Report

Anomaly Detection in Multivariate Time-Series

Submitted by:
Vijeet Nigam
20CS3068

Submitted to:
Dr. Pallabi Saikia
Assistant Professor
Department of Computer Science and Engineering

Content

List of Symbols	
List of Figures	
List of Tables	
List of Abbreviations	
INTRODUCTION: Outlier Detection and Analysis Methods	1
What is an outlier in machine learning?	1
Types of Outliers	
Point Outlier	2
Contextual Outlier	2
Collective Outlier	3
Detecting Anomalies	5
Problem Statement 1: ECG	5
Autoencoders	5
Analysis of Model 1	6
Accuracy of Model 1	7
Problem Statement 2: Expedia Hotel Search	8
K Means	9
3D Cluster	9
PCA	10
Isolation Forest	11
One Class SVM	11
Gaussian Distribution	12
Analysis of Model 2	13
Concluding Discussion	14
My Contribution	15
Code	15
Data Sets	15
REFERENCES	16

List of Symbols

Gaussian/Normal Distribution Formula	12
--------------------------------------	----

$$p(x) = p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi} \sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

List of Figures

Fig. 1	Point Outlier	2
Fig. 2	Contextual Outlier	3
Fig. 3	Collective Outlier	3
Fig. 4	Autoencoders	5
Fig. 5	n_test_data plot	6
Fig. 6	Accuracy of My Autoencoders Model – Code Screenshot	7
Fig. 7	Expedia Hotel Search Data Set	8
Fig. 8	K Means	9
Fig. 9	3D Cluster	9
Fig. 10	PCA	10
Fig. 11	Isolation Forest	11
Fig. 12	One Class SVM	11
Fig. 13	Observation of algorithms (K-Means, 3D Cluster, PCA, Gaussian Distribution, and Isolation Forest) - Code Screenshot	13
Fig. 14	Observation of One-Class SVM algorithm – Code Screenshot	13

List of Tables

N/A

List of Abbreviations

PCA	Principal Component Analysis
SVM	Support Vector Machine
ECG	Electrocardiogram
ANN	Artificial Neural Network

INTRODUCTION: Outlier Detection and Analysis Methods

Outlier detection is a key consideration within the development and deployment of machine learning algorithms. Models are often developed and leveraged to perform outlier detection for different organisations that rely on large datasets to function. Economic modelling, financial forecasting, scientific research, and e-commerce campaigns are some of the varied areas that machine learning-driven outlier detection is used.

Identifying and dealing with outliers is an integral part of working with data, and machine learning is no different. Algorithm development usually relies on huge arrays of training data to achieve a high level of accuracy. Once deployed, models will process huge amounts of data, providing insights into trends and patterns. In this data-rich environment, organisations can expect to have to deal with outlier data. Outliers can skew trends and have a serious impact on the accuracy of models. The presence of outliers can be a sign of concept drift, so ongoing outlier analysis in machine learning is needed.

Machine learning models learn from data to understand the trends and relationship between data points. Outliers can skew results, and anomalies in training data can impact overall model effectiveness. Outlier detection is a key tool in safeguarding data quality, as anomalous data and errors can be removed and analysed once identified.

Outlier detection is an important part of each stage of the machine learning process. Accurate data is integral during the development and training of algorithms, and outlier detection is performed after deployment to maintain the effectiveness of models. This guide explores the basics of outlier detection techniques in machine learning, and how they can be applied to identify different types of outlier.

What is an outlier in machine learning?

An outlier is an individual point of data that is distant from other points in the dataset. It is an anomaly in the dataset that may be caused by a range of errors in capturing, processing or manipulating data. Outliers can skew overall data trends, so outlier detection methods are an important part of statistics. Outliers will be a consideration for any area that uses data to make decisions. If an organisation is gaining insight from data, outliers are a real risk.

Outlier detection is particularly important within machine learning. Models are trained on huge arrays of training data. The model understands the relationship between data points to help predict future events or categorise live data. Outliers in the training data may skew the model, lowering its accuracy and overall effectiveness. Outlier analysis and resolution can lengthen the training time too. Outliers can be present in any data or machine learning use case, whether that's financial modelling or business performance analysis.

