OUTLIERS DETECTION IN REGRESSION AND FUNCTIONAL REGRESSION

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

MASTER OF SCIENCE

in

Mathematics and Computing

by

Aastha Barnwal

(Roll No. 222123001)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

April 2024

CERTIFICATE

This is to certify that the work contained in this report entitled **"Outliers Detection in Regression and Functional regression"** submitted by **Aastha Barnwal** (**Roll No: 222123001**) to Department of Mathematics, Indian Institute of Technology Guwahati towards the requirement of the course **MA699 Project** has been carried out by her under my supervision.

Guwahati - 781 039
April 2024

(Dr. Arabin Kumar Dey)
Project Supervisor

ABSTRACT

Since Outliers have the potential to significantly change model estimates and predictions, outlier detection is a crucial stage in statistical analysis, particularly in regression and functional regression modeling. The goal of this research is to examine methods for finding outliers in various regression frameworks, such as decision tree-based techniques, conventional regression, and the application of these ideas to functional regression situations.

Our system blends classical statistical methods like residual analysis, Cook's distance, and leverage points detection with cutting edge machine learning techniques like isolation forests and robust regression. Within the functional regression framework, we study penalized functional regression models and functional depth-based outlier detection. Performance evaluation is conducted using real-world datasets from the environmental science, healthcare, and finance sectors as well as synthetic data simulations.

We extend outlier detection techniques into functional regression, allowing us to identify abnormal functional relationships that are frequently missed in conventional scalar regression setups.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr. Arabin Kumar Dey, for all of his help and assistance during the study process. His knowledge and counsel have been really helpful in guiding my research and realizing my ideas, which has allowed me to successfully finish my assignment.

Additionally, I would like to thank IIT Guwahati for providing the tools I needed to carry out my research successfully. In addition, I am incredibly grateful for my family's and friends' constant belief in me, which has propelled both my professional and personal development.

To sum up, I would want to express my sincere gratitude to everyone who helped make this initiative a success. Without your invaluable assistance, I could not have completed it.

Contents

List of Figures

List of Tables

Chapter 1

Statistical Test for Outliers

Data patterns that differ from typical occurrences in terms of data attributes are called anomalies. In many different application domains, the identification of anomalies is highly relevant and frequently yields crucial information that can be put into practice. For instance, irregularities in credit card transactions may indicate credit card fraud. An unusual area in an astronomical photograph can be a sign of a newly discovered star.

Outliers and influential points are sensitive to regression analysis.

1.1 Influential Points

A point is considered influential if its removal from the calculations significantly alters a parameter estimate.

1.2 Cooks distance & Leverage Points

In statistics, estimating influential points from models is a difficult problem. The objective is to determine which sample set elements are more pertinent to the statistical model. A common method from descriptive statistics to evaluate the impact of individual observations on the fitted model is **Cook's distance**.

The conventional Cook's distance predicts the second image based on the first one using a linear model, i.e., $Y = XW$, where $W \in \mathbb{R}^{d \times d}$, and $\tilde{X}$ is the augmented design matrix with a column of ones to account for the bias term, $\tilde{X} = [X, 1_n]$. The solution to this least squares problem is given by the Wiener-Hopf normal equations,

$$W = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T Y$$

. The predictions can be expressed as $\widehat{Y} = \tilde{X}W = \tilde{X}(\tilde{X}^T\tilde{X})^{-1}\tilde{X}Y = HY$, where H which is known as the projection matrix(Hat matrix).

Now, by taking the derivative of the predictions $\widehat{Y}$ with respect to Y,

$$H = \frac{d\widehat{Y}}{dY}$$

. The i-th element of the diagonal of H is thus given by $h_i = x_i^T(\tilde{X}^T\tilde{X})^{-1}x_i$. and is known as the **leverage** of the i-th observation. Similarly, the i-th element of the residual vector

$$e = y - \widehat{y} = (I - H)y$$

is denoted by e_i.

The **Cook's distance** D_i for observation x_i, $i = 1, 2, 3....n$, is the total of all the modifications made to the regression model upon observation i-th is removed from it:

$$D_i = \sum_{j=1}^{n} \frac{(y_j - \hat{y}_{j|i})^2}{d \times \text{MSE}^2}$$

where $\hat{y}_{j|i}$ is the fitted response value that results from removing i, and MSE is the mean-square error of the regression model.

Equivalently, it can be expressed using the leverage:

$$D_i = \frac{e_i^2 \times h_i}{d \times \text{MSE}^2 \times (1 - h_i^2)}$$

1.3 Plotting of leverage, influential and regression line

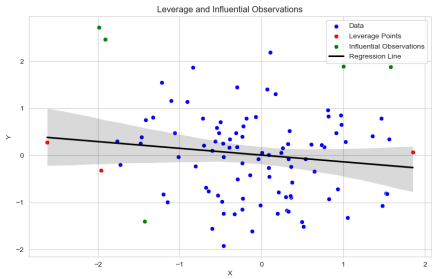


Figure 1.1: Leverage & influential observation

Remark 1.3.1. Although not all outliers are influential and not all influential points are outliers, outliers can be both influential and leverage points.

Chapter 2

Decision Tree

2.1 Problem Setting

1. **Set of possible instances (X):**
    - The examples that the decision tree will be trained on. A collection of pairs (x1, y1), (x2, y2),..., (xN, yN) that show instances and the target values that go with them
2. **Unknown target function (f):**
    - The method that associates values from X instances with matching outputs in Y.
3. **Input**
    - Typically represented by feature vectors.
    - Although they may handle continuous features as well, decision trees typically focus on discrete characteristics.

4. **Output Hypothesis $h \in H$ that best approximates target function f:**

    - The objective is to identify the decision tree hypothesis (h) that most closely approximates the unknown target function f among the set of hypotheses (H).
    - Can be either discrete (Classification decision tree) or continuous (Regression decision tree).

2.2 Why Decision Tree algorithm?

Because decision trees are good at effectively quantifying result values and probabilities, they are chosen as an algorithmic approach. They are useful for a variety of tasks, including regression and classification, and they make it possible to build prediction models that aid in decision-making. Training datasets are used in supervised learning to build these models. Decision tree visualizations are a popular data mining tool because they provide users with intuitive representations of decision processes, which aid in comprehension.

2.3 Types of Decision Tree

They are of two types:

- **Classification Tree:** These decision trees are used in classifying activities where the goal is to divide the content into different classes or classes.

- **Regression Tree:** Regression trees are used in regression tasks where the goal is to predict continuous values rather than discrete groups.

2.4 What is Decision Tree?

A supervised machine learning technique that can work with labeled data is called a decision tree. Applications using both regression and classification can benefit from it. It uses a hierarchical framework and attributes to classify data or forecast continuous values.

