“Evaluation of the Big Data Applications, Review Machine Learning Algorithms & Implementation of ML Algorithms for Classification of the Vehicles based on the data about Silhouettes.”

7141CEM- Big Data for Vehicles

Coursework-Individual Report

Prathmesh A. Gardi

Student Id: 13355397

Email: gardip@uni.coventry.ac.uk

Introduction

What is Big Data?
Big Data is basically a massive volume of structured and unstructured data that is generated, processed, and analyzed at an unprecedented scale. \
This data is characterized by its size, complexity, and the speed at which it is produced.
Big Data encompasses a wide variety of data types, including text, images, videos, social media interactions, sensor data, transaction records, and more.

The concept of Big Data gained prominence in the early 21st century as organizations began to encounter challenges in managing and extracting value from the massive amounts of data being generated.
Traditional data processing tools and relational databases were inadequate for handling such scale and diversity.
This led to the development of new technologies and methodologies to store, process, and analyze Big Data.

Big Data and Big data Analytics have tremendous opportunities and applications in various fields. In this report,
In the first section,
We will take a look at various applications of big data and the challenges and opportunities regarding that.
In the next section of the report, we will explore different machine learning algorithms for classification and Clustering. We will look at the overview of the algorithm and its applications.

In the third section of the report, we will explore the implementation of learning algorithms for classification like KNN, Random Forest, SVM for classifying the data about silhouettes of the vehicles and classification of those data points into different classes of vehicles.
We will experiment and try to get the maximum accuracy from these algorithms.
After that we will analyze results and get more understanding about the working or ML algorithms and dataset characteristics and their effect on the accuracy of the algorithms.

Applications of Big Data

In this section, we will take a look at various applications of Big Data.
We will also reflect upon the Advantages, Disadvantages, and challenges regarding big data.

1. Retails

The retail market makes up a large chunk of the global economy. With the rise of online shopping platforms, the retail companies have started leveraging the big data to their advantages.
In the past few years due to digitization, large amounts of data have been collected by retailers. This data includes information about customers and their purchasing and other activities.
This data helps companies to understand their customers and the market trends better.
Companies can use Big Data Analytics for the purposes such as,

- Companies can microsegment the customers and analyze the data like the time and location of purchase. This helps understand the preferences and likes of the user which helps provide better customization and personalized experience to users.
- Pricing of the product is one of the most challenging tasks. Big data analytics help optimize the price of products by understanding the pattern and finding out which prices lead to more sales.
- Predictive analytics helps companies to predict events even before they have happened, and this helps them make decisions regarding the future strategies.
- Companies can use big data analytics to prevent incidents of fraud and minimize losses. For example, to detect online fraud, companies can process their transactions against pre-defined fraud patterns in real time.

(Akter, S., Wamba, S.F. Big data analytics in E-commerce: a systematic review and agenda for future research,2016)

Use Case: Recommendation System of Online Ecommerce Platforms

One of the well-known use cases of BDA in retail is the recommendation system of any online retail store. These systems analyze the purchase history of use r and provide the recommendation on the basis of that information.

Challenges in Use of Big Data in Retails

- The first of them is collection and integration of data. The data for the retail industry is of varying types and comes from different locations. So, companies need to put efforts in collection and integration of the data.
- Another issue is about the quality of the data. The collected data needs to be processed and the meaningful data has to be extracted from the large amounts of the data. Cause the credibility of the data being collected is not guaranteed and there is a lot of noisy data.
- The most important challenge of all is the privacy concerns with using the collected data. Organizations need to take care of not breaching any privacy laws like GDPR and has to ensure the security of the data of the users.

(Aktas E, Meng Y, An Exploration of Big Data Practices in Retail Sector, 2017)

2. Finance

Finance and accounting organizations can use the BDA for purposes like,

- Use of BDA to make much better forecast and detailed forecast. The accuracy of the forecasting is improving as more and more data is becoming available for analysis.
- BDA can optimize the budgeting by combining various types of financial and non-financial data and create more comprehensive reporting.
- Improved Risk Management. Banks and financial institutions have large amounts of data stored, which can be analyzed and continuously audited to for fraud prevention and risk management.

Use Case: Danske Bank Fraud Detection System

- Danske Bank utilized a data driven methodology that works with machine learning and deep learning. The goal was to increase fraud detection rate and decrease false positive results. The result with using machine learning reaped a decrease in false positives by thirty five percent (35%). And by adding deep learning, accuracy of fraud detection rate increase by approximately fifty percent (50%)

(Big Data,Cloud Computing and Data Science Applications in Finance and Accounting,Jennifer Huttunen,Jaana Jauhiainen,Jaura Lehit,Annina Nylund,Minna Martikainen,Othmar Lehner,2019)

Challenges in Use of Big Data in Finance

- The biggest concern is with security and the privacy of the data, as data theft or security breach can cause substantial damage to the customers as well as the institution.
- Another major issue is about the regulations. Many of the financial institutions operate in multiple regions and countries. With each country, the rules and regulations for the use of data change. So, it becomes a very meticulous task to comply with different regulations and standards regarding the data.

(Big Data,Cloud Computing and Data Science Applications in Finance and Accounting,Jennifer Huttunen,Jaana Jauhiainen,Jaura Lehit,Annina Nylund,Minna Martikainen,Othmar Lehner,2019)

3. Healthcare

Healthcare is another domain which can be affected by Big Data.
The applications of big data in healthcare are,

- Much better predictions about disease using the past data.
- Using Big Data to create personalized medicinal plan as per the patient's lifestyle and habits.
- There are mobile healthcare apps which use Big Data analysis to help patients.
- Wearable devices like smartwatches can collect the physical data of the patients in real time and that data can be analyzed to detect any anomalies and issues.
- Big Data can also be used to collect, store, and maintain patients' data in Hospital Management Systems.
- Big Data can also be used in management of healthcare resources by government by recognizing the areas of concern and action required.

- Big data can be used to monitor, collect, and analyze the data of patients suffering with chronic diseases such as diabetes and help improve their lifestyle.

(Mumtaz Karatas, Levent Eriskin, Muhammet Deveci, Dragan Pamucar, Harish Garg, Big Data for Healthcare Industry 4.0: Applications, challenges and future perspectives,2022)

Use Case: IBM's Watson for Oncology

It is a clinical decision-support system for oncology therapy selection. It analyses medical literature, clinical trial data, and patient records and helps doctors with the treatment of the patient.

(Liu C, Liu X, Wu F, Xie M, Feng Y, Hu C. Using Artificial Intelligence (Watson for Oncology) for Treatment Recommendations Amongst Chinese Patients with Lung Cancer: Feasibility Study,2018)

Challenges in Use of Big Data in Healthcare

- Integration of data, as healthcare data has various sources and types. This means there is inheritance heterogeneity in the healthcare data. So, frameworks would be required to collect and integrate all that data.
- Also, the IOT based application, wearables and other sensor-based data collection leads to large amounts of data, so its required to process that data to extract the meaning full data. Machine learning algorithms can be used for these tasks.
- Privacy of the data is also a concern like any other application of Big Data.

(Mumtaz Karatas, Levent Eriskin, Muhammet Deveci, Dragan Pamucar, Harish Garg, Big Data for Healthcare Industry 4.0: Applications, challenges, and future perspectives,2022)

4. Manufacturing

Big Data and Analytics have various application in the manufacturing sectors such as,

- It can be used to create a predictive manufacturing environment, where the health and status of the machine and equipment is continuously monitored and analyzed. This can help reduce the uncertainties in machine failures.
- Optimizing the product designs.
- By analyzing the data collected, the production scheduling can be done to optimize the output and decrease the idle time.
- Big Data Analytics can be used to analyze the defect and quality issue and create a comprehensive plan for fixing the issues and quality control

Use Cases:

Some of the examples of the use of BDA are,

- Enterprise sourcing & performance excellence (ESPX) by Raytheon
- PredixTM by General Electric (GE)
- Mindsphere by Siemens
- Engine health monitoring Unit (EMU) by Rolls Royce
- Big data traffic information service by Toyota

(Mohd Azeem, Abid Haleem, Shashi Bahl, Mohd Javaid, Rajiv Suman, Devaki Nandan, Big data applications to take up major challenges across manufacturing industries: A brief review,2022)

Challenges in Use of Big Data in Manufacturing

- Implementation of Big Data in manufacturing is very resource intensive.
- Also, the legacy systems doesn't necessarily have the ability to support BDA application.
- Data Security is a major concern as the leak of data can lead to Intellectual Property Damages

(Mohd Azeem, Abid Haleem, Shashi Bahl, Mohd Javaid, Rajiv Suman, Devaki Nandan, Big data applications to take up major challenges across manufacturing industries: A brief review,2022)

5. Automotive Industry

Applications of Big Data Analytics in Automotive industry includes,

- Big Data can be used for training object detection and tracking algorithms.
- It can be used to analyze and detect behavioral patterns of drives from past data.
- It can be used in Diagnostics and detection of issues with vehicles. The data collected from the various sensors and ECUs can be used to find the root cause of the defect by looking at the data at the time of the occurrence of issue.
- Big data also helps in maintenance by predicting the issues that might arise in future.
- Autonomous driving is another application of big data.
- Telematics is also an application of Big Data. Telematics is basically a system that collects the data regarding driver's vehicles usage and other data related to vehicle. This data provides information about driver's behavior and vehicle health.
- BDA can also help improve the fuel efficiency of the vehicle.