Types of Outliers

There are three main types of outliers relevant to machine learning models. Each type differs by how the anomalous data can be observed and what makes the data point stand apart from the rest of the data set. Types are an important consideration for outlier analysis as each has a different pattern to identify.

The three main types of outliers are:

1. Point outliers
2. Contextual outliers
3. Collective outliers

Point Outlier

A point outlier is an individual data point that sits outside of the range of the rest of the dataset. There may be a clear pattern, trend or grouping within the dataset, and an outlier as a data point will be significantly different to this. Point outliers can often be attributed to an error with the measurement or input of the data.

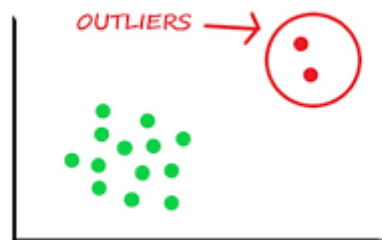


Fig. 1 Point Outlier

For example, an outlier may occur in the health sector data if a mistake is made when recording a unit of measurement within patient data. Missing a digit when recording the height of a patient would cause a noticeable point outlier within the dataset. This type of outlier can be relatively straightforward to identify through visual means. Point outliers are visible if the dataset is plotted across two or three dimensions, as the outlier as a data point would sit far apart from the rest of the dataset.

Contextual Outlier

A contextual outlier is when a data point is significantly different from the dataset, but only within a specific context. The context of a dataset may change seasonally or fluctuate with wider economic trends or behaviour. A contextual outlier will be noticeable when the context of the dataset changes. This could be seasonal weather changes, economic fluctuations, changes in customer behaviour for

key holidays, or even the time of the day. For this reason, a contextual outlier may seem like a normal data point in other contexts.

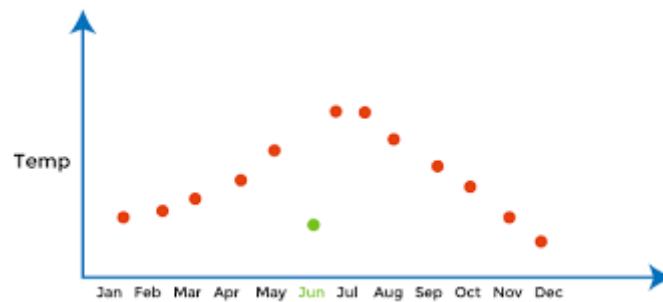


Fig. 2 Contextual Outlier

For example, in a dataset of UK temperatures over time, encompassing different years and seasons. A temperature reading below zero degrees at noon could be seen as normal during the winter. But this same reading would be deemed a contextual outlier if recorded at the height of summer during a heat wave. The data is contextualised within wider trends that are impacting the dataset.

Collective Outlier

A collective outlier is when a series of data points differ significantly from the trends in the rest of the dataset. The individual data points within a collective outlier may not seem like a point outlier or a contextual outlier. It's when the data points are considered as a collection that anomalous patterns are observed. For this reason collective outliers can be the hardest type of outlier to identify. Collective outliers are an integral part of monitoring for concept drift in machine learning. A sequence of data has shifted away from expected behaviour within the model.

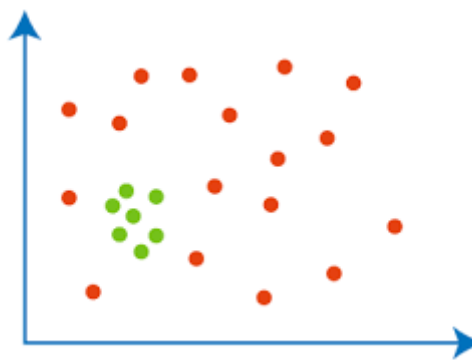


Fig. 3 Collective Outlier

For example, a time series plotting subscribers and unsubscribers to an email marketing list showing seasonal or daily fluctuations. A collective outlier could be flagged if the level of subscribed users stayed entirely static for many weeks with no fluctuation. Individual users unsubscribing and new users subscribing is a normal occurrence so a static count would be flagged as an outlier.