2.4.1 Structure of a Decision Tree

The following are the structure of decision tree:

- **Root node:** It is the topmost node of tree.At this, first split is performed.
- **Internal Node:**One of the decision trees that represents the middle selection based on a particular feature, giving rise to additional branches.
- **Leaf node:**The decision tree's terminal nodes are where the last categorization or forecast is produced.
- **Branches:**The decision tree's node connections show how decisions based on feature values are made. A response, such as yes or no, is represented by each branch.

2.4.2 How Decision Tree algorithm works?

- Select the feature (variable estimator) that best classifies the data set into the desired category and assign this feature to the root of the root.
- In order to maximize data classification, descend from the root node and make the right choices at each internal node.
- Iterate back to Step 1 until a class is assigned to the supplied data.

2.4.3 Build a Decision Tree using CART or ID3

There are many ways to build a decision tree. In this project, we will follow CART algorithm.

The algorithm follows the below work flow in order to build a Decision Tree:

- **Initialization:** Start with the entire dataset and define the target variable.
- **Feature Selection:** Choose the best feature to split the dataset.
- **Splitting:** Partition the data based on the selected feature's values.
- **Build the node:** Build a descendant of the node based on splitting
- **Recursion:** Repeat steps 2, 3 and 4 recursively until stopping criteria are met.
- **Leaf node:** If there is no more splitting possible, assign categorization labels to the leaf node.

2.5 Feature Selection

The decision tree algorithm begins by selecting the best attribute, sometimes referred to as the "best feature" or "predictor variable."

But what does this involve? We want to know how to find changes or features that best distribute information right now. This decision requires several factors:

1. Entropy
2. Information Gain
3. Gini Index
4. Weighted Average Variance(for Regression Problem)

2.5.1 Entropy

In physics, entropy is only a metric used to express how chaotic or unpredictable a system is. It can be viewed as a measure of disorder or uncertainty with regard to target prediction in the context of decision trees. It is employed in decision trees to determine how to separate data.

If a random variable X can take N different values x_i, the ith value x_i occurs with probability $p(x_i)$, we can associate the following entropy with X:
$H(x) = -\sum_{i=1}^{N} p(x_i) \log_2 p(x_i)$

As the uncertainty or degree of uncertainty in the value of X increases, entropy increases. When X is zero chaotic (i.e., there is no uncertainty), the entropy will be zero.

2.5.2 Information Gain

Information gain (IG) is a measure of the amount of information a given or different outcome has and is important information used in creating a decision tree. It is important to determine the difference between the optimal distribution of data at each decision tree node;

Equation for Information Gain (IG):

$$\text{InformationGain} = \text{Entropy(Parent)} - \text{Entropy(Children)}$$

In other words:

$$\text{IG}(T_a) = H(T) - H(T_a)$$

where:

$$H(T) = -\sum_{i=1}^{J} p(i) \log_2 p(i)$$

$$H(T_a) = \sum_{a} \sum_{i=1}^{J} p(i|a) \log_2 p(i|a)$$

Here, $p(i)$ is the probability of class i in the root node, and $p(i|a)$ is the probability of class i in the child given a.

2.5.3 Gini Index

If a data set E contains examples from n classes, the Gini index, $gini(E)$, is defined as:

$$gini(E) = \sum_{j=1}^{v} p_j(1 - p_j)$$

where p_j is the relative frequency of class j in E. The Gini index is a measure of the total variation in category v. If all p_j are close to 0 or 1, its value is small.

If the data set E is split on attribute A into two subsets E_1 and E_2, the Gini index $gini(E)$ is defined as:

$$giniA(E) = \frac{|E_1|}{|E|}gini(E_1) + \frac{|E_2|}{|E|}gini(E_2)$$

Reduction in Impurity:

$$\text{gini}(A) = \text{gini}(E) - \text{giniA}(E)$$

To split the node, the attribute with the highest reduction in impurity, or the smallest giniSplit(E), is selected. It is necessary to list every conceivable dividing point for every attribute.

2.5.4 Weighted Average Variance

For feature F,

$$VarF(E) = \sum_{j=1}^{v} \frac{|E_j|}{|E|}Var(E_j)$$

where $Var(E_j) = \sum_{j=1}^{v} \frac{E_j}{E}Var(E_j)\left(y - \frac{\sum_{j=1}^{v} y_j}{E_j}\right)^2$ and $y = \frac{\sum_{j=1}^{v} y_j}{E_j}$.

Pick the feature with the lowest weighted average variance.

2.6 Advantage & Disadvantage

Advantages

- Decision trees' high definition and ability to be expressed as a series of questions or other statements make them ideal for real-world business applications.
- Making forecasts takes relatively little time. It is merely the assessment of a specific set of circumstances for a certain data point.
- Imputation of missing values is not necessary because the algorithm adjusts itself.

Disadvantages

- Adding new elements causes the entire tree to be rebuilt, which means the nodes must be recomputed.
- High output variance from overfitting leads to mistakes in judgment calls and inaccurate outcomes.
- They are affected by noise, and even a very small noise can make the decision tree model unstable.

Chapter 3

Outlier Detection using emsemble Technique: Isolation Forest

In many different application domains, the identification of anomalies is highly relevant and frequently yields crucial information that can be put into practice.This chapter explains how to build a tree structure that isolates each and every instance. Anomalies are isolated nearer the tree's base due to their vulnerability to isolation, whereas normal points are isolated at the tree's base. We refer to this type of tree as an isolation tree because it is the foundation of our anomaly detection technique.

The suggested technique, dubbed **Isolation Forest** or iForest, creates an ensemble of iTrees for a particular data collection; anomalies are those occurrences on the iTrees with short average path lengths. The number of trees to be built and the size of the subsampling are the only two variables

in this method.

3.1 Isolation Tree

Let T be a node of an isolation tree. T is either an interior node with one test and precisely two daughter nodes, or an external node without children. (T_l, T_r). A test consists of an feature q and a split value p such that the test $q < p$ divides data points into T_l and T_r.

Build an Isolation Tree

Given a sample of data $X = \{x_1, ..., x_n\}$ of n instances from a d-variate distribution, to build an isolation tree (iTree).

- Randomly select the feature and splitting values.
- Partition the data based on the selected features's value.
- Build a descendant of the node based on splitting
- Repeat 2,3,4 recursively until stopping criteria are met.
- **Stopping Criteria:**(i) the tree reaches a height limit, (ii) $|X| = 1$ or (iii) all data in X have the same values.

Creating a ranking that represents the degree of oddity is the job of anomaly detection.Sorting data points based on their path lengths or anomaly scores is therefore one technique to find anomalies; anomalies are points that are rated highest on the list. The following is how we define path length and anomaly score:

3.2 Anomaly Score

Path Length

The number of edges x traverses in an iTree from the root node until the traversal is ended at an external node determines the route length $h(x)$ of a point x.

