CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

Module Title:	*Programming for DA* *Statistics for Data Analytics* *Machine Learning for Data Analysis* *Data Preparation & Visualisation*
Assessment Title:	*MSC_DA_CA1*
Lecturer Name:	*Sam Weiss* *John O'Sullivan* *Muhammad Iqbal* *David McQuaid*
Student Full Name:	*Usukhbayar Tsendgombo*
Student Number:	*2022418*
Assessment Due Date:	*11st November 2022*
Date of Submission:	*11st November 2022*

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Group ID - MSc in Data Analytics

Author: Usukhbayar Tsendgombo
E-mail: 2022418@student.cct.ie
Student ID: 2022418

Abstract

The my assignment scenario is Transport and Infrastructure. A large amount of data has been collected by Dublin City Council (DCC) regarding to the this scenario. I have chosen Maintenance Cleaning Programme for Drains and Gullies DCC.

Gullies collect water and transfer it to the mains sewage network. Over time, a gully can get blocked with debris and grease causing it to become inefficient and eventually can cause flooding and disruption to roadways. In this work, I have tried to explain the leading factor of blocked drains and gullies and predict grease value.

The analysis plan is understand the problems from the data, data preparing, modelling, analysis and solution.

List of Figures

Why using Pyhton?

Python is a General-Purpose object-oriented programming language, which means that it can model real-world entities. It is also dynamically typed because it carries out type-checking at runtime. It does so to make sure that the type of construct matches what we expect it to be. The distinctive feature of Python is that it is an interpreted language. The Python IDLE (Integrated Development Environment) executes instructions one line at a time.

Python is the “most powerful language you can still read”, Says Paul Dubois

Python is one of the richest Programming languages. Going by the TIOBE Index, it is the Second Most Popular Programming Language in the world. This makes a career in Python a great choice.

Python is one of the most commonly used languages, and among its many applications are data mining, AI, web development, embedded systems, and many others. Data analysis and machine learning tools have advanced significantly in recent years thanks to new Python packages. Additionally, there are packages like numpy and pandas that make data comprehension and transformation possible. There is also pyspark, which serves as an API for working with Spark, a framework that makes it easy to work with big data sets. Python is one of the most highly scalable languages and is therefore used by many of the world's largest and most advanced businesses.

Python has become the lingua franca for many data science applications. It combines the power of general-purpose programming languages with the ease of use of domain-specific scripting languages like MATLAB or R. Python has libraries for data loading, visualization, statistics, natural language processing, image processing, and more. This vast toolbox provides data scientists with a large array of general- and special-purpose functionality. One of the main advantages of using Python is the abil- ity to interact directly with the code, using a terminal or other tools like the Jupyter Notebook, which we’ll look at shortly. Machine learning and data analysis are funda- mentally iterative processes, in which the data drives the analysis. It is essential for these processes to have tools that allow quick iteration and easy interaction. As a general-purpose programming language, Python also allows for the creation of complex graphical user interfaces (GUIs) and web services, and for integration into existing systems. (Andreas C. Müller, Sarah Guido, O'Reilly Media, Inc. 2016, Introduction to Machine Learning with Python)

What is EDA? Perform exploratory data analysis

Exploratory Data Analysis (EDA) is understanding the data set by summarizing its main characteristics and often plotting them visually. This step is very important especially when we arrive at modelling the data to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plots and many more. Through the process of EDA, we can also refine the problem statement or definition of our problem.

What is machine learning?

In the early days of "intelligent" applications, many systems used handcoded rules of "if " and "else" decisions to process data or adjust to user input. Think of a spam filter whose job is to move the appropriate incoming email messages to a spam folder. You could make up a blacklist of words that would result in an email being marked as 1 spam. This would be an example of using an expert-designed rule system to design an "intelligent" application. Manually crafting decision rules is feasible for some applica- tions, particularly those in which humans have a good understanding of the process to model. However, using handcoded rules to make decisions has two major disad- vantages:

• The logic required to make a decision is specific to a single domain and task. Changing the task even slightly might require a rewrite of the whole system.

• Designing rules requires a deep understanding of how a decision should be made by a human expert.

(Andreas C. Müller, Sarah Guido, O'Reilly Media, Inc. 2016, Introduction to Machine Learning with Python)

Report plan

The report will be divided into following phases:

- Phase 1: Data understanding
- Phase 2: Data preparation (cleaning, formatting, handling missing value, detect outliers and any other)
- Phase 3: Anaysis of statistics (summarise data, plots, discrete distribution, Normal distribution)
- Phase 4: Machine Learning(Choice of modelling techniques, model building, analysis, present final results)

1. Data understanding

1.1. Data Set

Gullies collect water and transfer it to the main sewage network throughout the city, often seen by the roadside playing an important part of the nations waste water drainage system. If you have a blocked drain gully, it's not likely an issue that will resolve itself and will require inspection and cleaning from a professional drainage contractor.