Taken in isolation, each data point is within the expected boundaries of the data so would not be flagged as a contextual or point outlier. But when taken as a series, the data's behaviour is flagged as anomalous. Once a collective outlier is identified, steps can be taken to investigate any systematic errors in the process.

Detecting Anomalies

Simple statistical techniques such as mean, median, quantiles can be used to detect univariate anomalies feature values in the dataset. Various data visualization and exploratory data analysis techniques can be also be used to detect anomalies.

Discussing some machine learning algorithms to detect anomalies that I have used to solve the problem statement 1 & 2-

1. Autoencoders
2. K-Means
3. 3D Cluster
4. PCA
5. One Class SVM
6. Gaussian Distribution
7. Isolation Forest

Problem Statement 1: ECG

[Data Set](#)

An electrocardiogram (ECG) is a simple test that can be used to check your heart's rhythm and electrical activity. Sensors attached to the skin are used to detect the electrical signals produced by your heart each time it beats.

The objective of detecting anomalies in ECG signals consists of finding the irregular heart rates, heartbeats, and rhythms. To achieve this goal, an anomaly detection system must be able to find them on all heartbeat sequences; therefore, to obtain the essential metrics.

Autoencoders

Autoencoder is an important application of Neural Networks or Deep Learning. It is widely used in dimensionality reduction, image compression, image denoising, and feature extraction. It is also applied in anomaly detection and has delivered superior results.

For example, given an image of a handwritten digit, an autoencoder first encodes the image into a lower dimensional representation, then decodes it back to an image. It learns to compress the data while minimizing the reconstruction error.

The notebook "DataSet 1" uses autoencoders for detecting anomaly in ECG(electrocardiogram) readings.

This is one of the very good practical application of autoencoders. Autoencoders are a specific type of feedforward neural network.

It compresses the input into a so-called "code" of lower dimensionality and then tries to reconstruct the output from this code. It is an **unsupervised** learning model.

It consists of two parts:- Encoder and Decoder

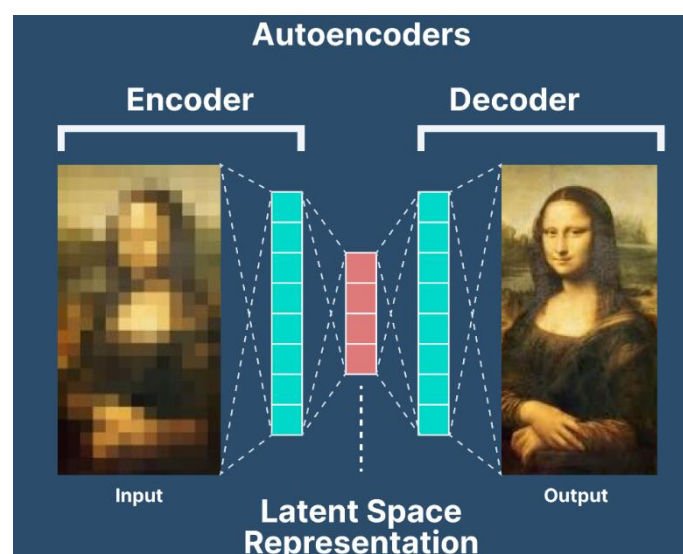


Fig. 4 Autoencoder

Analysis of Model 1

We will create an encoder and a decoder using an ANN architecture. We are going to provide the ECG data as input and the model will try to reconstruct it. The error between the original data and reconstructed output will be called the reconstruction error. Based on this reconstruction error we are going to classify an ECG as anomalous or not. In order to do this we are going to train the model only on the normal ECG data but it will be tested on the full test set so that when an abnormal ECG is provided in the input the autoencoder will try to reconstruct it since it has been only trained on normal ECG data the output will have a larger reconstruction error. We will also define a minimum threshold for the error i.e. if the reconstruction error is above the threshold then it will be categorized as anomalous.

```
In [14]: #Lets see some more result visually !!
plot(n_test_data, 0)
plot(n_test_data, 1)
plot(n_test_data, 3)
```