Anomaly Score

The anomaly score s of an instance x is defined as:

$$s(x, n) = 2^{-}\frac{E(h(x))}{c(n)}$$

where

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

, $H(i)$ is the harmonic number,$E(h(x))$ is the average of $h(x)$ from a collection of isolation trees. In Equation:

- When $E(h(x)) \to c(n)$, $s \to 0.5$.
- When $E(h(x)) \to 0$, $s \to 1$.
- When $E(h(x)) \to n-1$, $s \to 0$.

s is monotonic with respect to $h(x)$.

Finding Anomalies

Using the anomaly score s, we are able to make the following assessment:

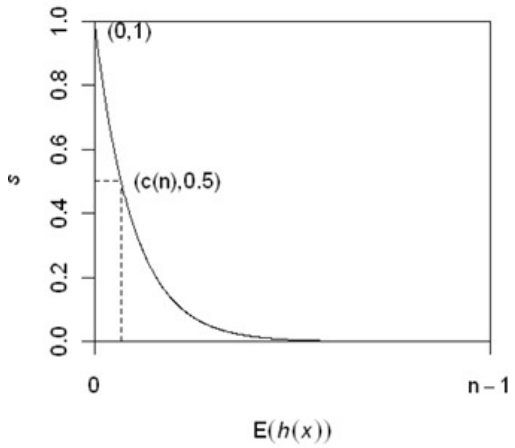


Figure 3.1: Expectation of h(x) Vs anomaly score

- (a) If examples return s very close to 1, then they are definitely anomalies.
- (b) If examples have s much smaller than 0.5, then it is acceptable to consider them to be typical occurrences.
- (c) If all the examples return $s \approx 0.5$, then the entire sample does not really have any distinct anomalies.

3.3 Isolation Forest

Using iForest, anomaly detection is a two-step procedure. Using training set subsamples, the first (training) stage constructs isolation trees. In order to determine an anomaly score for each test instance, the second (testing) stage puts them through isolation trees.

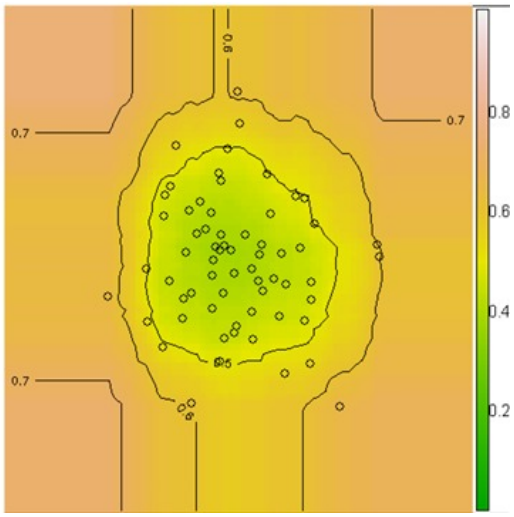


Figure 3.2: Isolation of data points

How Algorithm works:

1. Set the parameters n_{tree} and subspace (where n_{tree} represents the number of isolation trees to be constructed and subspace is the number of samples taken by each isolation tree).

2. Generate n_{tree} bootstrap samples with replacement from the dataset.

3. Select predictor variables randomly and at every node to determine the split for building the isolation tree.

4. Obtain ensemble anomaly score by taking average of the score from individual trees.

3.3.1 Performance Measures

The learning algorithm's performance on the anomaly class has been evaluated using metrics including sensitivity, specificity, prevalence, detection

rate, and positive prediction value.

Predicted	**Actual**	
	Non-anomalies	**Anomalies**
Non-anomalies	True Negative (TN)	False Negative (FN)
Anomalies	False Positive (FP)	True Positive (TP)

Table 3.1: Confusion Matrix

The number of accurately anticipated non-anomalies is known as True Negative (TN). A True Positive (TP) is the quantity of anomalies that are accurately predicted. The quantity of real anomalies that are assumed to be non-anomalies is known as the False Negative (FN). The quantity of non-anomalies that are forecast as anomalies is known as false positives, or FP. The performance measures taken into account in this study can be computed with the following formulas:

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Prevalence} = \frac{TP + FN}{TP + FN + TN + FP}$$

$$\text{Detection Rate} = \frac{TP}{TP + FN + TN + FP}$$

$$\text{Positive Prediction values} = \frac{TP}{TP + FP}$$

3.4 Isolation Forest on Synthetic Dataset

Train Model:

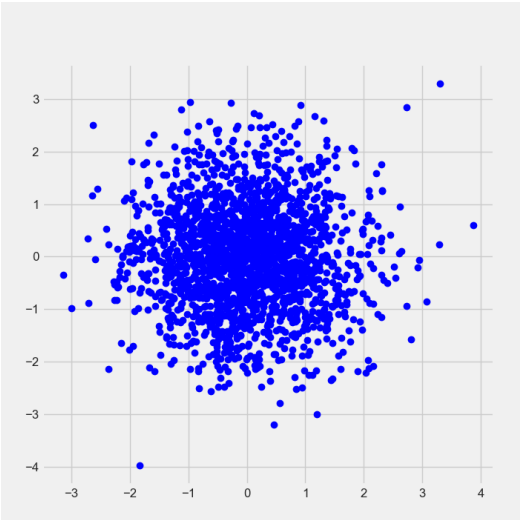


Figure 3.3: Synthetic data

Isolationforest(ntrees=20, maxdepth=None, subspace=256)

Number of Estimator : 20

Subspace : 256

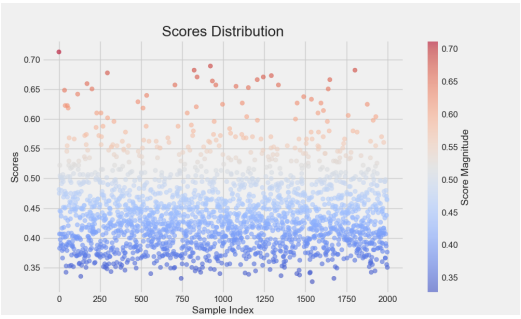


Figure 3.4: Decision Score of data points

Confusion Matrix for performance of model

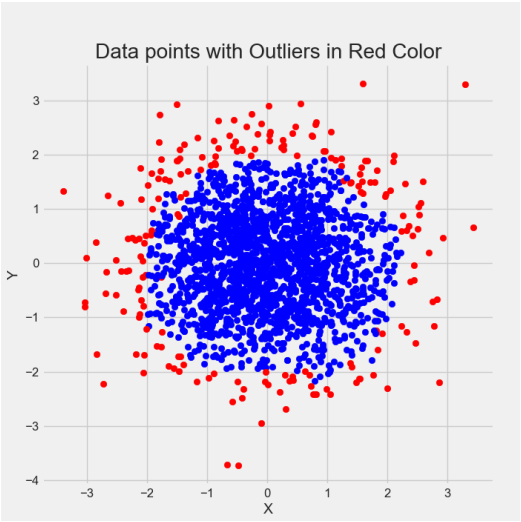


Figure 3.5: Outliers and Inliers using Isolation Forest

	Predicted Class	
True Class	Anomalies	Non-anomalies
Anomalies	1	0
Non-anomalies	202	1797

Sensitivity: 0.8989494747373686

Specificity: 1.0

Positive Predictive Value (PPV): 1.0

Detection Rate (DR): 0.8989494747373686

Chapter 4

Random Forest

Random Forest is a powerful learning algorithm used for classification and regression in machine learning. It is based on the concept of decision trees, which combine predictions from multiple trees to increase accuracy and reduce overfitting.