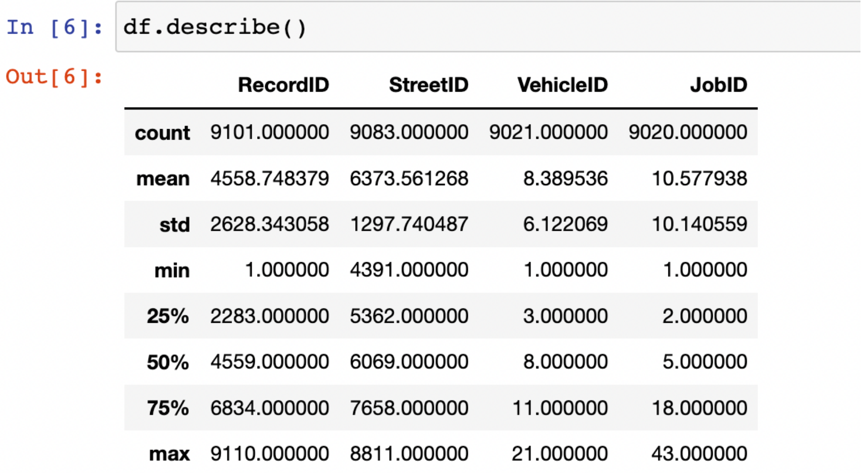
My data Maintenance Cleaning Programme for Drains and Gullies DCC**.** Published by Dublin City Council. And licensed under Creative Commons Attribution 4.0.

DATA.GOV.IE

Register of identified problem drains and status The dataset consists a record of daily vehicle log sheets as they carry out maintenance of problem drains and gullies. Fields include record (RecordID), street (ID), date, worktype (jobID), grease (indicates if grease is present in drain to monitor progress of Fats, Oils and Greases programme) and vehicle type. Explanations for street ID, jobID and vehicle type are given in a separate reference table to be read in conjunction with the main dataset.

1.2. Describe data

The dataset contains 9101 rows and 6 columns. Type of variables are: The greatest number of columns have a numeric data type. The only categorical attribute is column number 5, which depicts the true, false value of Grease. Using method describe() in Python to have a look at the statistics of the data.



```
In [6]: df.describe()

Out[6]:
           RecordID     StreetID    VehicleID        JobID
count   9101.000000  9083.000000  9021.000000  9020.000000
mean    4558.748379  6373.561268     8.389536    10.577938
std     2628.343058  1297.740487     6.122069    10.140559
min        1.000000  4391.000000     1.000000     1.000000
25%     2283.000000  5362.000000     3.000000     2.000000
50%     4559.000000  6069.000000     8.000000     5.000000
75%     6834.000000  7658.000000    11.000000    18.000000
max     9110.000000  8811.000000    21.000000    43.000000
```

[Figure 1] - Data describe

2. Data prepration and Visualation

There are many and various approaches to resolve data quality issues that exist in real-world data. These attempts include, among others, data transformation, in case of different file formats, encodings, attribute names and types, data standardization (or normalization), so that the values conform to the uniformly agreed rules, and data integration, to merge different sources of data. Data preparation is a general term we use to refer to all these different tasks, as most of them are not always easily distinguishable.

A clever selection and application of data preparators on data can transform the latter to a cleaner state.

Preparators

A preparator is a method that transforms a set of input values into a set of output values that are of higher quality or more useful for the use-case at hand. A preparator's complexity can vary from being fairly simple, such as upper-casing all strings, to being quite complex, such as geocoding address fields. The number of input and output attributes can vary and be of any datatype, although in this work we focus only on alphanumeric values.

- Split attribute: Extract parts of an attribute, moving them into other attribute
- Normalize address: Convert address to its commonly accepted form, fixing inconsistencies
- Geocode: Get the geolocation of an address
- Remove special characters: Remove non-alphanumeric characters: [!@#&$*]
- Transliterate: Remove diacritics from words
- Merge attributes: Merge multiple attributes into a single one
- Acronymize Keep the first character of all tokens
- Capitalize characters Convert all characters to upper case
- Syllabify: Word →syllables preparation
- Phonetic encode: Convert value to its pronunciation representation
- Stem: Reduce word to base form

(Ioannis Koumarelas, Lan Jiang, and Felix Naumann. 2020. Data Preparation for Duplicate Detection.)

2.1 Exploratory data analysis

Exploratory Data Analysis (EDA) is a crucial step in any data science project. However, existing Python libraries fall short in support- ing data scientists to complete common EDA tasks for statistical modeling. Their API design is either too low level, which is optimized for plotting rather than EDA, or too high level, which is hard to specify more fine-grained EDA tasks. In response, we propose DataPrep.EDA, a novel task-centric EDA system in Python. Dat- aPrep.EDA allows data scientists to declaratively specify a wide range of EDA tasks in different granularity with a single func- tion call. We identify a number of challenges to implement Dat- aPrep.EDA, and propose effective solutions to improve the scal- ability, usability, customizability of the system. In particular, we discuss some lessons learned from using Dask to build the data processing pipelines for EDA tasks and describe our approaches to accelerate the pipelines. We conduct extensive experiments to compare DataPrep.EDA with Pandas-profiling, the state-of-the-art EDA system in Python. The experiments show that DataPrep.EDA significantly outperforms Pandas-profiling in terms of both speed and user experience.