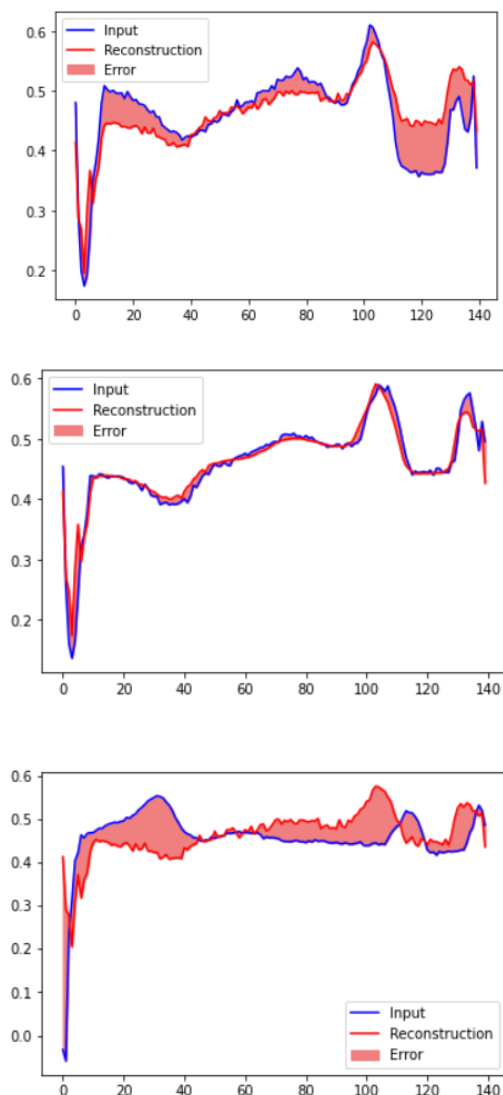
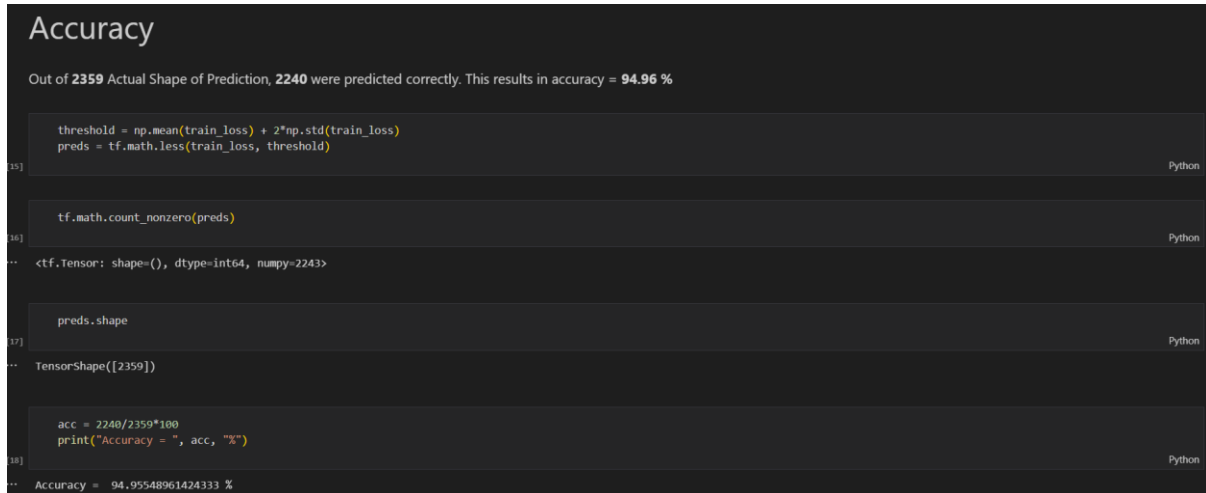