4.1 Why Random Forest?

Random Forest is a widely used option in machine learning because of its durability and versatility. It performs exceptionally well at handling noisy data, nonlinear relationships, and complex datasets with a high dimensionality. Random Forest reduces overfitting and yields more accurate and dependable predictions than single decision trees by integrating the predictions of numerous decision trees trained on various subsets of the data. Its suitability for a variety of classification and regression problems is further increased by its capacity to handle missing values and estimate feature importance.

4.2 How Random Forest Algorithm works?

The Random Forest algorithm operates through the following steps:

1. Set the parameters n_{tree} and m_{try} (where n_{tree} represents the number of decision trees to be constructed and m_{try} is the number of variables considered at each split).

2. Generate n_{tree} bootstrap samples with replacement from the dataset.

3. Select m_{try} randomly estimate the variable and evaluate each variable to determine the best split to create a decision tree.

4. The overall estimate is obtained by averaging the predictions of individual trees. This randomness in the selection process ensures that the trees are uncorrelated, thus reducing the number of variations in the integrated estimate.

The main way that Random Forests differ from conventional tree-based ensemble techniques, such bootstrap aggregation, is in the way that predictors are chosen at random.

4.3 Advantages & Disadvantages

Advantages:

- Generally speaking, Random Forest generates quite accurate models, especially when compared to individual decision trees.

- Provides a measure of importance, aids feature selection and data understanding
- Its efficiency for huge datasets and parallel computing settings stems from its easy parallelization of individual tree training.

Disadvantages:

- Random Forest can be memory- and computationally-intensive, particularly for large datasets with a high number of trees and attributes.
- Random forests may encounter inconsistent data where one class dominates others, leading to bias.
- Scores for feature relevance may be skewed in favor of continuous variables with large variance or categorical variables with several levels.

4.4 Interpretation of Performance Metrics

To determine which model is best for a situation, it is necessary to compare and evaluate the effectiveness of each model. Testing helps choose the best models and best measures. Complex mathematical evaluation, sensitivity, and performance measurements such as receiver operating characteristic (ROC) curves and likelihood ratio are quantitative methods. In competitive classification, error rate is often used to evaluate classification performance.

4.4.1 Confusion Matrix

An N x N matrix, where N is the number of target classes, is called a confusion matrix when it comes to assessing the effectiveness of a classification model.

The machine learning model's projected values are compared with the actual target values in the matrix.

A classification model's performance can be assessed using a matrix by computing performance metrics such as accuracy, precision, recall, and F1-score.

Suppose we have two classes, then confusion matrix is

Actual	**Predicted**	
	YES	**NO**
YES	True Positive (TP)	False Negative (FN)
NO	False Positive (FP)	True Negative (TN)

Table 4.1: Confusion Matrix

The total of the four entries $TP + TN + FP + FN = n$, the number of test data.

Accuracy:

Simply put, accuracy quantifies how frequently the classifier predicts the future correctly. It is the ratio of the total number of predictions to the number of accurate predictions.

Put simply, it indicates the fraction of all positive forecasts that are actually realized.

When evaluating classification issues that are well-balanced, non-skewed, and free of class imbalance, accuracy is a suitable option.

$$\text{ACCURACY} = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision:

The ratio of the total number of successfully classified positive classes to the total number of anticipated positive classes is known as precision. or the extent to which we accurately forecasted, out of all the predictive positive groups. High precision is desired (preferably 1).

$$\text{PRECISION} = \frac{TP}{FN + FP}$$

Recall:

The ratio of the entire number of positively classed classes that were correctly classified divided by the total number of positive classes is known as recall.

"When False Negative outweighs False Positive, recall is a useful metric."

$$\text{RECALL} = \frac{TP}{TP + FN}$$

F-measure / F1-Score:

The harmonic mean of recall and precision is represented by the F1 score, which is a value between 0 and 1. The F1 score helps your classifier maintain a balance between recall and precision. Your F1 score will be low if your recall is poor as well as low if your precision is poor.

$$\text{F1-score} = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Specificity & Sensitivity:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Sensitivity} = \frac{TP}{TP + FP}$$

4.5 Random Forest on Breast Cancer Dataset

Description of the dataset are given below this section.

Target variable represents the binary classification of tumors

We have trained the set with 0.8 ratio.

Train Model:

RandomForest(data, $ntrees = 20$)

Type of random forest : Classification

No.of variables tried at each split: 4

Number of trees: 20

OOB estimate of error rate: 37.57 %

Confusion matrix:

	Predicted Class		
True Class	1	2	Class Error
1	40	3	0.244
2	1	70	0.159

Accuracy (using test data): 0.96

Area under the curve(calculated from test set):0.99

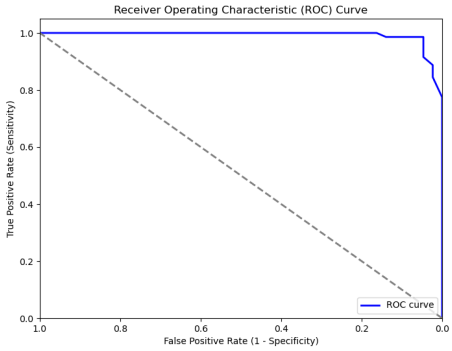


Figure 4.1: Receiver Operating Characteristic (ROC) Curve

4.6 Description of Dataset

A collection of data with details about breast cancer tumors is called the breast cancer dataset. It includes a label identifying whether each tumor is benign (not cancerous) or malignant (cancerous), along with other qualities or characteristics of the tumors.

Features

- Features from the dataset characterize the properties of cell nuclei discovered during breast carcinoma biopsies.
- These features include attributes such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.
- Quantitative measurements of the dimensions, form, and texture of the cell nuclei are provided by each characteristic.

Label

- Every tumor's diagnosis is also listed on a label included with the dataset.
- Tumors labeled as "M" are malignant, indicating the presence of cancerous cells.
- Tumors labeled as "B" are benign, indicating the absence of cancerous cells.

Objective

- Developing machine learning models that can reliably identify breast cancers as benign or malignant based on their features is the main goal of employing this dataset.