Exploring the data in this step is about understanding the usage of tables and producing visualizations to have a suitable approach to the storyline of the case and a better understanding of the data.

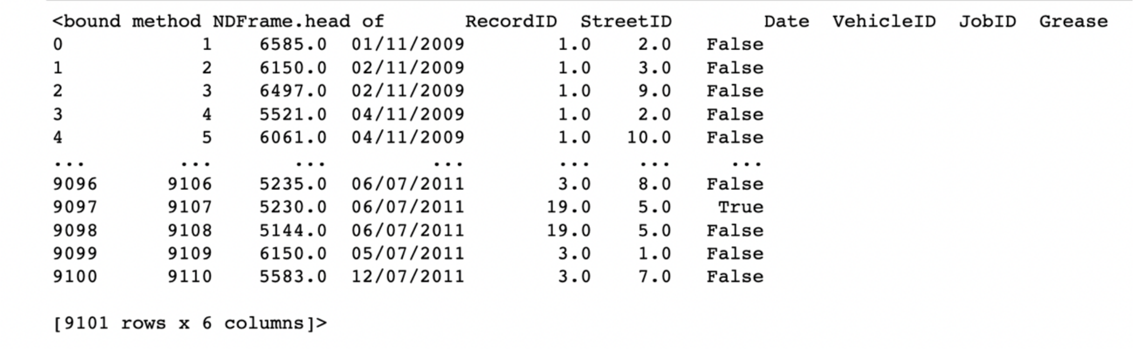
2.1.1. Importing the required libraries for EDA

Below are the libraries that I will use to perform EDA (Exploratory data analysis). Imported required libraries: pandas, numpy, seaborn, matplotlib, spicy, statsmodels. And import warnings.filterwarning for the suppress the warnings.

2.1.2. Loading the data into the data frame

Load the data into the pandas data frame is certainly one of the most important steps in EDA, as we can see that the value from the data set is comma-separated. So I have to do is to just read the CSV into a data frame and pandas data frame does the job for me.

```
        df = pd.read_csv("dccdrainagecleaningprogrammerecordsp20110926-
1405.csv")
```



```
<bound method NDFrame.head of       RecordID  StreetID        Date  VehicleID  JobID  Grease
0            1    6585.0  01/11/2009        1.0    2.0   False
1            2    6150.0  02/11/2009        1.0    3.0   False
2            3    6497.0  02/11/2009        1.0    9.0   False
3            4    5521.0  04/11/2009        1.0    2.0   False
4            5    6061.0  04/11/2009        1.0   10.0   False
...        ...       ...         ...        ...    ...     ...
9096      9106    5235.0  06/07/2011        3.0    8.0   False
9097      9107    5230.0  06/07/2011       19.0    5.0    True
9098      9108    5144.0  06/07/2011       19.0    5.0   False
9099      9109    6150.0  05/07/2011        3.0    1.0   False
9100      9110    5583.0  12/07/2011        3.0    7.0   False

[9101 rows x 6 columns]>
```

[Figure.2] The dataset.

2.1.3. Check the types of data

Check for the datatypes because sometimes features (variables) be stored as a string or an object. In this dataset, I have to convert any strings to integer data so that I could plot the data via a graph. In this case, the data is already in integer format so there's nothing to worry about.

Variable	Description	Data types
RecordID	Record of work	int64
StreetID	Fields include street ID	float64
Date	Date	object
VehicleID	Vehicle type	float64
JobID	Work type	float64
Grease	Indicates if grease is present in drain to monitor progress of Fats, Oils and Greases programme	bool

[Figure.3] Description of the dataset.

2.1.4. Checking duplicates

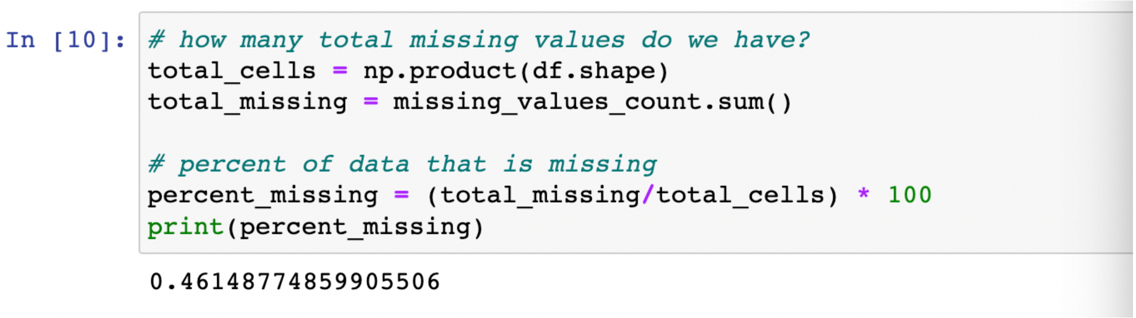
Duplicate observations can usually cause confusion in a data analysis. I need to check the number of duplicates. In this case we did not have an any duplicated rows and columns.

```
df.duplicated().sum()
(0)
```

2.1.5. Drop the missing or null values

Using Pandas and NumPy for handling missing values in dataset. A common occurrence in a data-set is missing values. This can happen due to multiple reasons like unrecorded observations or data corruption.