Fig. 5 n_test_data plot

Accuracy of Model 1

The Accuracy of my model comes out to be **94.96 %**



```
Accuracy

Out of 2359 Actual Shape of Prediction, 2240 were predicted correctly. This results in accuracy = 94.96 %

threshold = np.mean(train_loss) + 2*np.std(train_loss)
preds = tf.math.less(train_loss, threshold)

tf.math.count_nonzero(preds)

<tf.Tensor: shape=(), dtype=int64, numpy=2243>

preds.shape

TensorShape([2359])

acc = 2240/2359*100
print("Accuracy = ", acc, "%")

Accuracy = 94.95548961424333 %
```

Fig. 6 Accuracy of My Autoencoders Model - Code Screenshot

Problem Statement 2: Expedia Hotel Search

[Data Set](#)

“Hotel” refers to hotels, apartments, B&Bs, hostels and other properties appearing on Expedia’s websites. Room types are not distinguished and the data can be assumed to apply to the least expensive room type.

Most of the data are for searches that resulted in a purchase, but a small proportion are for searches not leading to a purchase.

So the main objective is to check the prices of Hotel Rooms.

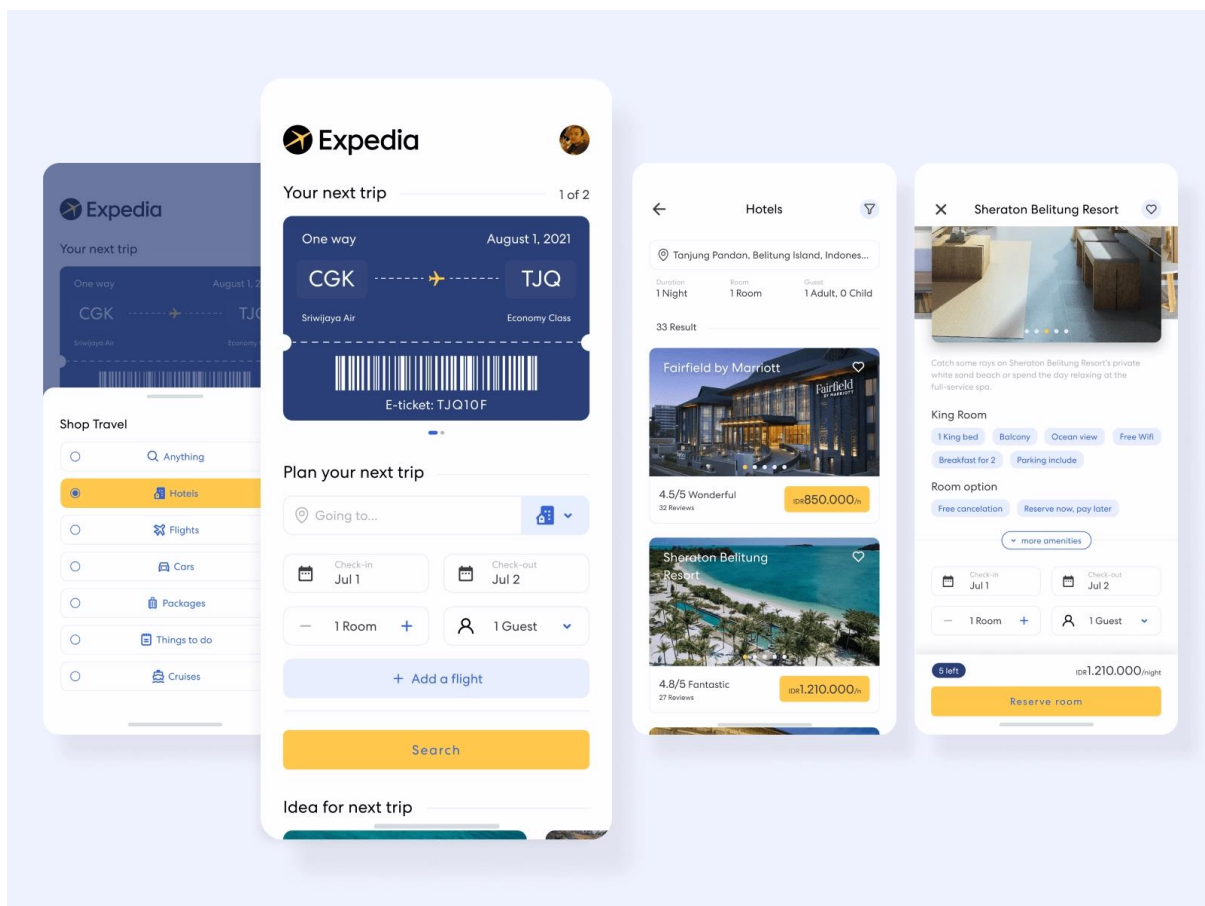


Fig. 7 Expedia Hotel Search Data Set

K Means

This method looks at the data points in a dataset and groups those that are similar into a predefined number K of clusters. A threshold value can be added to detect anomalies: if the distance between a data point and its nearest centroid is greater than the threshold value, then it is an anomaly.

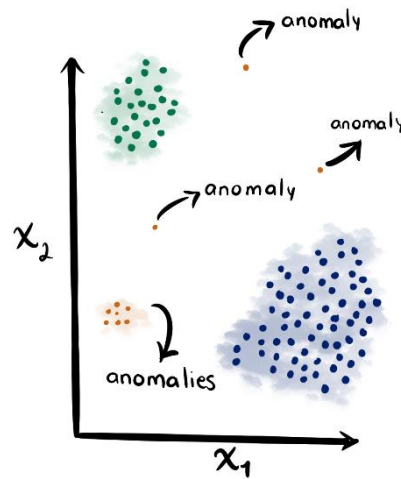


Fig. 8 K Means