Source

A popular Python machine learning toolkit called Scikit-learn comes with a number of built-in datasets for practice and experimentation. Among these pre-built datasets is the breast cancer dataset.

4.7 Outlier Detection using Random Forest:

4.7.1 Proposed Agorithm:

I have implemented here two algorithm for detecting outliers using random forest.

4.7.2 Algorithm 1:

Decision Tree Agorithm:

- Select best feature and best splitting values
- Partition the data based on the selected feature.
- Build descendant of the node based on splitting
- Repeat these step untill stopping criteria are met.
- **Stopping Criteria:** (i) $|X| = 1$ or (ii) all data in X are homogeneous.

Random Forest Algorithm

1. Set the parameters ntree and subspace (where n_tree represents the number of decision trees to be constructed and subspace is the number of samples taken by each isolation tree).
2. Generate n_tree bootstrap samples with replacement from the dataset
3. Select best predictor variable for every node to determine the best split for decision tree.
4. Obtain ensemble anomaly score by taking average of the score from individual tree(same approach used in isolation forest)

4.7.3 Algorithm 2: (Hybrid Method)

Decision Tree Algorithm:

- Select best feature and best splitting values

- Partition the data based on the selected feature.
- Build descendant of the node based on splitting
- Repeat these step untill stopping criteria are met.
- **Stopping Criteria:** (i) $|X| = 1$ or (ii) **if all data in X are homogeneous then randomly isolates each data point till we reach** $|X| = 1$

Random Forest Algorithm

1. Set the parameters ntree and subspace (where n_tree represents the number of decision trees to be constructed and subspace is the number of samples taken by each isolation tree).
2. Generate n_tree bootstrap samples with replacement from the dataset
3. Select best predictor variable for every node to determine the best split for decision tree.
4. Obtain ensemble anomaly score by averaging the score from individual tree(same approach used in isolation forest)

4.8 Outlier detection using Random Forest on Synthetic data

Algoritm 1:

Train Model:

RandomForest(ntree=20,subspace:256)

Number of tree: 20

Subspace: 256

Confusion Matrix for performance of model:

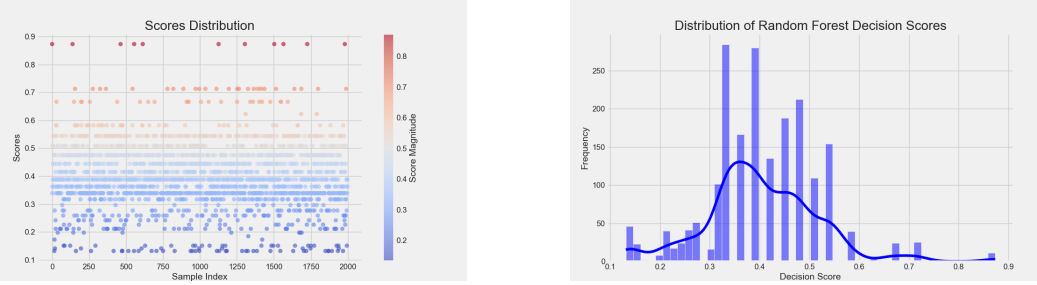
	Predicted Class	
True Class	Anomalies	Non-anomalies
Anomalies	1	0
Non-anomalies	35	1964

Sensitivity: 0.9824912456228114

Specificity: 1.0

Positive Predictive Value (PPV): 1.0

Detection Rate (DR): 0.9824912456228114



(a) Anomaly Score (b) Anomaly Score

Figure 4.2: Decision Score in different plot

Algoritm 2:

Train Model:

RandomForest(ntree=20,subspace:256)

Number of tree: 20

Subspace: 256

Confusion Matrix for performance of model:

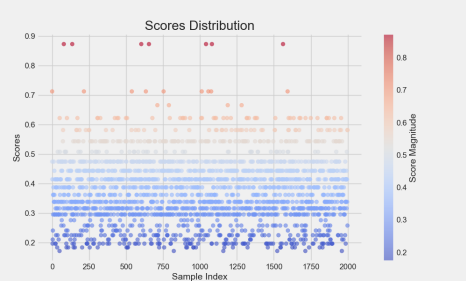
	Predicted Class	
True Class	Anomalies	Non-anomalies
Anomalies	1	0
Non-anomalies	19	1980

Sensitivity: 0.9904952476238119

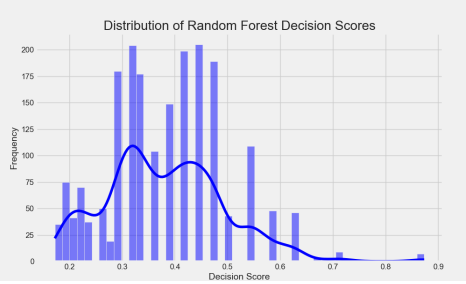
Specificity: 1.0

Positive Predictive Value (PPV): 1.0

Detection Rate (DR): 0.9904952476238119

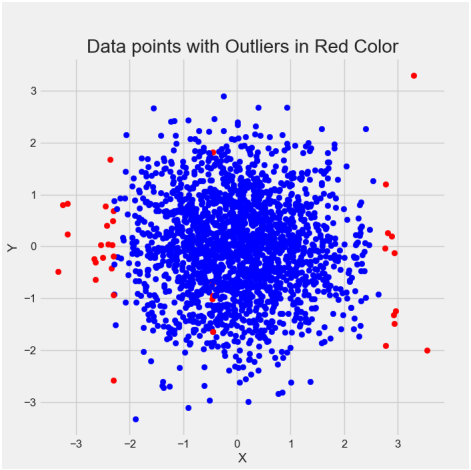


(a) Anomaly Score

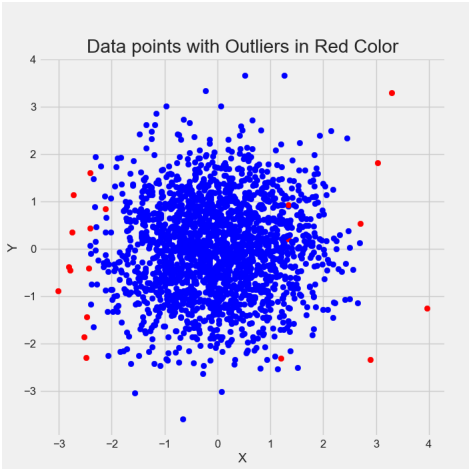


(b) Anomaly Score

Figure 4.3: Decision Score in different plot



(a) Agorithm 1



(b) Agorithm 2

Figure 4.4: Outliers for both Algorithm

Chapter 5

Survival Analysis

5.1 Introduction

The main area of interest in many clinical trials is the time until a certain event occurs. A few examples are:

1. Time it takes for a patient to die.
2. Time it takes until an infection occurs.
3. Time from response until a disease returns.