Use cases:

- Tesla Autopilot system and Over the Air Update system
- Ubers Dynamic Pricing

Challenges in Use of Big Data in Manufacturing

- Data security and privacy
- The cost of implementation can be high.
- Legacy vehicles might not support the Big Data application, so system updates might be needed.
- The volume of data collected is very high, which makes the data management complex.
- Compliance with rules and regulations. Given that vehicle is a safety critical application while developing the BDA applications, other than regulations about data usage other guidelines also need to be followed.

6. Applications in other domains

Other than the applications mentioned above, big data also has applications in other fields like

- Infrastructure
- Energy Management
- Agriculture, where it can be used to predict weather and crop patterns and also provide reports on soil fertility.
- Transportation, where it can be used to optimize the traffic and travel routes.
- Big data is also used in the education domain, where it can help analyze a student's progress and history and create personalized study material and plan.

Summary:

To summarize the above section, even though Big Data has immense potential, there are still some domains which will need to be addressed. Currently the in manufacturing and Automotive domain system level overhaul is needed. Also, companies need to adapt to a data driven culture. And issues like data privacy, security and ethical compliance need to be managed continuously.

Machine Learning Algorithms and Their Role in solving the Classification and Clustering Problems

What are Machine Learning Algorithms?

These are basically computational models, which can be used to find the patterns in the data and make predictions based on it.

In order to improve the accuracy of those predictions we need to train these models with existing data.

So, machine learning algorithms are of two types,

- Supervised Learning Algorithms
- Unsupervised Learning algorithms

Supervised Learning algorithms are trained with labeled data and Unsupervised algorithms are trained with unlabeled data.

Basically, algorithms used for classification are Supervised Learning Algorithms and algorithms used for clustering are Unsupervised learning algorithms.

Role of Machine Learning algorithms in Big Data Analytics

One of challenges in Big Data is to find meaningful information from the large amount data at disposal.

For that purpose, data analysis is done. It usually consists of processes like data cleaning, data visualization, exploratory analysis to find patterns and anomalies.

Machine learning algorithms like the ones with have been explored above help to find the patterns in the data and classify the data.

These machine learning algorithms are capable of learning from the existing data to improve its accuracy and give better results with time.

We can use the big data to train and test these algorithms and then these algorithms can be used to find the patterns in new data.

Machine learning algorithms have capability of iterative and continuous improvement as these keep learning from the new coming data.

When looking at specifically the classification and clustering problems,

Machine Learning algorithms can classify the new data into respective classes and categories based on its learning from previous data.

Clustering algorithms can be used to segment the unlabeled data into different cluster and these clusters can used find the similarities in data and then can be used to do predictive analysis.

Clustering algorithms are helpful in finding the previously unknown patterns in the data.

Supervised Machine Learning algorithms for Classification

There are various algorithms for classification. We will take a look at a few of them.

1. Logistic Regression

Logistic regression is a model which is used for binary classification. Logistic regression is variant of a linear regression with difference being, output of linear regression is continuous. Whereas logistic regression gives the probability of given data point being in either of the classes.
This classification is done using a sigmoid function.
A logistic regression model has following requirements:

- Each observation is independent of the other.
- The dependent variable must be binary. For more than two categories SoftMax functions are used
- There should be no outliers in the dataset.

Advantages:

- Easy to implement.
- Very fast
- Good accuracy for simpler datasets

Disadvantages:

- Can lead to overfitting.
- Assumption of linearity
- Can be used to predict only discrete values.

(https://medium.com/@akshayjain_757396/advantages-and-disadvantages-of-logistic-regression-in-machine-learning-a6a247e42b20)

2. K Nearest Neighbors (KNN)

KNN is a non-parametric learning algorithm used for classification.
KNN predicts the label or value of a new data point by considering its K closest neighbors in the training dataset.
The KNN algorithm is easy to implement and versatile. It does not need any assumptions for the underlying data, and it can handle both numerical and categorical data.
The metric used by KNN to determine the similarity of data point to any class is distance.
The only hyperparameter required is K, which is set based on the values of features in dataset.

Advantages:

- Easy to Implement
- Adaptable, as the KNN algorithm stores all the data, whenever a new point is added, it adjusts itself accordingly.
- Only 1 hyperparameter.

Disadvantages:

- Does not scale.
- It has a curse of dimensionality.
- Prone to overfitting

(https://www.geeksforgeeks.org/k-nearest-neighbours/)

3. Random Forest

Random forest makes use of the concept of Ensemble Learning, which means combining multiple classifiers to solve a problem.
It is a classifier that contains a number of decision trees on various subsets of the given dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
The greater the number of trees in the algorithm, more the accuracy of the model and it prevents overfitting.

Advantages:

- Takes less training time.
- Can run efficiently even with a large dataset.
- Can maintain the accuracy, when the large portion of dataset is missing.
- Prevents overfitting.

Disadvantages:

- Not very efficient in regression tasks
- We do not have much control on how algorithm works.
- Can be computationally intensive.

(https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04)

4. Support Vector Machines (SVM)

SVM can be used for both the Linear and nonlinear classification problems. Though it can be used for both regression and classification, it is much more suited for classification.
SVM, finds an optimal hyperplane that best classifies the classes that need to be classified. Support vectors are the closest data points from the hyperplane. SVN tries to maximize the distance of these support vectors from hyperplane, which is called the margin.
In the case of the linear classification the hyperplane is simply a linear equation. i.e.mx+c=0

Advantages:

- More effective in high dimensional spaces
- Effective in cases where the number of dimensions is greater than the number of samples.
- Relatively memory efficient

Disadvantages:

- Not suited for large datasets
- Does not perform well when the datapoints of classes are overlapping.
- As SVM works by putting points, above and below the hyperplane, there is not probabilistic value for classification

(https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107)

Unsupervised Machine Learning algorithms for Clustering

1. K-Means Clustering

K Means clustering algorithm, separates the unlabeled data into different groups of points which are comparable to each other and different from the data points in the other groups.
K in this algorithm represents the number of groups the points need to be divided into. We can provide the value of K to the algorithm.
The algorithm first randomly plots the K points as centers and starts categorizing data using them as reference for each class.

Advantages:

- Easy to understand and implement.
- Requires only a few hyperparameters.
- Fast and scalable
- Can handle the noise and outliers.

Disadvantages:

- Sensitivity to value of K and initial centroid locations
- Its hand to determine the optimal value of K.
- It assumes clusters are spherical and have similar variance, which may not be suitable for large and complex datasets.
- Can be affected by skewed and correlated variables.

(https://www.linkedin.com/advice/3/what-advantages-disadvantages-using-k-means)

2. Hierarchical Clustering

Hierarchical clustering is of two types,

- Divisive Clustering
- Agglomerative Clustering

In the case of Divisive clustering, the whole dataset is considered as one big cluster and then it is broken down into smaller clusters.
Whereas in agglomerative clustering, every datapoint is considered as individual cluster and then they are combined to create single cluster, until a threshold criterion is reached.
Output of hierarchical clustering is tree-like structure, called a dendrogram, which illustrates the hierarchical relationships among the clusters.

Advantages:

- Easy to implement.
- Output is in the form of hierarchy rather than flat clusters which makes it easier to understand the correlation.
- Ability to handle non-convex clusters and clusters of different sizes and densities.

Disadvantages:

- Not good for large datasets
- Sensitive to outliers
- Results dependent on initial conditions, linkage criterion, and distance metric used.

(Ahuja, R., Chug, A., Gupta, S., Ahuja, P., Kohli, S, Classification and Clustering Algorithms of Machine Learning with their Applications,2020)

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

The DBSCAN algorithm is based on the idea that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
Other clustering methods like KNN and hierarchical clustering tend to cluster the data in spherical or convex shape. Also, they are affected by outliers and noise in data.
DBSCAN needs two parameters:

a. Esp

This parameter defines the neighborhood of the point. If the distance of other point is less than esp. it considered as part of cluster and vice versa.
It's important to choose the value of this parameter carefully as too lower value will make other points outliers and too high value will make more than suitable points as part of the cluster

b. Minpts

This parameter defines the minimum number of points within esp radius. For larger data sets this value has to be high.

Advantages:

- Can discover clusters of arbitrary shapes.
- Can not be affected by noise easily.
- Doesn't need number clusters to be defined in advance.