The main difficulty resides in choosing K , since data in a time series is always changing and different values of K might be ideal at different times. Besides, in more complex scenarios where there are both local and global outliers, many outliers might pass under the radar and be assigned to a cluster.

3D Cluster

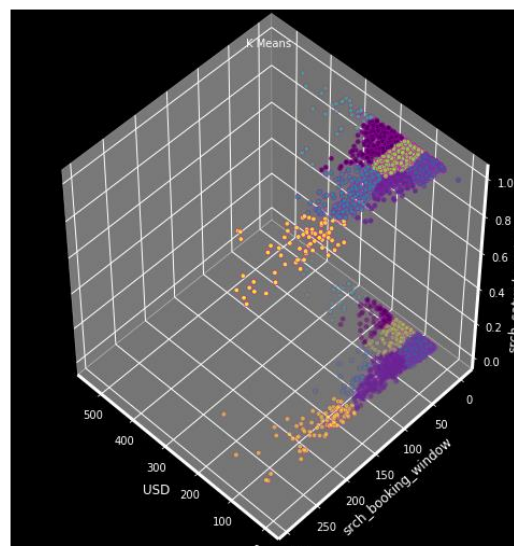


Fig. 9 3D Cluster Model - Code Screenshot

PCA

The PCA-Based Anomaly Detection component solves the problem by analyzing available features to determine what constitutes a "normal" class. The component then applies distance metrics to identify cases that represent anomalies. This approach lets you train a model by using existing imbalanced data.

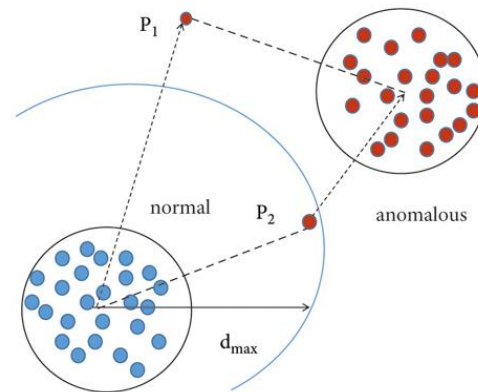


Fig. 10 PCA

More about principal component analysis

PCA is an established technique in machine learning. It's frequently used in exploratory data analysis because it reveals the inner structure of the data and explains the variance in the data.

PCA works by analyzing data that contains multiple variables. It looks for correlations among the variables and determines the combination of values that best captures differences in outcomes. These combined feature values are used to create a more compact feature space called the principal components.

For anomaly detection, each new input is analyzed. The anomaly detection algorithm computes its projection on the eigenvectors, together with a normalized reconstruction error. The normalized error is used as the anomaly score. The higher the error, the more anomalous the instance is.