In this work, I got the number of missing data points per column. And found the total number of missing value and the percent of data that is missing. The missing data precent is 0.46%.



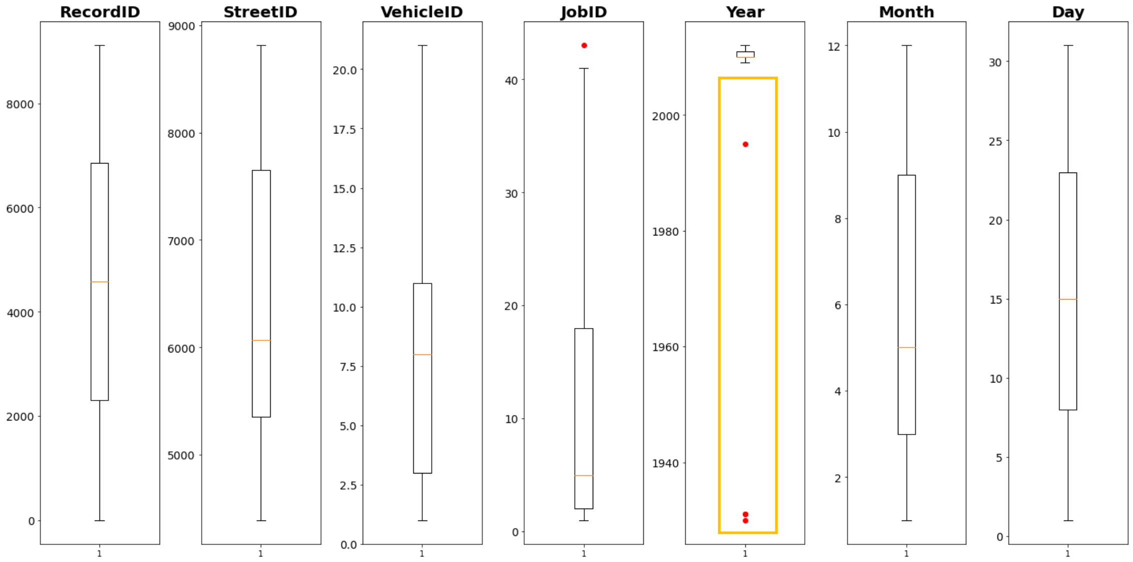
```
In [10]: # how many total missing values do we have?
         total_cells = np.product(df.shape)
         total_missing = missing_values_count.sum()

         # percent of data that is missing
         percent_missing = (total_missing/total_cells) * 100
         print(percent_missing)

         0.46148774859905506
```

[Figure.4] Percent of missing data

2.1.6. Detect Outliers

An outlier is a datapoint or set of datapoints that are vastly different from other data points in the dataset. Sometimes they can be very high or very low.

It’s often a good idea to detect and remove the outliers because outliers are one of the primary reasons for models becoming less accurate. I used the IQR (Interquartile Range) Scoring technique to detect and remove any outliers. Often outliers can be seen with visualizations such as a box plot. In all the plots, I can find some points are that outside the “box”. The Figure.2 shows the value of outlines.



[Figure.5] Boxplot of each variable with outliers

I found the outline value from “Year” variable. Not important outline early years removed (1930, 1931, 1931, 1995).

news1_df=news_df.drop(df.index[[7671,6389,342,6769]])

An outlier is an observation that lies abnormally far away from other values in a dataset. Outliers can be problematic because they can affect the results of an analysis.

3. Statistics

Descriptive Statistics: the analysis of data that helps describe, show or summarise data in a meaningful way. It analyses past events
Probability Theory: deals with predicting the likelihood of future events

Mean: It is the sum of the observation divided by the sample size. It is not a robust statistics as it is affected by extreme values. So, very large or very low value(i.e. Outliers) can distort the answer.

Median: It is the middle value of data. It splits the data in half and also called 50th percentile. It is much less affected by the outliers and skewed data than mean. If the no. of elements in the dataset is odd, the middle most element is the median. If the no. of elements in the dataset is even, the median would be the average of two central elements.

Mode: It is the value that occurs more frequently in a dataset. Therefore a dataset has no mode, if no category is the same and also possible that a dataset has more than one mode. It is the only measure of central tendency that can be used for categorical variables.