Disadvantages:

- Sensitive to choose of esp and minpts
- Doesn't work well with clusters of varying density.
- Not guaranteed to find all clusters in dataset.

(https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/)

Classification of vehicles based on the data of silhouettes using the ML algorithms.

In this section, we will be performing the classification on a dataset that contains the data about the shape of silhouettes of vehicles of different type. We will be using that data to train an algorithm to classify those data points in vehicle types.
We will be using R to analyze the data and train, test and evaluate our model.
Libraries/packages Used while building and evaluating models:

- base
- caTools
- class
- dplyr
- caret
- randomForest
- e1071
- kernlab

A. Descriptive and Visual Analysis of the dataset

1. Structure of the data set:

After importing the dataset in CSV format to a data frame, we can check the structure and specification of our dataset using following command:

```
str(vehicles)
```

The output of the command will be,

```
'data.frame':   846 obs. of  19 variables:
 $ compactness                : int  95 91 104 93 85 107 97 90 86 93 ...
 $ circularity                : int  48 41 50 41 44 NA 43 43 34 44 ...
 $ distance_circularity       : int  83 84 106 82 70 106 73 66 62 98 ...
 $ radius_ratio               : int  178 141 209 159 205 172 173 157 140 NA ...
 $ pr.axis_aspect_ratio       : int  72 57 66 63 103 50 65 65 61 62 ...
 $ max.length_aspect_ratio    : int  10 9 10 9 52 6 6 9 7 11 ...
 $ scatter_ratio              : int  162 149 207 144 149 255 153 137 122 183 ...
 $ elongatedness              : int  42 45 32 46 45 26 42 48 54 36 ...
 $ pr.axis_rectangularity     : int  20 19 23 19 19 28 19 18 17 22 ...
 $ max.length_rectangularity  : int  159 143 158 143 144 169 143 146 127 146 ...
 $ scaled_variance            : int  176 170 223 160 241 280 176 162 141 202 ...
 $ scaled_variance.1          : int  379 330 635 309 325 957 361 281 223 505 ...
 $ scaled_radius_of_gyration  : int  184 158 220 127 188 264 172 164 112 152 ...
 $ scaled_radius_of_gyration.1: int  70 72 73 63 127 85 66 67 64 64 ...
 $ skewness_about             : int  6 9 14 6 9 5 13 3 2 4 ...
 $ skewness_about.1           : int  16 14 9 10 11 9 1 3 14 14 ...
 $ skewness_about.2           : int  187 189 188 199 180 181 200 193 200 195 ...
 $ hollows_ratio              : int  197 199 196 207 183 183 204 202 208 204 ...
 $ class                      : chr  "van" "van" "car" "van" ...
```

The dataset has a total of 19 variables out of which 18 are the input features and 1 is the target variables, which we will be trying to predict.

2. Summary and Distribution of all the Input variables:

Now, we will look at every individual feature and try and analyze that variable by creating a histogram to visualize the distribution of values of that feature.
This will be achieved using following code,

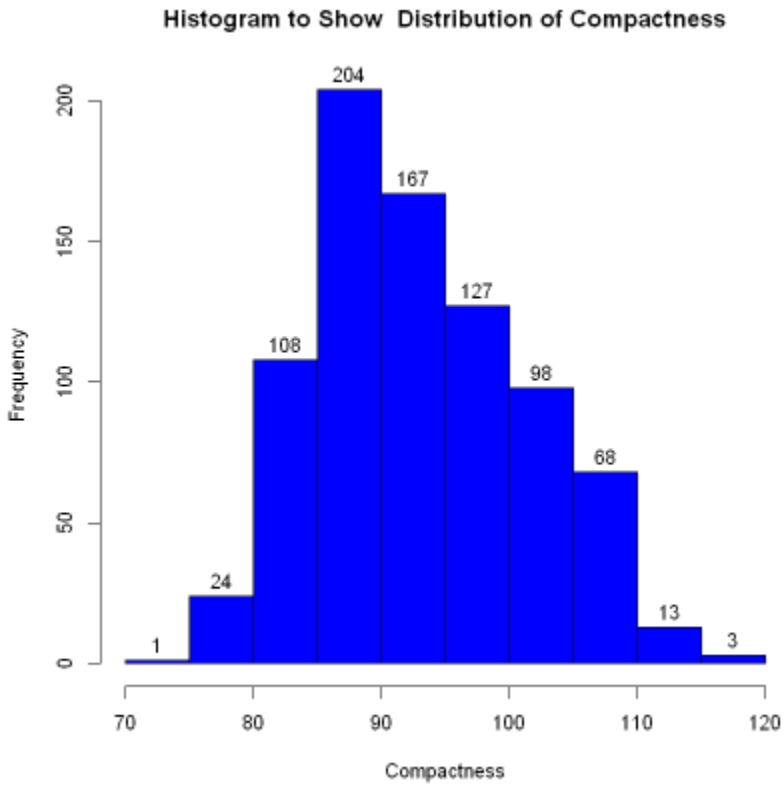
```
summary(vehicles$feature)
hist(vehicles$feature,
     col="blue",
     main="Histogram to Show  Distribution of Feature",
     xlab="Feature",
     ylab="Frequency",
   labels=TRUE)
```

a. Compactness

Summary:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  73.00   87.00   93.00   93.66  100.00  119.00
```

Distribution:

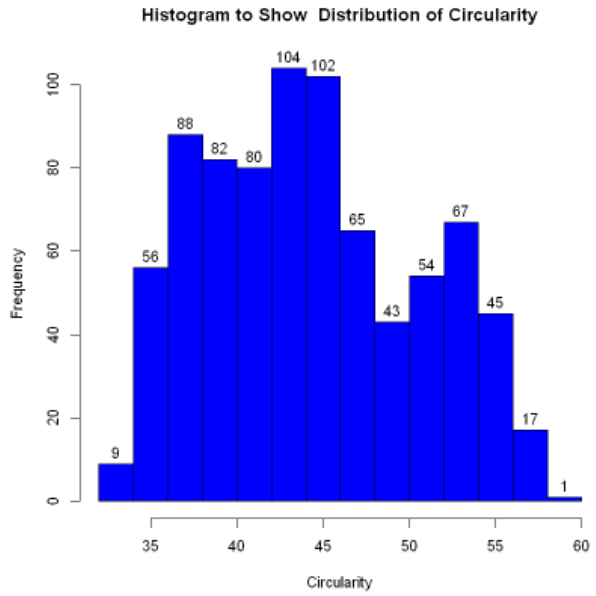


b. Circularity

Summary:

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
33.0    40.0    44.0    44.8    49.0    59.0
```

Distribution:

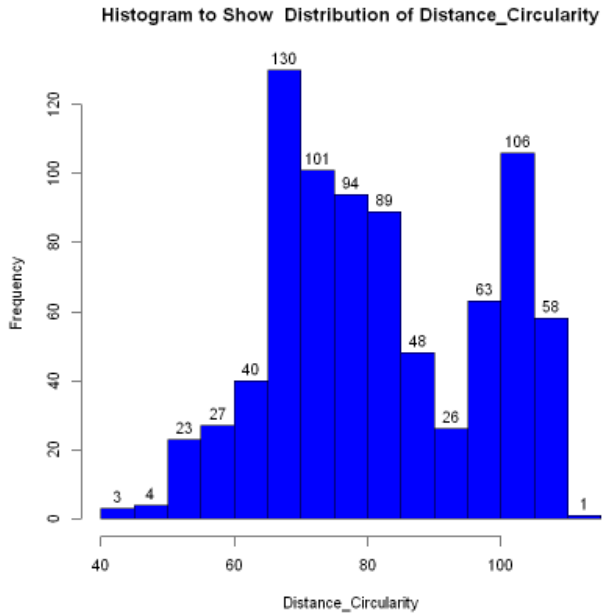


c. Distance Circularity

Summary:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 40.00   70.00   79.00   82.04   98.00  112.00
```

Distribution:

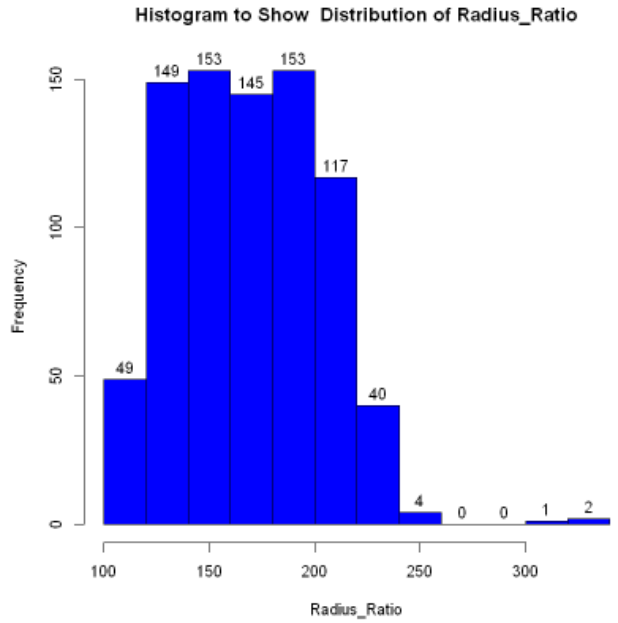


d. Radius Ratio

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
104.0   141.0   167.0   169.1   195.0   333.0
```