Clinical trials usually have a limited time frame for data collection, which means that some research participants may not have any time to event recorded at all, even though they may have experienced it later. When an individual (subject) in the study declines to participate, the analysis becomes more complex. These kinds of observations are referred to as right-censored observations.

The term "survival time" refers to the period of time starting at a predetermined moment and ending when the event occurs. Survival analysis is the process of examining data that includes information about the group.

5.2 Notations and Basic Concepts

To begin our examination of survival analysis, let's review some terminology and annotations that will help us understand the rest of the report.

Let T be the random variable denoting the time to event. We can see that T is greater than or equal to zero.

The distribution of the random variable T can be expressed as:

$$F(t) = P[T \geq t], \quad t \geq 0$$

where $F(t)$ is the probability that a randomly selected subject will die before time t.

If T is a continuous random variable, then its density function $f(t)$ can be expressed as follows:

$$f(t) = \frac{dF(t)}{dt}, F(t) = \int_0^t f(u)\, du$$

The survival function $S(t)$ is defined as:

$$S(t) = P[T > t] = 1 - F(t)$$

When T is the survival time, $S(t)$ is the probability that a randomly selected subject will survive to time t and beyond.

If T is a continuous random variable, we have

$$S(t) = \int_0^\infty f(u)\, du$$

$$f(t) = -\frac{dS(t)}{dt}$$

The survival function S(t) is a non-increasing function over time taking the value 1 at $t = 0$, i.e., $S(0) = 1$.

5.2.1 Hazard Function

The hazard rate is the instantaneous rate of experiencing an event at time t given that the event has not occurred until time t. Specifically, the hazard rate $\lambda(t)$ is defined by the following equation:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P[t \leq T \leq t + \Delta t | T \geq t]}{\Delta t}$$

where $\Delta(t)$ is very small.

$$P[t \leq T \leq t + \Delta t | T \geq t] = \lambda(t)\Delta t$$

This implies,

$$\lambda(t) = \frac{\frac{P[t \leq T \leq t+\Delta t]}{\Delta t}}{P[T \geq t]} = \frac{f(t)}{S(t)} = -\frac{S'(t)}{S(t)} = -\frac{d\log(S(t))}{dt}$$

On integrating both sides,

$$\Lambda(t) = \int_0^t \lambda(u)d(u) = -\log(S(t))$$

where $\Lambda(t)$ is referred to as the cumulative hazard function. Here we have used the fact that $S(0) = 1$.

Hence,

$$S(t) = e^{-\Lambda(t)} = e^{\int_0^t -\lambda(u)du}$$

. Our main aim is to estimate this survival function.

5.2.2 Kaplan-Meier Estimate

The likelihood of a random individual surviving in a given amount of time when at most one distinct "time to event" happens within an interval is represented by the Kaplan-Meier survival curve.

Kaplan Meier estimate is based on the following three assumptions:

1. For participants who are admitted early or late in our trial, the survival probability are the same.

2. The event takes place at the specified time for every entry.

3. The chances of survival for censored subjects are same to those of study participants.

Let $d(x)$ denote the count of events that happen at time point x. According to the definition of Kaplan-Meier, $d(x)$ would be at most one, but there can be a possibility of subjects having the same time to event, and in that case, $d(x)$ can be greater than or equal to one.

Let $n(x)$ denote the count of subjects that are at risk just prior to time point x, which is equal to the number of subjects in our current sample who have neither been censored nor died prior to time x.

Then, Kaplan-Meier estimate can be expressed as:

$$\hat{S}(t) = \prod_{x \leq t} \left(1 - \frac{d(x)}{n(x)}\right)$$

Proof:

Consider the time interval $[t_i, t_{i+1}]$. Let us assume the following:

- n be the number of people at risk at time t_i,
- d be the number of deaths that occur in the specified time interval,
- p be the probability of death of a subject.

As can be seen, d is a binomial random variable with parameters n and p as defined above. We want an estimate of the parameter p. From the derivation of Maximum Likelihood estimate of binomial random variable, it can be seen that

$$\hat{p} = \frac{d}{n}$$

The probability of survival at the given time interval would then be $1-p$. To calculate survival at time t, we take the cumulative product of survival at each of the time intervals that are less than t to get the result.

5.3 Nelson-Aalen Estimation:

In the event of missing or censored data, the Nelson-Aalen estimator provides a non-parametric estimate of the cumulative hazard rate function.Given is

the estimate by:

$$\tilde{H}(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}$$

with d_i is the number of events at t_i and n_i is the total individuals at risk t_i.The Nelson-Aalen estimator's curvature provides insight into the shape of the hazard rate.

Chapter 6

Random Survival Forest

Now, let's talk about Random Survival Forest. Random survival forests are the common method used to study survival outcomes using random forests. In clinical trials, **random survival forests (RSF)** are an effective tool for risk prediction of right-censored outcomes. An ensemble of survival trees is created using Random Survival Forest using the log-rank splitting criterion.

6.1 Random Survival Forests algorithm

Following is the detailed algorithm of Random Survival Forest:

1. Draw B bootstrap samples from the original data.

2. Grow a survival tree for each bootstrap sample. At each node of the tree, randomly select p candidate variables. The node is split using the candidate variable that maximizes survival difference between daughter nodes.

3. Grow the tree to full size under the constraint that a terminal node should have no less than $d_0 > 0$ unique deaths.

4. Calculate a Cumulative Hazard Function (CHF) for each tree. Average to obtain the ensemble CHF.

6.2 Ensemble cumulative hazard

6.2.1 Terminal node prediction

- For a new observation, drop the observation down to the nal nodes of a tree and do this for all n_tree trees. Predictions can be summarized over all trees in several ways. The most common strategy is to apply the Nelson-Aalen estimator in each terminal node and to average the resulting estimates of the cumulative hazard function over all trees and time points to give a one-dimensional score.

- Let $(T_{1,h}, \delta_{1,h}), \ldots, (T_{n(h),h}, \delta_{n(h),h})$ be the survival times and the $0-1$ censoring information for individuals (cases) in a terminal node $h \in T$. An individual i is said to be right-censored at time $T_{i,h}$ if $\delta_{i,h} = 0$; otherwise, if $\delta_{i,h} = 1$, the individual is said to have died (experienced an event) at $T_{i,h}$. Let $t_{1,h} < t_{2,h} < \ldots < t_{N(h),h}$ be the $N(h)$ distinct event times. Define $d_{l,h}$ and $Y_{l,h}$ to be the number of deaths and individuals at risk at time $t_{l,h}$. The CHF estimate for h is the Nelson-Aalen estimator

$$\hat{H}_h(t) = \sum_{t_{l,h} \leq t} \frac{d_{l,h}}{Y_{l,h}}$$

- Let $H(t|x_i)$ be the CHF for ith individual. To determine this value,drop covariate vector x_i down the tree which will eventually fall into a unique terminal node h of the tree. The CHF for i is given as follows:

$$H(t|x_i) = \begin{cases} \hat{H}_h(t) & \text{if } x_i \in h \end{cases}$$

- The CHF given above is derived from a single tree. To compute an ensemble CHF, average over B survival trees is taken as follows:

$$\hat{H}_e(t|x_i) = \frac{1}{B}\sum_{b=1}^{B} \hat{H}_b(t|x_i)$$

6.3 Prediction of Survival Function

- RSF algorithm aggregates the Nelson-Aalen terminal node estimators of all bootstrap based trees to obtain the predicted survival functions $\hat{S}_r(t|x_i) = exp(-\hat{H}_e(t|x_i))$