Measures of Variability

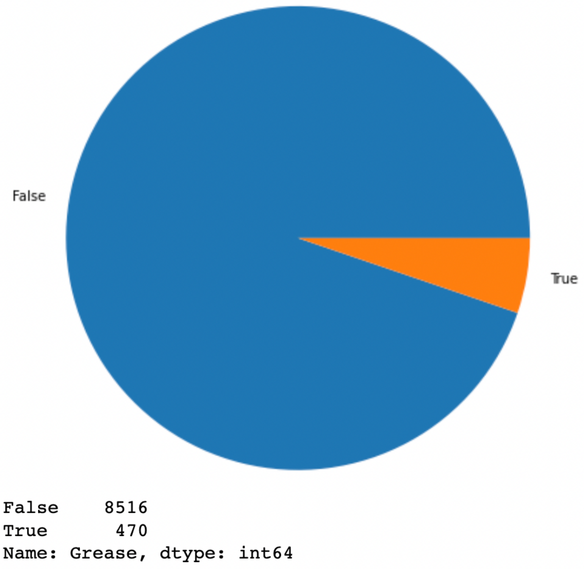
Measures of Variability also known as spread of the data describes how similar or varied are the set of observations. The most popular variability measures are the range, interquartile range (IQR), variance, and standard deviation.

Range: The range describes the difference between the largest and the smallest points in your data. The bigger the range the more spread out is the data.

Standard Deviation: Standard Deviation is used more often because it is in the original unit. It is simply the square root of the variance and because of that, it is returned to the original unit of measurement.

3.1. Stastistical graphics

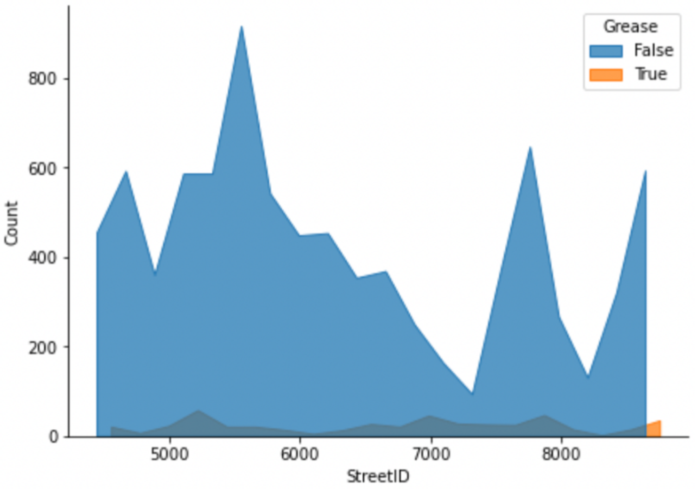
I need to know relative frequencies of the Grease value. Figure.4 shows statistical information of total Grease. I can use a Pie chart because relative frequencies. In this work, the main problem is grease of maintenance cleaning job.



```
False    8516
True      470
Name: Grease, dtype: int64
```

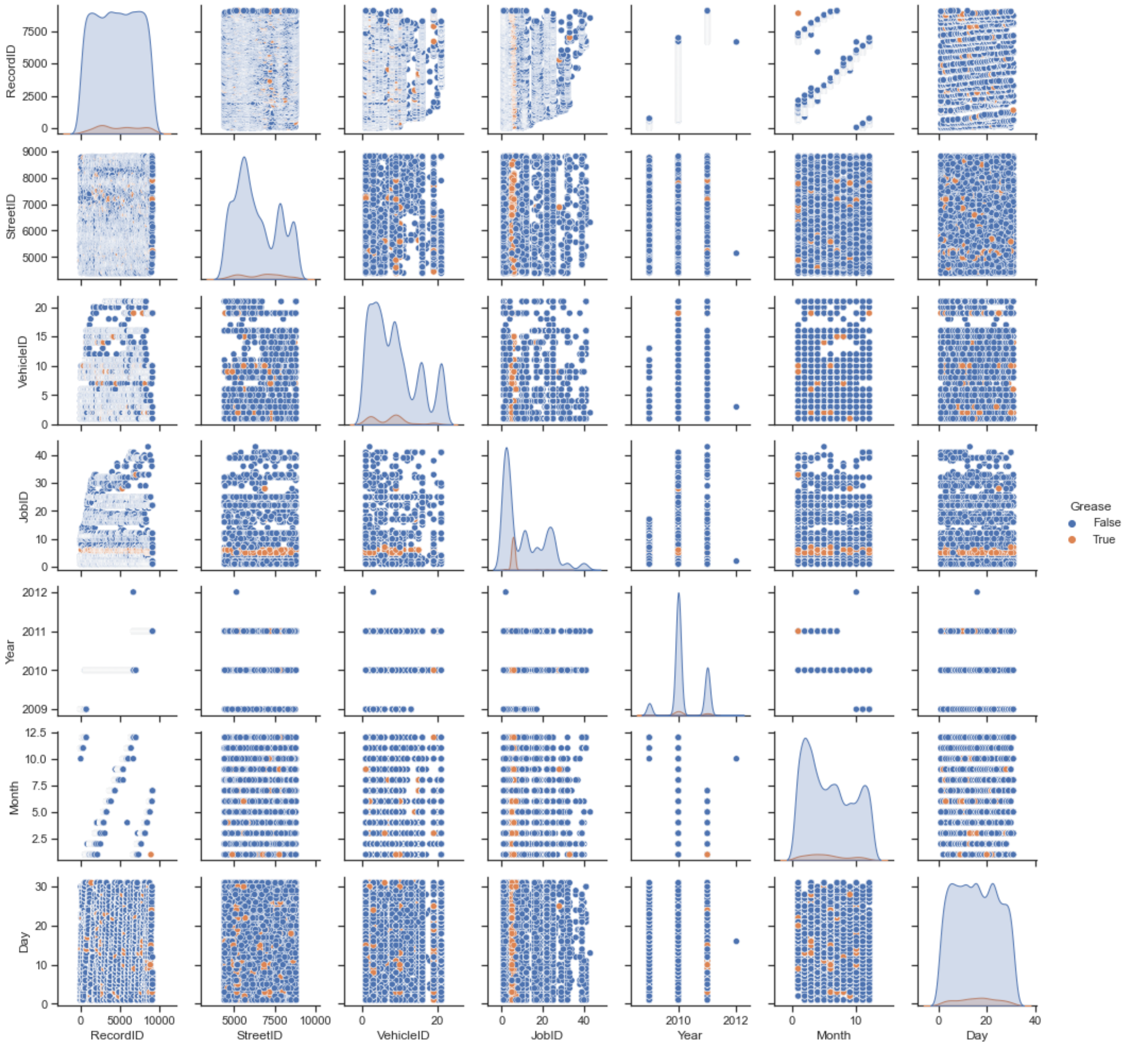
[Figure.6] Statistical information of total Grease.