Distribution:

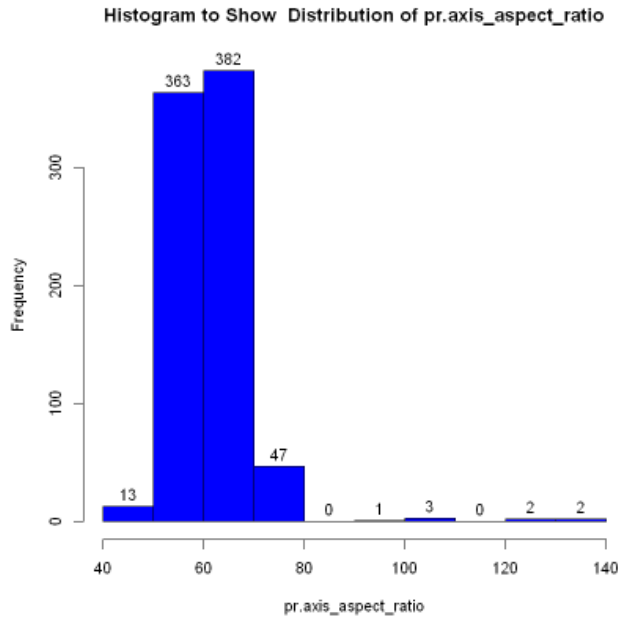


e. pr.axis_aspect_ratio

Summary:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 47.00   57.00   61.00   61.77   65.00  138.00
```

Distribution:

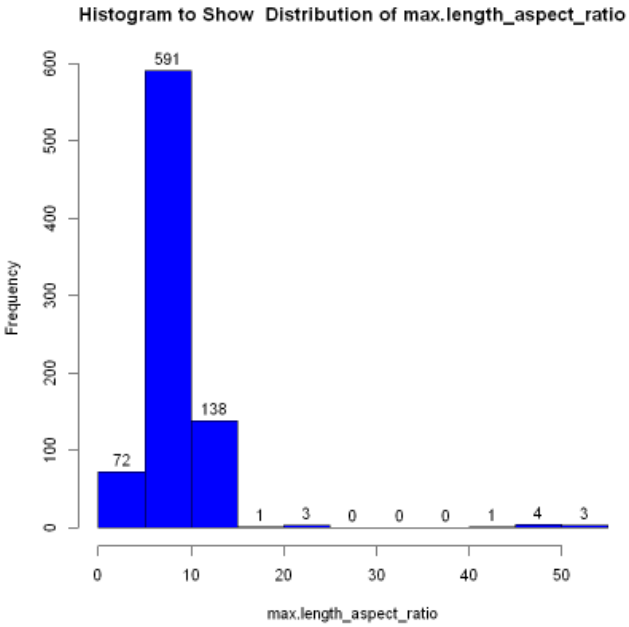


f. max.length_aspect_ratio

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2.000   7.000   8.000   8.599  10.000  55.000
```

Distribution:

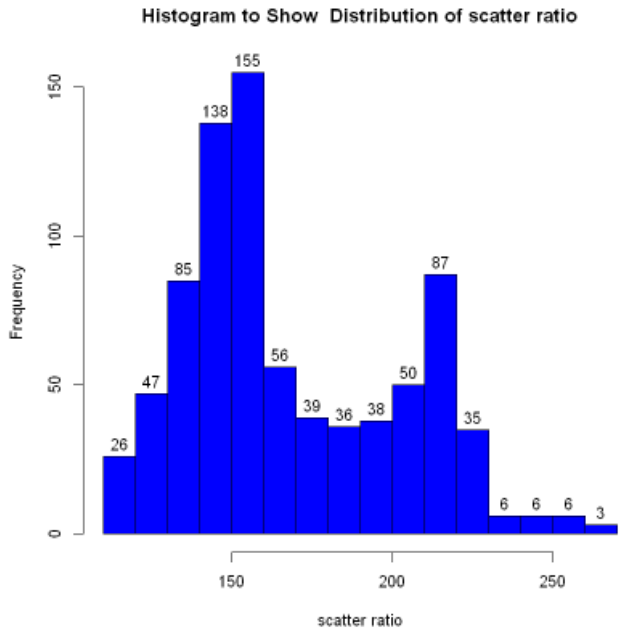


g. scatter ratio

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
112.0   146.0   157.0   168.6   198.0   265.0
```

Distribution:

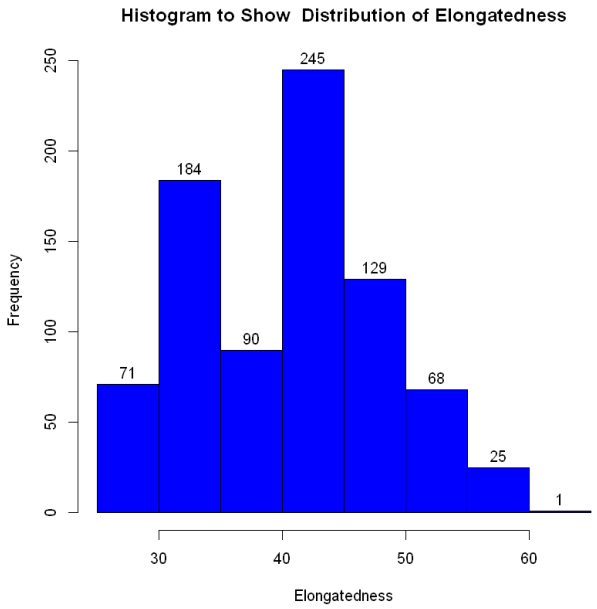


h. Elongatedness

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
26.00   33.00   43.00   40.99   46.00   61.00
```

Distribution:

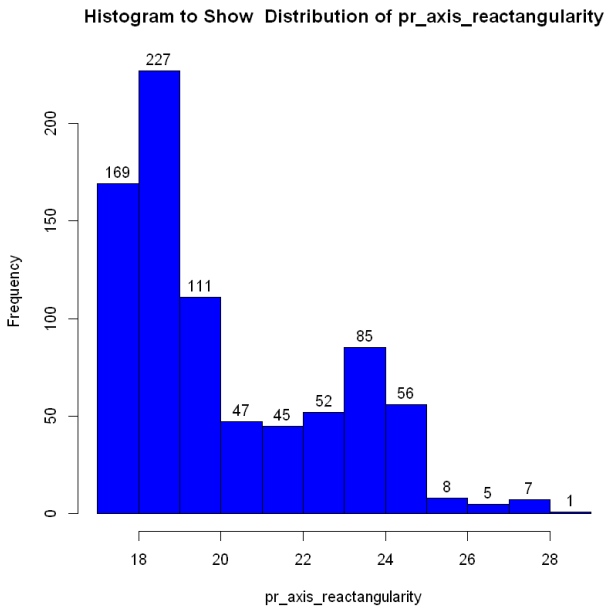


i. pr_axis_reactangularity

Summary:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
17.00   19.00   20.00   20.56   23.00   29.00
```

Distribution:

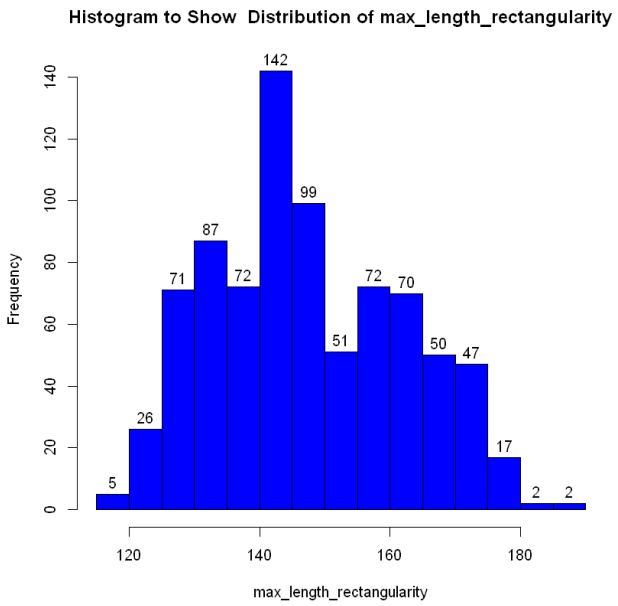


j. max_length_rectangularity

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
118.0   137.0   146.0   147.9   159.0   188.0
```