- Now we go by calculating weighted Kaplan-Meier estimates at each terminal node.As for RSF algorithms ,the final prediction for the conditional survival function $S_0(t|x_i) = P(T > t|x_i)$,the predicted survival function $S^c(t|x_i)$ can be expressed as:

$$S^c(t|x_i) = \prod_{t_{l,h} \leq t} 1 - \frac{\sum_{j=1}^{n} d_{l,h}^j}{\sum_{j=1}^{n} Y_{l,h}^j}$$

where n is number of trees.

6.4 Prediction performance

6.4.1 Concordance Index

Prediction performance was calculated using the C-index. The following steps:

1. Form all possible pairs of cases over the data.

2. Omit those pairs whose shorter survival time is censored. Omit pairs i and j if $T_i = T_j$ unless at least one is a death. Let Permissible denote the total number of permissible pairs.

3. For any valid pair in which $T_i = T_j$, assign a value of 1 if the expected outcome for the shorter survival time is worse, and 0.5 if the predicted outcomes are equal. If the projected outcomes are tied for every allowed pair where $T_i = T_j$ and both are deaths, count 1; if not, count 0.5. If the death has a worse expected outcome than the other, count 1 for each acceptable pair where $T_i = T_j$, but not both, are deaths; if not, count 0.5. Let Concordance represent the total of all allowed pairings.

4. The C-index, C, is defined by C = Concordance/Permissible.

6.5 Random Survival Forest on "German Breast Cancer Study Group 2"

The target variable consists of two components:

(i) Event Indicator (cens) (ii) Time of Event (time)

We have trained our model with 0.75 ratio

Train Model:

RandomSurvivalForest(nestimators=1000)

Number of survival trees : 1000

Score : 0.675

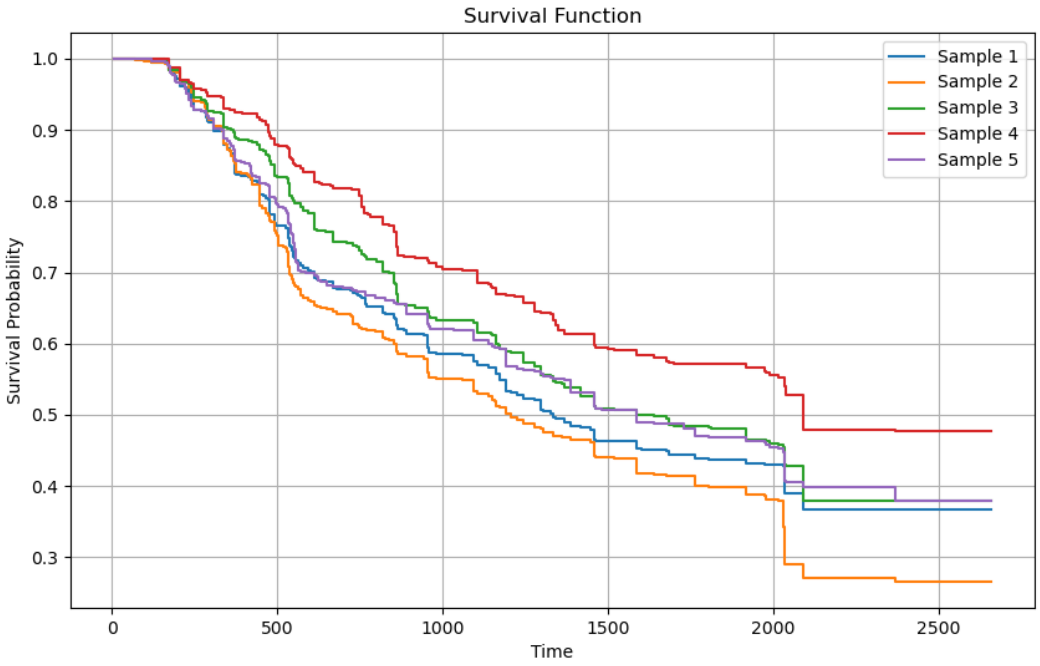


Figure 6.1: Survival Function of first 5 samples

6.6 Description of Dataset

The German Breast Cancer investigation Group 2 (GBSG2) dataset collects information about patients with breast cancer who are taking part in a clin-

ical trial. It covers an array of characteristics and outcomes of individuals diagnosed with breast cancer. Researchers utilize this information to look into the factors that influence the prognosis of patients with breast cancer. By analyzing the relationship between patient characteristics and survival rates, researchers aim to identify prognostic factors that could help predict how the disease will progress and guide treatment decisions.

Format:

age: Patient's age in years.

estrec: Estrogen receptor status.

horTh: Hormone therapy status.

menostat: Menopausal status.

pnodes: Number of positive axillary nodes.

progrec: Progesterone receptor status.

tgrade: Tumor grade.

tsize: Tumor size.

Source:

It is is available in the sksurv.datasets module of scikit-survival library.

6.7 Outlier Detection in Functional Regression: Random Survival Function

6.7.1 Proposed Agorithm:

I have implemented here two algorithm for detecting outliers using random forest.

6.7.2 Algorithm 1:

Survival Tree Agorithm:

- Select best feature and best splitting values
- Partition the data based on the selected feature.
- Build descendant of the node based on splitting
- Repeat these step untill stopping criteria are met.
- **Stopping Criteria:** (i) $|X| = 1$ or (ii) all data in X are homogeneous.