Isolation Forest

Isolation Forest is an unsupervised anomaly detection algorithm that uses a random forest algorithm (decision trees) under the hood to detect outliers in the dataset. The algorithm tries to split or divide the data points such that each observation gets isolated from the others.

Usually, the anomalies lie away from the cluster of data points, so it's easier to isolate the anomalies compare to the regular data points.

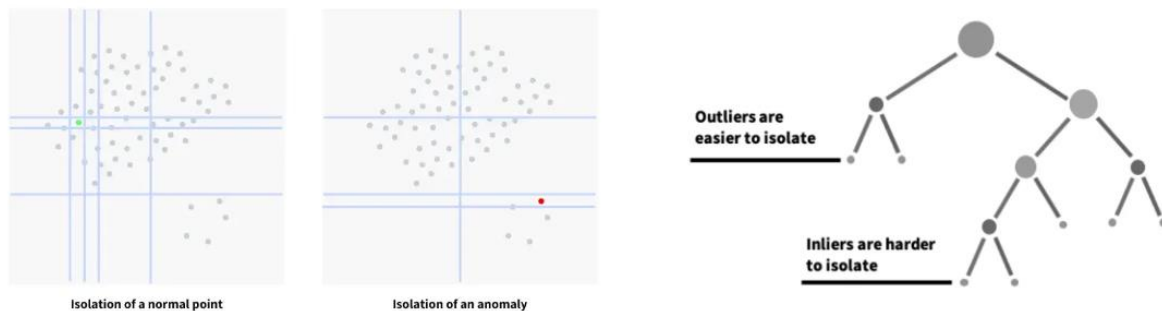


Fig. 11 Isolation Forest

From the above-mentioned images, it can be observed that the regular data points require a comparatively larger number of partitions than an anomaly data point.

The anomaly score is computed for all the data points and the points anomaly score $>$ threshold value can be considered as anomalies.

One Class SVM

An unsupervised setting works well for anomaly detection task. The task is also sometimes referred to as **novelty detection** since we are not given class labels for both the classes.

The intuition behind a one-class SVM

Regular SVM for classification finds a max-margin hyperplane that separates the positive examples from the negative ones. The one-class SVM finds a hyper-plane that separates the given dataset from the ***origin*** such that the hyperplane is as close to the datapoints as possible.

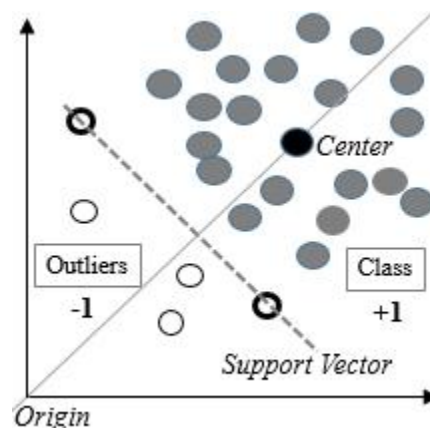


Fig. 12 One Class SVM

Gaussian Distribution

In this approach, all the features are modelled on a Gaussian Distribution and given a new data-point, the probability of the data-point is given by Gaussian/Normal Distribution Function. If the probability is below a particular threshold (which is set depending upon the performance of the model on the Validation Set), the new data-point is claimed to be an outlier or anomalous.

According to Gaussian/Normal Distribution:

$$p(x) = p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi} \sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

In case of building the model for data set 2:

- Assume data is normally distributed
- Use covariance.EllipticEnvelope from scikit-learn to find key params of general distribution by assuming entire dataset = an expression of an underlying multivariate Gaussian distribution

Analysis of Model 2

Based on this study, we have observed that Algorithms - K-Means, 3D Cluster, PCA, Gaussian Distribution, and Isolation Forest have detected the high prices.