Distribution:

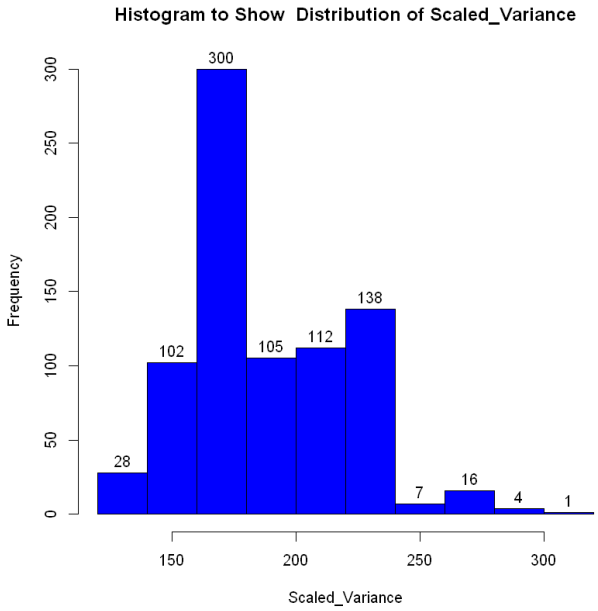


k. Scaled Variance

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
130.0   167.0   179.0   188.4   217.0   320.0
```

Distribution:

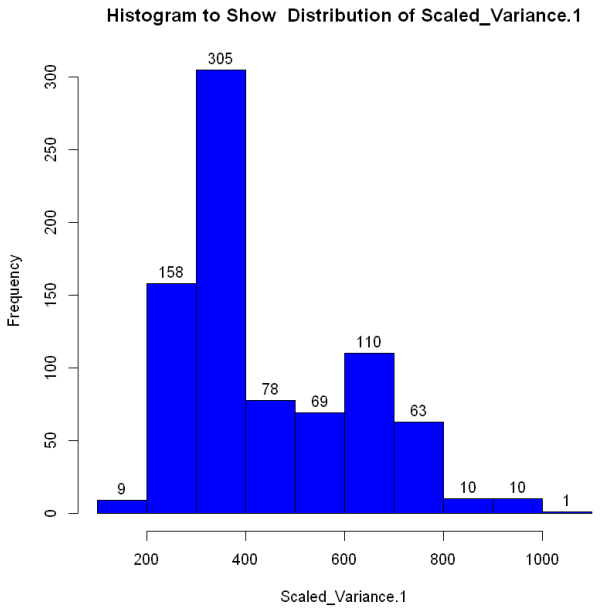


l. Scaled Variance 1

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
184.0   318.0   364.0   438.4   586.0  1018.0
```

Distribution:

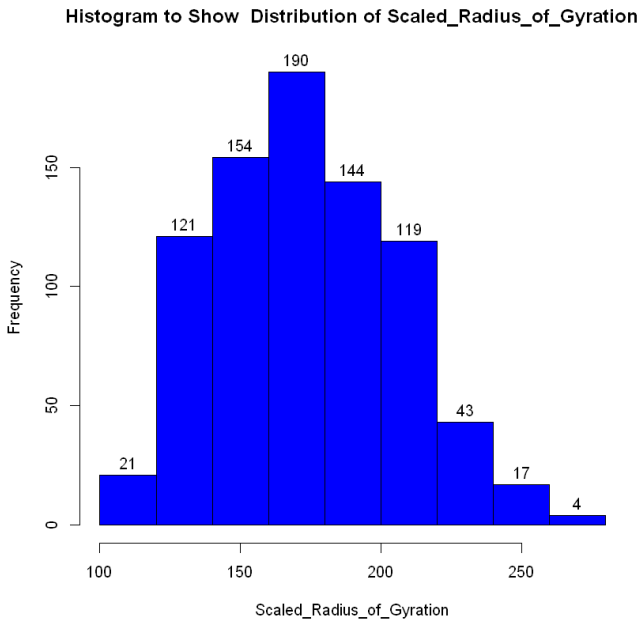


m. Scaled_Radius_of_Gyration

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
109.0   149.0   173.0   174.3   198.0   268.0
```

Distribution:

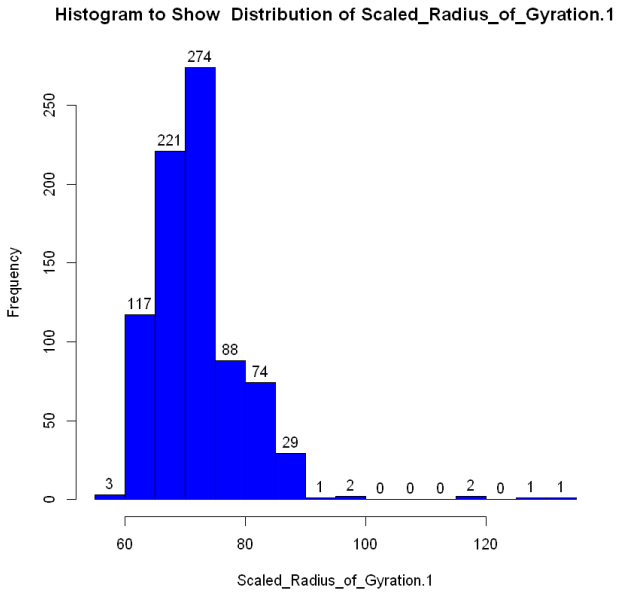


n. Scaled_Radius_of_Gyration 1

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 59.0    67.0    71.0    72.4    75.0   135.0
```

Distribution:

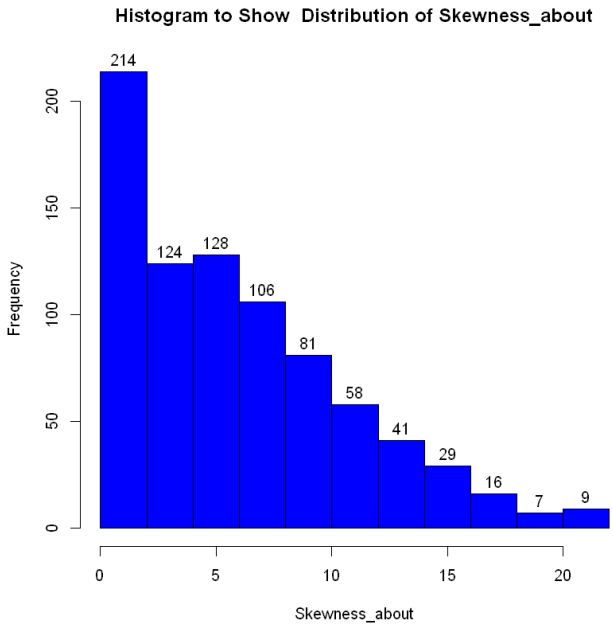


o. Skewness about

Summary:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   2.000   6.000   6.352   9.000  22.000
```

Distribution:

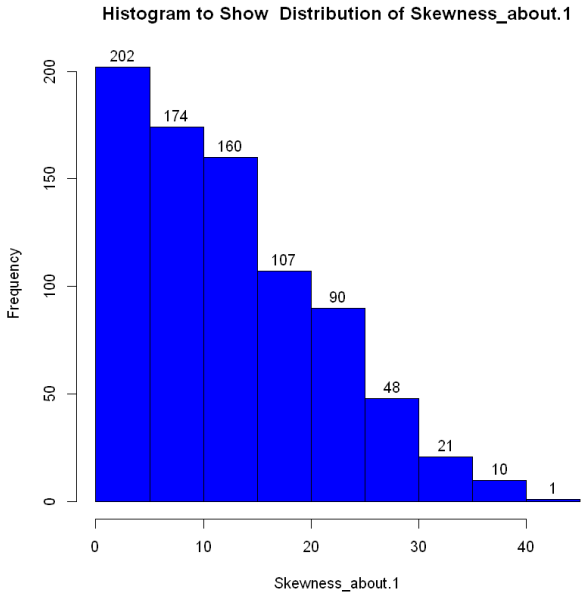


p. Skewness about 1

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00    6.00   11.00   12.69   19.00   41.00
```

Distribution:

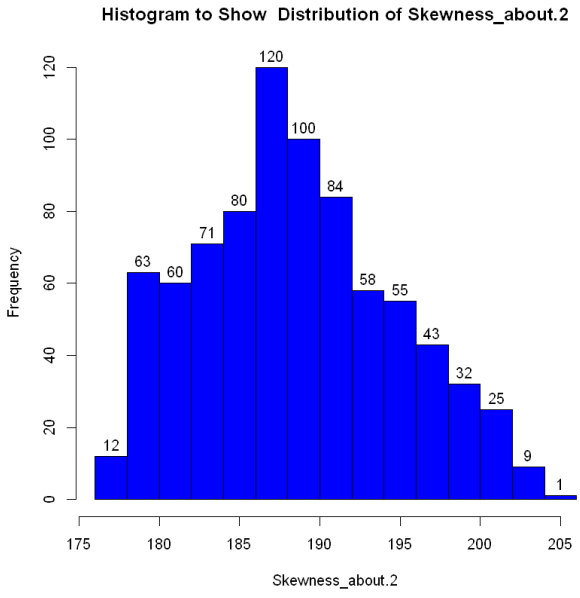


q. Skewness about 2

Summary:

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  176     184     189     189     193     206
```

Distribution:

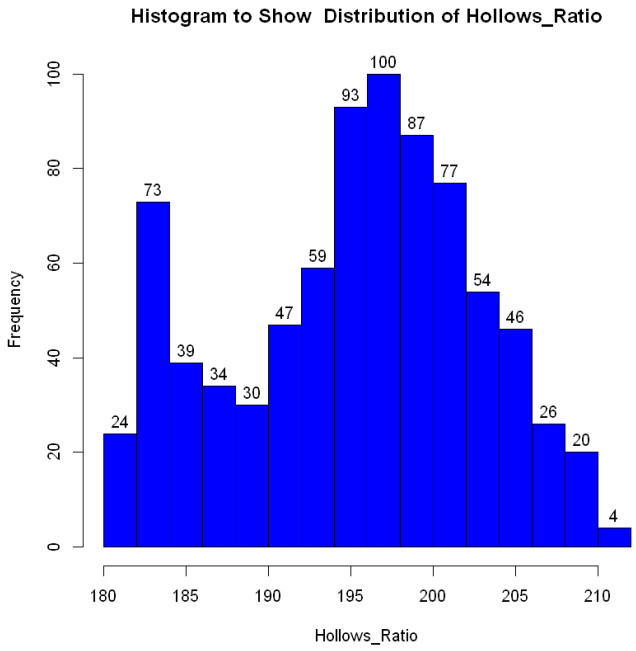


r. Hollows_Ratio

Summary:

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 181.0   191.0   197.0   195.7   201.0   211.0 
```

Distribution:



Summary:

Above data and data visualization can be used to analyze the data and find out the outliers and skewness of the data.

As we can see from the above stats, some of the features have outlier values.

These outliers' values are determined by following method,

- Calculate
  IQR = 3rd Quarter – 1st Quarter
- Now the upper and lower thresholds will be.
  Upper = 3rd Quarter + 1.5*IQR
  Lower = 1st Quarter – 1.5*IQR

Any values beyond these thresholds can be considered as outliers.

From the analysis we found out that many of the variables have outlier values in their distribution.

So, as we will see in the next section we will train and evaluate our model with the original dataset and then with filtered dataset by removing the outliers and compare the results.

We can also use descriptive analysis to look at the central tendency, spread of the data and shape of distribution. We analyzed the normality of distribution and consistency and symmetry of the spread.

3. Preparation of data to be used in the training and evaluation of models.

Once we are done with the visualization and analysis of dataset, we will prepare our data set for the purposes of training and evaluating the model generated.

The next code bit will be used for that,

```
split = sample.split(vehicles,SplitRatio = 0.7)
train = subset(vehicles,split==TRUE)
test = subset(vehicles,split==FALSE)
train$class <- NULL
test$class <- NULL
trainingoutcomes <- subset(vehicles,split == TRUE)
testoutcomes <- subset(vehicles,split==FALSE)
trainingoutcomes <- trainingoutcomes[,c("class")]
testoutcomes<- testoutcomes[,c("class")]
```

4. Creating a KNN model and evaluating it

Now we will create a KNN model and create a confusion matrix to analyse the performance of the model.

A. Using original data

```
predictions <- knn(train = train,test = test,cl=trainingoutcomes,k=1)
testoutcomes <- factor(testoutcomes,levels = levels(predictions))
cm <- confusionMatrix(testoutcomes,predictions)
```

The output confusion matrix for the model is as follow:

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  44   3   1
       car  13 120  14
       van   6   2  53

Overall Statistics

               Accuracy : 0.8477
                 95% CI : (0.7977, 0.8894)
    No Information Rate : 0.4883
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7503

 Mcnemar's Test P-Value : 0.0002977

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.6984     0.9600     0.7794
Specificity              0.9793     0.7939     0.9574
Pos Pred Value           0.9167     0.8163     0.8689
Neg Pred Value           0.9087     0.9541     0.9231
Prevalence               0.2461     0.4883     0.2656
Detection Rate           0.1719     0.4688     0.2070
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.8388     0.8769     0.8684
```

We were able to achieve the accuracy of 84.77 percent with this model.
The significance of the other statistics in the above model and what is the significance of them is as follow:

i. Accuracy:
Accuracy is the ratio of correctly predicted instances to the total number of instances.
We were able to get 84.77 percent accuracy on this model.

ii. 95% CI (Confidence Interval):
This statistic tells us about the range of values between which the true accuracy of the model might lie. This accuracy is predicted with 95% confidence, due to that it's called 95% CI.
As per that, accuracy of the model above might be between, 79.77% to 99.94%

iii. No. Information rate:
This is the accuracy of the model when it always predicts the majority class as the output for any given instance. High value of NIR means one of the classes is more dominant. Now given that we have 3 classes the ideal value of NIR would be 33.33 %. But we are getting the value of 48.83%.

iv. P-Value:
P value measures whether the accuracy of the model is greater than what it would be if prediction is made with random chance. A low P-value means the accuracy of the model is significantly better than a random chance.
We have the P value of $2.2*10^{-16}$.

v. Cohen's Kappa:
This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.
The value of kappa in our case is 0.75.

vi. McNemars Test P value
the p-value for a statistical test comparing the number of false positives and false negatives. A large p-value (typically greater than 0.05) indicates that there is no significant difference between the number of false positives and false negatives. Value of the parameter in our case is 0.0002777.

(https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/)

B. Using Scaled Training and Testing datasets

```
train_scaled <- scale(train[])
test_scaled <- scale(test[])
predictionsforscaled <- knn(train = train_scaled,test = test_scaled,
                            cl = trainingoutcomes,k = 1)
cm_scaled <- confusionMatrix(testoutcomes,predictionsforscaled)
```

Output Confusion matrix for the model when trained with scaled datasets is as follow

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  46   1   1
       car   8 129  10
       van   7   3  51

Overall Statistics

               Accuracy : 0.8828
                 95% CI : (0.8369, 0.9195)
    No Information Rate : 0.5195
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8045

 Mcnemar's Test P-Value : 0.003322

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.7541     0.9699     0.8226
Specificity              0.9897     0.8537     0.9485
Pos Pred Value           0.9583     0.8776     0.8361
Neg Pred Value           0.9279     0.9633     0.9436
Prevalence               0.2383     0.5195     0.2422
Detection Rate           0.1797     0.5039     0.1992
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.8719     0.9118     0.8855
```

Scaling the datasets is one of the solutions to increase the accuracy of our model. Scaling especially helps in the case of KNN as it's a distance-based algorithm. Cause if one of the data points has a high value it can affect the model disproportionately.
So, when we scale the data, it brings all the data points on similar scale. It also normalizes the variables. All this leads to much better distribution of data and all features contribute equally to modeling and improve accuracy.
In our case we were able to achieve the accuracy of 88.28 percent with scaled data.

(https://medium.com/analytics-vidhya/why-is-scaling-required-in-knn-and-k-means-8129e4d88ed7)

C. Using the filtered data by removing outliers

While filtering the data, initially the threshold values derived from the calculations using the formula discussed earlier in the report.
The code for filtering the data is as follow,

```
filterdata<- vehicles[vehicles$radius_ratio < 276,]
filterdata <- vehicles[vehicles$compactness > 77 & vehicles$compactness < 111,]
filterdata <- filterdata[filterdata$pr.axis_aspect_ratio < 77,]
filterdata <- filterdata[filterdata$max.length_aspect_ratio < 15,]
filterdata <- filterdata[filterdata$scatter_ratio < 230,]
filterdata <- filterdata[filterdata$scaled_variance < 292,]
filterdata <- filterdata[filterdata$scaled_variance.1 < 988,]
filterdata <- filterdata[filterdata$scaled_radius_of_gyration.1 < 84 &
filterdata$scaled_radius_of_gyration.1 >55,]
filterdata <- filterdata[filterdata$skewness_about < 20,]
filterdata <- filterdata[filterdata$skewness_about.1 < 39,]
```

When trained using this data we got the accuracy of 89.19 and 91.89 percent for unscaled and scaled data resp.