In my data, I selected 2 important feature which are StreetID and Grease value. I need to know histplot relationships between Grease and StreetID.



[Figure.7] Histplot of StreetID and Grease

I used a scatter plot matrix for the visualize the relationship between pair of variables. Because a scatter plot matrix is a grid (or matrix) of scatter plots used to visualize bivariate relationships between combinations of variables. Each scatter plot in the matrix visualizes the relationship between a pair of variables, allowing many relationships to be explored in one chart.



[Figure.8] The scatter plot matrix

Figure 8, shows importance features of Grease value. At this time, we can see that JobID and SteetID are the essential features in this dataset.

Propapility

3.2. Poisson distribution

The Poisson process is a simple kind of random process, which models the occurrence of random points in time or space. There are numerous ways in which processes of random points arise: some examples are presented in the first section. The Poisson process describes in a certain sense the most random way to distribute points in time or space. This is made more precise with the notions of homogeneity and independence.
Definition. A discrete random variable X has a Poisson distribution with parameter μ, where μ > 0 if its probability mass function p is given by

p(k) = P(X = k) = μk k! e−μ for k = 0, 1, 2,....

We denote this distribution by Pois (μ).

We denote this distribution by Ber (p). Note that we wrote pX instead of p for the probability mass function of X. This was done to emphasize its dependence on X and to avoid possible confusion with the parameter p of the Bernoulli distribution.

Definition. A discrete random variable X has a binomial distribution with parameters n and p, where n = 1, 2,... and 0 ≤ p ≤ 1, if its probability mass function is given by

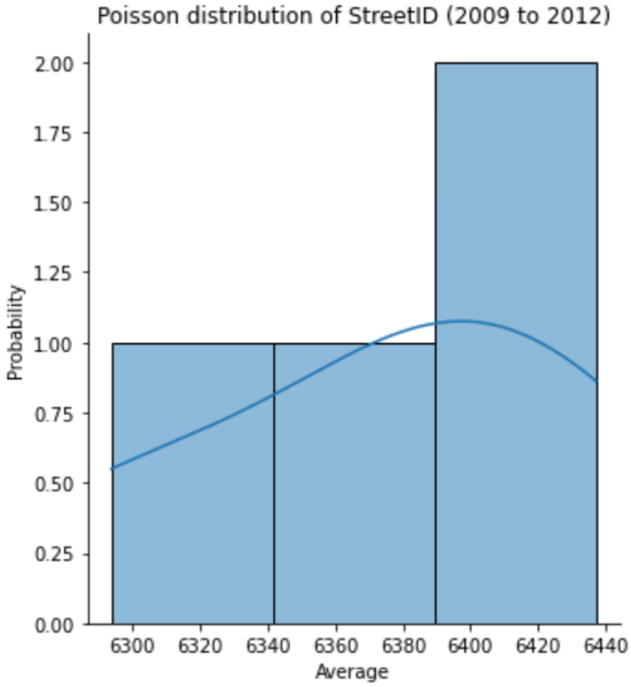
pX(k) = P(X = k) = n k pk (1 − p) n−k for k = 0, 1,...,n.

We denote this distribution by Bin(n, p).

(F.M. Dekking, C. Kraaikamp, H.P. Lopuhaa, L.E. Meester. 2005, A Modern Introduction to Probability and Statistics, Springer Texts in Statistics)

In my data, have no need to analyze by distribution. The Record number is just numbering of maintenanace, StreetID is name of drain locations, JobID is type of the work, VehicleID is type of the vehicle. But, I checked Poison distribution and Normal distribution on my data(StreetID).