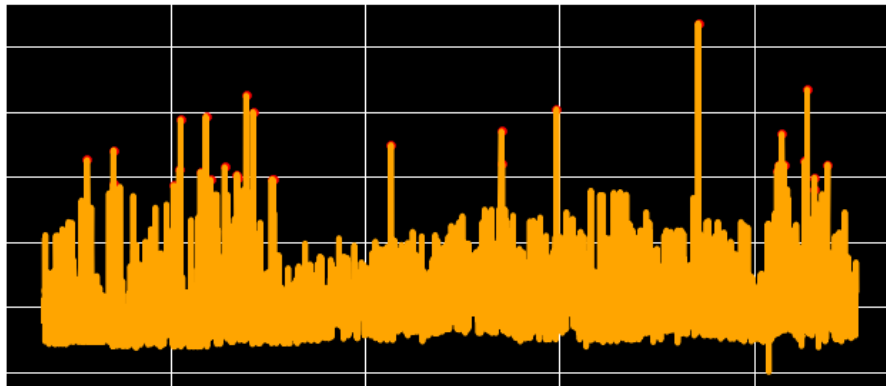


Fig. 13 Observation of algorithms (K-Means, 3D Cluster, PCA, Gaussian Distribution, and Isolation Forest) - Code Screenshot

But only One Class SVM has detected both high prices as well as low prices.

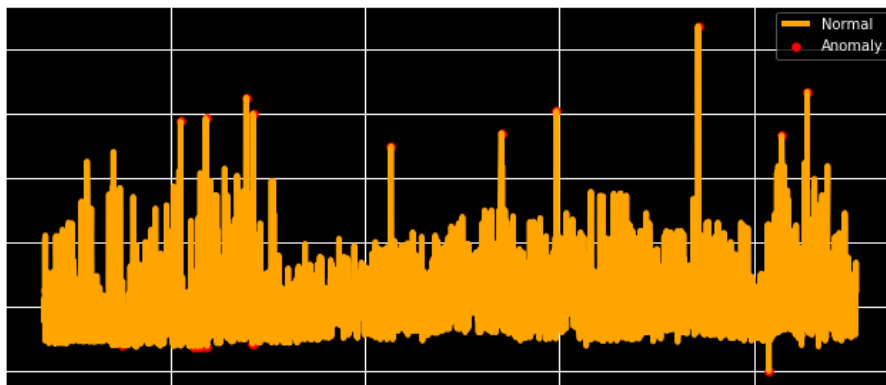


Fig. 14 Observation of One-Class SVM algorithm – Code Screenshot

Conclusion Discussion

Problem Statement 1

I built the autoencoders model to check for anomalies in the ECG Data set. The anomalies here reflected the irregularities or the abnormal function. The accuracy of my model is **94.96%**.

Problem Statement 2

I used different algorithms to check the prices of Hotel Rooms with the given [Expedia Hotel Search Data Set](#). I observed that Algorithms - K-Means, 3D Cluster, PCA, Gaussian Distribution, and Isolation Forest have detected the high prices. But only One Class SVM has detected both high prices as well as low prices, and thus it was the most accurate algorithm among all these.

My Contribution

This project focuses on solving three Problem Statements.

Problem Statement 1: ECG

Problem Statement 2: Expedia Hotel Search

Problem Statement 3: Sensors

Out of these 3, I have worked on Problem Statements 1 & 2.

My teammate worked on the 3rd Problem Statement.

Code

Problem Statement 1: ECG

[Code/Notebook](#)

Problem Statement 2: Expedia Hotel Search

[Code/Notebook](#)

Problem Statement 3: Sensors

[Code/Notebook](#)

Data Sets

Problem Statement 1: ECG

[Kaggle Link](#)

Problem Statement 2: Expedia Hotel Search

[Kaggle Link](#)

Problem Statement 3: Sensors

[Kaggle Link](#)

References

- [1] Li HZ, Boulanger P. A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG). *Sensors (Basel)*. 2020 Mar 6;20(5):1461. doi: 10.3390/s20051461. PMID: 32155930; PMCID: PMC7085598.
- [2] <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/anomaly-detection>
- [3] <https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/pca-based-anomaly-detection>
- [4] <https://towardsdatascience.com/5-anomaly-detection-algorithms-every-data-scientist-should-know-b36c3605ea16>