For Un-scaled data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  41  11   2
       car   8 100   9
       van   0   4  47

Overall Statistics

               Accuracy : 0.8468
                 95% CI : (0.7926, 0.8915)
    No Information Rate : 0.518
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7503

 Mcnemar's Test P-Value : 0.2217

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.8367     0.8696     0.8103
Specificity              0.9249     0.8411     0.9756
Pos Pred Value           0.7593     0.8547     0.9216
Neg Pred Value           0.9524     0.8571     0.9357
Prevalence               0.2207     0.5180     0.2613
Detection Rate           0.1847     0.4505     0.2117
Detection Prevalence     0.2432     0.5270     0.2297
Balanced Accuracy        0.8808     0.8553     0.8930
```

For Scaled data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  52   1   1
       car   3 106   8
       van   1   4  46

Overall Statistics

               Accuracy : 0.9189
                 95% CI : (0.8749, 0.9512)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8688

 Mcnemar's Test P-Value : 0.5062

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9286     0.9550     0.8364
Specificity              0.9880     0.9009     0.9701
Pos Pred Value           0.9630     0.9060     0.9020
Neg Pred Value           0.9762     0.9524     0.9474
Prevalence               0.2523     0.5000     0.2477
Detection Rate           0.2342     0.4775     0.2072
Detection Prevalence     0.2432     0.5270     0.2297
Balanced Accuracy        0.9583     0.9279     0.9032
```

Then we filtered the original data by using the observation from the visualization analysis and the calculated thresholds to modify the filtering as follow:

```
filterdata<- vehicles[vehicles$radius_ratio < 276,]
filterdata <- filterdata[filterdata$pr.axis_aspect_ratio < 80,]
filterdata <- filterdata[filterdata$max.length_aspect_ratio < 15,]
filterdata <- filterdata[filterdata$scatter_ratio < 230,]
filterdata <- filterdata[filterdata$elongatedness < 60,]
filterdata <- filterdata[filterdata$pr.axis_rectangularity < 25,]
filterdata <- filterdata[filterdata$max.length_rectangularity < 180,]
filterdata <- filterdata[filterdata$scaled_variance < 280,]
filterdata <- filterdata[filterdata$scaled_variance.1 < 800 &
filterdata$scaled_variance.1 >200,]
filterdata <- filterdata[filterdata$scaled_radius_of_gyration.1 < 90 &
filterdata$scaled_radius_of_gyration.1 >60,]
filterdata <- filterdata[filterdata$skewness_about.2 < 204,]
```

After using the dataset from this data, we were able to achieve the accuracy of 89.19% and 94.14 percent respectively.

For Un-scaled data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  55   9   2
       car   7  83   3
       van   1   2  60

Overall Statistics

               Accuracy : 0.8919
                 95% CI : (0.8434, 0.9295)
    No Information Rate : 0.4234
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.835

 Mcnemar's Test P-Value : 0.8534

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.8730     0.8830     0.9231
Specificity              0.9308     0.9219     0.9809
Pos Pred Value           0.8333     0.8925     0.9524
Neg Pred Value           0.9487     0.9147     0.9686
Prevalence               0.2838     0.4234     0.2928
Detection Rate           0.2477     0.3739     0.2703
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9019     0.9024     0.9520
```

For Scaled data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  65   1   0
       car   1  91   1
       van   1   9  53

Overall Statistics

               Accuracy : 0.9414
                 95% CI : (0.9019, 0.9685)
    No Information Rate : 0.455
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.91

 Mcnemar's Test P-Value : 0.06018

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9701     0.9010     0.9815
Specificity              0.9935     0.9835     0.9405
Pos Pred Value           0.9848     0.9785     0.8413
Neg Pred Value           0.9872     0.9225     0.9937
Prevalence               0.3018     0.4550     0.2432
Detection Rate           0.2928     0.4099     0.2387
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9818     0.9422     0.9610
```

D. Using other Learning Model

1. Random Forest Model

Code bit for the model Implementation and evaluation,

```
target_var <- subset(vehicles$class,split==TRUE)
target_var <- factor(target_var)
formula <- target_var ~ .
rfmodel<- randomForest(formula,data = train)
rfmodpred<- predict(rfmodel,newdata = test)
cm_rf <- confusionMatrix(testoutcomes,rfmodpred)
```

Results of the evaluation are as follows,

a. Using original Data,

With Un-scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  48   0   0
       car   2 131  14
       van   2   2  57

Overall Statistics

               Accuracy : 0.9219
                 95% CI : (0.8819, 0.9516)
    No Information Rate : 0.5195
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8692

 Mcnemar's Test P-Value : 0.004637

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9231     0.9850     0.8028
Specificity              1.0000     0.8699     0.9784
Pos Pred Value           1.0000     0.8912     0.9344
Neg Pred Value           0.9808     0.9817     0.9282
Prevalence               0.2031     0.5195     0.2773
Detection Rate           0.1875     0.5117     0.2227
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.9615     0.9274     0.8906
```

With Scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  48   0   0
       car   2 131  14
       van   2   2  57

Overall Statistics

               Accuracy : 0.9219
                 95% CI : (0.8819, 0.9516)
    No Information Rate : 0.5195
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8692

 Mcnemar's Test P-Value : 0.004637

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9231     0.9850     0.8028
Specificity              1.0000     0.8699     0.9784
Pos Pred Value           1.0000     0.8912     0.9344
Neg Pred Value           0.9808     0.9817     0.9282
Prevalence               0.2031     0.5195     0.2773
Detection Rate           0.1875     0.5117     0.2227
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.9615     0.9274     0.8906
```

While using the original data,
We were able to get the accuracy of 92.19% with Un-scaled data and the accuracy remained unchanged with scaled data.

b. Using Filtered Data,

With Un-scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  64   2   0
       car   1  90   2
       van   1   3  59

Overall Statistics

               Accuracy : 0.9595
                 95% CI : (0.9244, 0.9813)
    No Information Rate : 0.4279
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.938

 Mcnemar's Test P-Value : 0.6746

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9697     0.9474     0.9672
Specificity              0.9872     0.9764     0.9752
Pos Pred Value           0.9697     0.9677     0.9365
Neg Pred Value           0.9872     0.9612     0.9874
Prevalence               0.2973     0.4279     0.2748
Detection Rate           0.2883     0.4054     0.2658
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9784     0.9619     0.9712
```

With Scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  64   2   0
       car   1  90   2
       van   1   3  59

Overall Statistics

               Accuracy : 0.9595
                 95% CI : (0.9244, 0.9813)
    No Information Rate : 0.4279
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.938

 Mcnemar's Test P-Value : 0.6746

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9697     0.9474     0.9672
Specificity              0.9872     0.9764     0.9752
Pos Pred Value           0.9697     0.9677     0.9365
Neg Pred Value           0.9872     0.9612     0.9874
Prevalence               0.2973     0.4279     0.2748
Detection Rate           0.2883     0.4054     0.2658
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9784     0.9619     0.9712
```

While using the Filtered data,

- We were able to get the accuracy of 95.95% percent.
- Using Scaled data for the model didn't improve the accuracy of the model.

The reason for the no change in accuracy of the model lies in the way the model works.
Random Forest ensembles the results from various decision tree, to give the result, this negates the impact of individual feature on the performance of model.
So when we scale the data, the model is unaffected cause it wasn't impacted from skewness of individual featured in the first place.

2. Support Vector Machine Model (SVM)

Code bit for the model Implementation and evaluation,

```
target_var <- subset(vehicles$class,split==TRUE)
target_var <- factor(target_var)
formula <- target_var ~ .
svmmodel <- svm(formula,data = train,kernel = "radial")
svmpreds <- predict(svmmodel,newdata = test)
cm_svm <- confusionMatrix(testoutcomes,svmpreds)
```

Results of the evaluation are as follows,

a. Using original Data

b.