I calculated Poisson distribution of StreetID (2009 to 2012). lam=6370.6



[Figure.9] Poisson distribution of StreedID

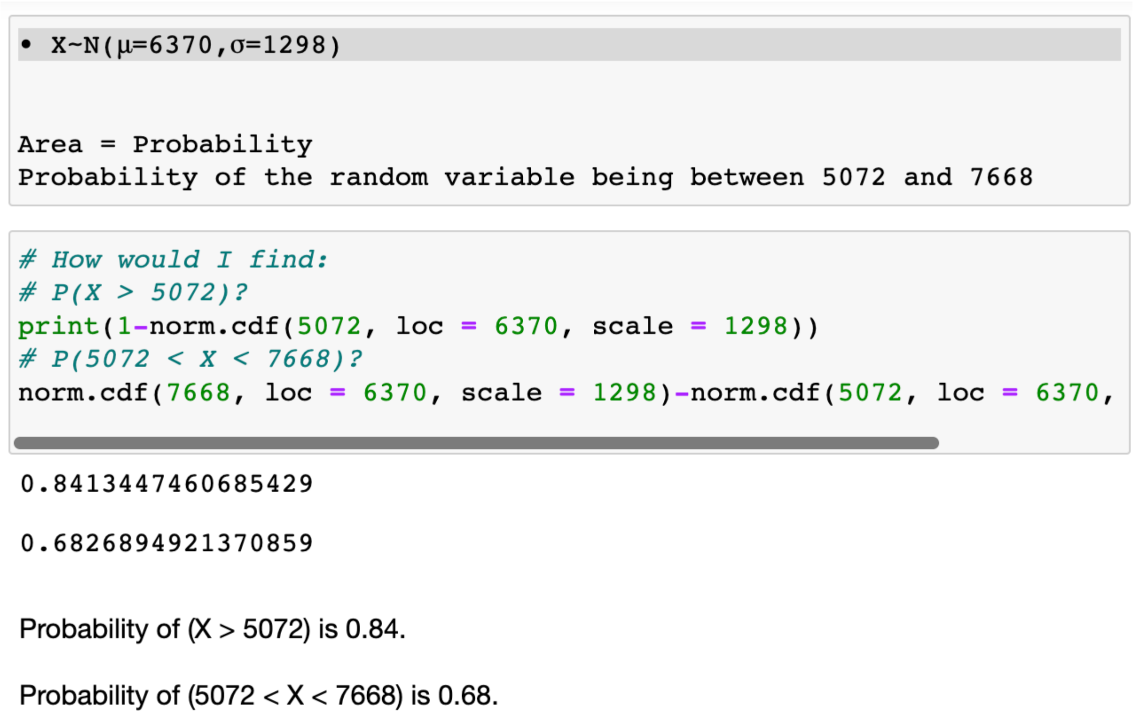
3.3. Normal distribution

The continuous random variable X follows a Normal distribution (or Gaussian distribution) if the probability density function (PDF) is:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{(2\sigma^2)}}$$

It basically describes how large samples of data look like when they are plotted. It is sometimes called the “bell curve“ or the “Gaussian curve“. In a perfect normal distribution, each side is an exact mirror of the other. It should look like the distribution on the picture below:

Calculated single probablities using the Normal distribution. I found probability of (X > 5072) and (5072 < X < 7668).



```
• X~N(μ=6370,σ=1298)


Area = Probability
Probability of the random variable being between 5072 and 7668
```

```
# How would I find:
# P(X > 5072)?
print(1-norm.cdf(5072, loc = 6370, scale = 1298))
# P(5072 < X < 7668)?
norm.cdf(7668, loc = 6370, scale = 1298)-norm.cdf(5072, loc = 6370,
```