Random Survival Forest Algorithm

1. Set the parameters ntree and subspace (where n_tree represents the number of decision trees to be constructed and subspace is the number of samples taken by each isolation tree).
2. Generate n_tree bootstrap samples with replacement from the dataset. At each node of the tree, randomly select p candidate variables.
3. Select best predictor variable for every node to determine the best split for decision tree.
4. Obtain ensemble anomaly score by averaging the score from individual tree(same approach used in isolation forest)

6.7.3 Algorithm 2: (Hybrid Method)

Survival Tree Algorithm:

- Select best feature and best splitting values
- Partition the data based on the selected feature.
- Build descendant of the node based on splitting
- Repeat these step untill stopping criteria are met.
- **Stopping Criteria:** (i) $|X| = 1$ or (ii) **if all data in X are homogeneous then randomly isolates each data point if possible till we reach** $|X| = 1$

Random Survival Forest Algorithm

1. Set the parameters ntree and subspace (where n_tree represents the number of decision trees to be constructed and subspace is the number of samples taken by each isolation tree).
2. Generate n_tree bootstrap samples with replacement from the dataset. At each node of the tree, randomly select p candidate variables.-
3. Select best predictor variable for every node to determine the best split for decision tree.
4. Obtain ensemble anomaly score by averaging the score from individual tree(same approach used in isolation forest)

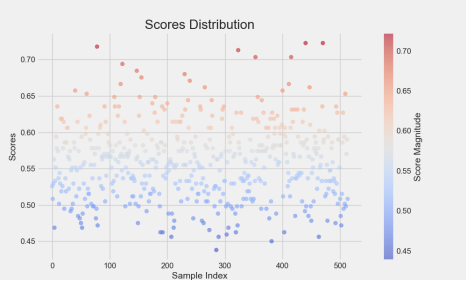
6.7.4 Outliers detection on "German Breast Cancer Study Group 2"

Algorithm 1:

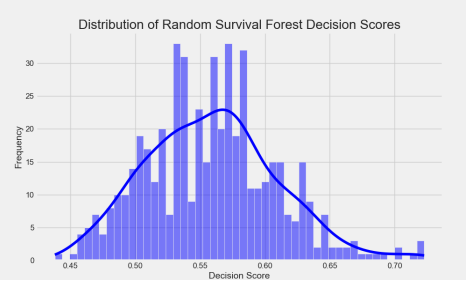
RandomSurvivalForest(ntree=20,subspace:256)

Number of tree: 20

Subspace: 256

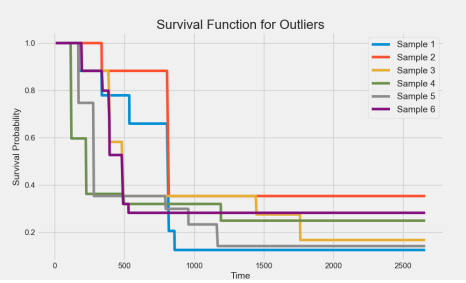


(a) Anomaly Score

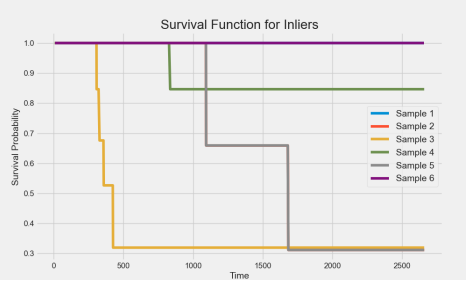


(b) Anomaly Score

Figure 6.2: Decision Score in different plot



(a) Survival Function for Outliers



(b) Survival Function for Inliers

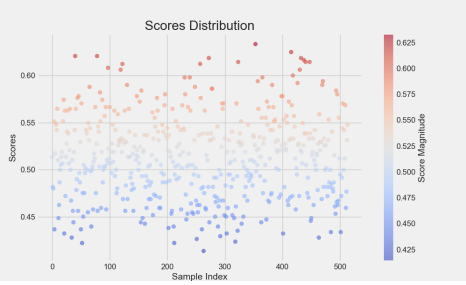
Figure 6.3: Survival Function

Algorithm 2:

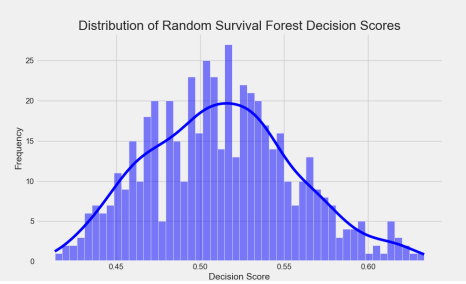
RandomSurvivalForest(ntree=20,subspace:256)

Number of tree: 20

Subspace: 256

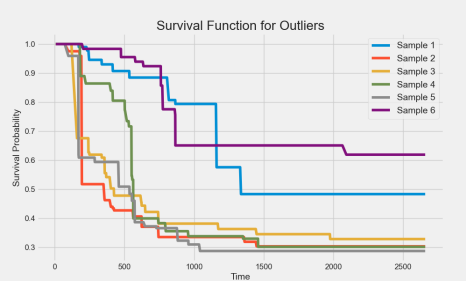


(a) Anomaly Score

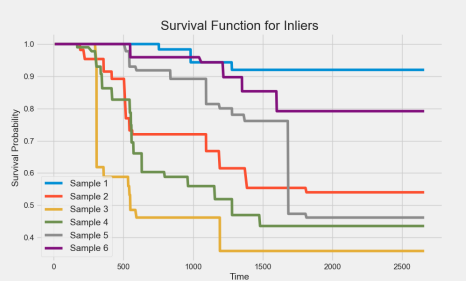


(b) Anomaly Score

Figure 6.4: Decision Score in different plot



(a) Survival Function for Outliers



(b) Survival Function for Inliers

Figure 6.5: Survival Function

Conclusion

Outliers in Regression

- Time complexity of outlier detection by random forest is quite high as compared to isolation forest
- By applying methods such as Random Forests or Isolation Forests, we were able to pinpoint the data points that markedly vary from the distribution or overall pattern of the data.
- We identified the observations that showed abnormally big or minor errors in comparison to the bulk of the data by looking at the residuals or prediction errors of the regression model.
- Regression outlier identification helped us find possible anomalies or significant points that might have an outsized effect on the model's performance or the accuracy of its predictions.

Outlier in Functional Regression

- Analyzing the behavior of curves or functions as opposed to individual data points was required for extending outlier detection to functional regression.
- The functional relationship between predictors and response variables was examined for anomalous trends or patterns using methods like Functional Data Analysis (FDA) and Functional Random Forests.

- We were able to identify anomalous functional forms, or situations in which the relationship between variables considerably differed from the predicted behavior, by identifying outliers in functional regression.

Future Work

- To maintain the efficacy of outlier identification strategies in regression and functional regression over a range of datasets and modeling scenarios, further refinement and validation of these techniques are required.
- Furthermore, investigating more complex outlier identification techniques and applying domain-specific expertise may improve the resilience and usefulness of outlier detection in regression tasks.

Bibliography

[1] M. K. Goel, P. Khanna, and J. Kishore. Understanding survival analysis: Kaplan-meier estimate. *International Journal of Ayurveda Research*, 2011.

[2] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2:841–860, 2008.

[3] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts; London, England, 2022. Printed and bound in the United States of America. Includes bibliographical references and index. Series: Adaptive Computation and Machine Learning Series.

[4] José A. Padrón Hidalgo, Adrián Pérez-Suay, Fatih Nar, and Gustau Camps-Valls. Nonlinear Cook Distance for Anomalous Change Detection. *Journal Name or Conference Name*, 2022.