With Un-scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  48   0   0
       car   0 142   5
       van   3   2  56

Overall Statistics

               Accuracy : 0.9609
                 95% CI : (0.9293, 0.9811)
    No Information Rate : 0.5625
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.933

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9412     0.9861     0.9180
Specificity              1.0000     0.9554     0.9744
Pos Pred Value           1.0000     0.9660     0.9180
Neg Pred Value           0.9856     0.9817     0.9744
Prevalence               0.1992     0.5625     0.2383
Detection Rate           0.1875     0.5547     0.2188
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.9706     0.9707     0.9462
```

With Scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  48   0   0
       car   0 141   6
       van   1   0  60

Overall Statistics

               Accuracy : 0.9727
                 95% CI : (0.9445, 0.9889)
    No Information Rate : 0.5508
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9534

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9796     1.0000     0.9091
Specificity              1.0000     0.9478     0.9947
Pos Pred Value           1.0000     0.9592     0.9836
Neg Pred Value           0.9952     1.0000     0.9692
Prevalence               0.1914     0.5508     0.2578
Detection Rate           0.1875     0.5508     0.2344
Detection Prevalence     0.1875     0.5742     0.2383
Balanced Accuracy        0.9898     0.9739     0.9519
```

- We were able to get the accuracy of 96.09% with original data.
- When using the scaled data, the accuracy increased to 97.27%

c. Using Filtered Data

With Un-scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  66   0   0
       car   1  90   2
       van   0   2  61

Overall Statistics

               Accuracy : 0.9775
                 95% CI : (0.9482, 0.9926)
    No Information Rate : 0.4144
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9657

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9851     0.9783     0.9683
Specificity              1.0000     0.9769     0.9874
Pos Pred Value           1.0000     0.9677     0.9683
Neg Pred Value           0.9936     0.9845     0.9874
Prevalence               0.3018     0.4144     0.2838
Detection Rate           0.2973     0.4054     0.2748
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9925     0.9776     0.9778
```

With Scaled Data,

```
Confusion Matrix and Statistics

          Reference
Prediction bus car van
       bus  66   0   0
       car   1  90   2
       van   0   3  60

Overall Statistics

               Accuracy : 0.973
                 95% CI : (0.9421, 0.99)
    No Information Rate : 0.4189
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9588

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: bus Class: car Class: van
Sensitivity              0.9851     0.9677     0.9677
Specificity              1.0000     0.9767     0.9812
Pos Pred Value           1.0000     0.9677     0.9524
Neg Pred Value           0.9936     0.9767     0.9874
Prevalence               0.3018     0.4189     0.2793
Detection Rate           0.2973     0.4054     0.2703
Detection Prevalence     0.2973     0.4189     0.2838
Balanced Accuracy        0.9925     0.9722     0.9745
```

When using the filtered data,
With unscaled data, the SVM model yielded 97.75% accuracy.
With scaled data, the accuracy was 97.30%

Results & Critical Reflection

Evaluation of the applications of Big Data:

In the first section of the report, we reviewed the applications of Big Data in various fields. After looking at all the applications, some of the observations are,

- The newly emerging fields like E-Commerce, social media and entertainment have adapted and leveraged Big Data and its applications very well. Given that these businesses are driven by data.
- Some of the conventional fields like Manufacturing and Automotive industry have started to embrace and use Big Data to their benefits, but they are still not up to the pace with the E-commerce and Entertainment field.
- These conventional fields have challenges like the cost of implementation and migration of their legacy system to the Big Data compatible systems.
- In the case of Automotive, the rules and regulations varying in different countries make it difficult to manage the data and its applications.
- One of the most common issues across all domains is the issue of data security and privacy.
- To accelerate and better regulate the use of Big Data, regulatory bodies and institutions need to frame the rules and guidelines in a way that they will be practical and applicable across the globe.
- Also, the framework for collection and storage of big data needs to be implemented in a much more comprehensive way for fields like agriculture in order to collect more and more data.

Machine Learning Algorithms and their Use in classification and Clustering Problems

- We looked at various machine learning algorithms for the classification and Clustering, and their pros and cons.
- There are various algorithms for clustering like K-Means, Hierarchical Clustering and DBSCAN
- For classification algorithms like Logistic Regression, KNN, Random Forest, SVM can be used.
- These algorithms can help in analyzing Big Data and they can be trained using Big data to do predictions and find the patterns in the data.

Implementation of ML algorithms and Results:

Out of all the algorithms, Implemented and review the KNN algorithm for the classification and we also implemented Support Vector Machine, Random Forest.

The resultant accuracy for the algorithms was as follow:

Accuracy of Models in Percentage	Original Dataset		Filtered Dataset	
	Un-Scaled	Scaled	Un-Scaled	Scaled
K-Nearest Neighbors	84.77	88.28	89.19	94.14
Support Vector Machine	96.09	97.27	97.75	97.3
Random Forest	92.19	92.19	95.95	95.95

For KNN Model,

- As we can see in the table the KNN algorithms showed a significant improvement in accuracy when comparing the accuracy for the Un-scaled and scaled data for both the original and filtered dataset.
- Also, the accuracy of KNN increased overall with Filtered Dataset.

For SVM Model,

- SVM algorithm also showed some improvement in accuracy with Scaled data compared to Un-scaled data with original dataset. For Filtered dataset, the accuracy nearly remained unchanged for SVM.
- The accuracy of SVM improved by some margin for Filtered dataset compared to original dataset in case of SVM.

For the Random Forest algorithm,

- The accuracy of random forest algorithm also increases when using the filtered dataset compared to original dataset.
- But the accuracy of the model remained the same in both the Unscaled and Scaled data.

Reason for the increase of accuracy in filtered data compared to original data are,

- The original dataset can have some outlier data instances, which was the case in the dataset used in this report.
- These outliers affect the accuracy of the models. When we remove the outliers from the dataset it leads to better normalization and better spread of distribution i.e. even and symmetric.
- Due to this reason, the accuracy of the models increased when using the filtered dataset.
- The outliers and anomalies in the dataset can be found during the Visual and descriptive analysis of the dataset. You can use the Inter-quartile range (IQR) to determine the outliers in the dataset.

Reasons for the increase in accuracy while using scaled data for SVM and KNN are,

- KNN uses the distance metric as a factor while making the predictions. SVN also tries to determine the hyperplane in a way that the margin between two classes is maximum. It uses the support vectors to decide the decision boundary and margin.

- Due to that these two algorithms are affected by scaling of the data. Cause when one the instance has a disproportionately more distance, it affects the model more. When the data is scaled this effect is minimized as all the data points are scaled to the same level.
- Although, KNN is affected more than SVM as KNN relies solely on the distance metric while making the decision. Whereas in the case of SVN, the relative position of Support matters more than the actual distance between them.

Reason that the Random Forest is not affect by data scaling is that,
The decision of model depends on the outcomes of different decision trees. So due to ensemble of the data, the effect of scaling is minimized as if one decision tree is affected by it, other trees compensate for that effect.

Can we increase the accuracy further?

- We tried to increase the accuracy of the model more by filtering out more data to improve the distribution of the data. But that lead to a decrease in data points and the datapoints for one class started to get more dominant and effectively the accuracy of model decreased.
- Another way to increase the accuracy and try to achieve 100% would be to collect and include more data to the dataset.
- Currently the data points for the Bus class make up nearly 50% of the dataset. We can introduce more data points for Car and Van, this will give more uniformity to dataset and increase the accuracy.

Conclusion

We looked at the different machine learning algorithms for classification and clustering and their pros and cons.

After implementing and experimenting with the models,

- We were able to increase the accuracy of the model by using techniques like data scaling and filtering the dataset to remove the outliers and other anomalies.
- We achieved the highest accuracy for KNN at 94.1%, for random forest 95.95% and we got the highest accuracy with SVM model which is 97.75 percent.

We explored different applications of big data and evaluated the applications and the challenges of implementation of big data in various domains. This helped come up with possible solutions to the issues.

References

(1) (Aktas E, Meng Y)
An Exploration of Big Data Practices in Retail Sector, 2017
https://www.mdpi.com/2305-6290/1/2/12

(2) (Akter, S., Wamba, S.F.)
Big data analytics in E-commerce:a systematic review and agenda for future research,2016
https://link.springer.com/article/10.1007/s12525-016-0219-0#citeas

(3) (Jennifer Huttunen,Jaana Jauhiainen,Jaura Lehit,Annina Nylund,Minna Martikainen,Othmar Lehner)
Big Data,Cloud Computing and Data Science Applications in Finance and Accounting,2019

(4) (Mumtaz Karatas, Levent Eriskin, Muhammet Deveci, Dragan Pamucar, Harish Garg)
Big Data for Healthcare Industry 4.0:Applications, challenges and future perspectives,2022

(5) (Mohd Azeem, Abid Haleem, Shashi Bahl, Mohd Javaid, Rajiv Suman, Devaki Nandan)
Big data applications to take up major challenges across manufacturing industries: A brief review,2022
https://www.sciencedirect.com/science/article/pii/S2214785321012098#s0095

(6) (Liu C, Liu X, Wu F, Xie M, Feng Y, Hu C.)
Using Artificial Intelligence (Watson for Oncology) for Treatment Recommendations Amongst Chinese Patients with Lung Cancer: Feasibility Study,2018
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6231834/

(7) (Ahuja, R., Chug, A., Gupta, S., Ahuja, P., Kohli, S.)
Classification and Clustering Algorithms of Machine Learning with their Applications,2020
https://link.springer.com/chapter/10.1007/978-3-030-28553-1_11

(8) https://medium.com/@akshayjain_757396/advantages-and-disadvantages-of-logistic-regression-in-machine-learning-a6a247e42b20

(9) https://www.geeksforgeeks.org/k-nearest-neighbours/

(10) https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04

(11) https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107

(12) https://www.linkedin.com/advice/3/what-advantages-disadvantages-using-k-means

Appendix

Link of the Repository for the Jupyter notebook with implementation of Algorithms:
https://github.coventry.ac.uk/gardip/Coursework_7141CEM_Big_Data_for-Vehicles/upload/main