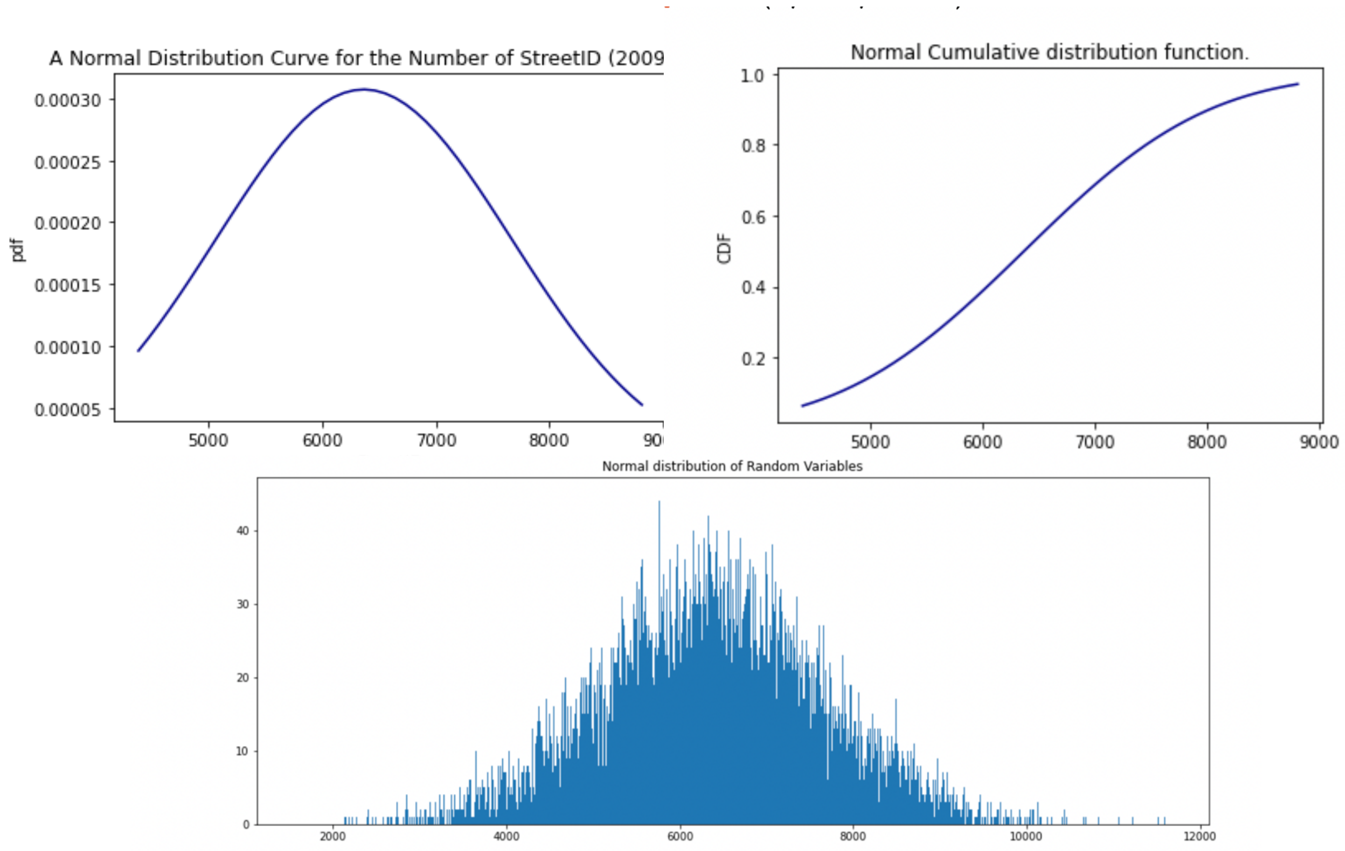
```
0.8413447460685429

0.6826894921370859
```

Probability of (X > 5072) is 0.84.

Probability of (5072 < X < 7668) is 0.68.

[Figure.10] Probability of the area

Plots of RVS, PDF and CDF



[Figure.11] Plots of RVS, PDF and CDF

4. Machine learning

Machine learning is the intersection between theoretically sound computer science and practically noisy data. Essentially, it's about machines making sense out of data in much the same way that humans do. (Thoughtful Machine Learning with Python, Matthew Kirk)

The modelling techniques that better adapt to the problem are related to the supervised ML category.

Types of machine learning: Supervised Learning, Unsupervised Learning, Reinforcement Learning.

Types of Supervised Learning: Classification, Regression. I have chosen the Classification method. Because the Classification method is a kind of problem wherein the outputs are categorical. In my data, outputs "True" or "False" of the Grease.

Solutions under Classification: K-Nearest Neighbor (KNN), Naïve Bayes, Decision Tree, Random Forest.

Decision Tree: Simple to understand, interpret and visualize, little effort required for data preparation, can handle both numerical and categorical data, non linear parameters do not effect its performance.

Random Forest: no overfitting, use multiple trees reduce the risk of overfitting, runs efficiently on large database and training time is less.

Supervised - Classification – Decision tree, Random forest

Supervised:

Supervised learning is used whenever we want to predict a certain outcome from a given input, and we have examples of input/output pairs. We build a machine learning model from these input/output pairs, which comprise our training set. Our goal is to make accurate predictions for new, never-before-seen data. Super- vised learning often requires human effort to build the training set, but afterward automates and often speeds up an otherwise laborious or infeasible task.

There are two major types of supervised machine learning problems, called classification and regression.

Classification:

Classification is the task of learning a target function f that maps each attribute set x to one of the. Predefined class labels y.

The target function is also known informally as a classification model. Classification model is useful for the following purposes.

A classification technique (of classifier) is a systematic approach to building classification models from an input data set. Examples include decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naïve ayes classifiers. Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seem before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.

The training test set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix.

(Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 2006, Introduction to Data Mining, 1st Edition, Chapter 4)

4.1. Decision Tree:

To illustrate how classification with a decision tree works, consider a simple version of the vertebrate classification problem. Instead of classifying the vertebrates into five distinct groups of species

Decision trees are widely used models for classification and regression tasks. Essentially, they learn a hierarchy of if/else questions, leading to a decision.

(Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 2006, Introduction to Data Mining, 1st Edition, Chapter 4)

In my data, I used only “StreetID” and “Year” features into “x” and store the “Grease” feature into “y”.

Splited the dataset into the Training set and Test set. After that, Trained the Decision Tree classification model on the Training set.

Loaded the library sklearn.tree, greated and initialized a DecisiontreeClassifier and trained the classifier by calling a method “fit()”.

```
classifier = DecisionTreeClassifier(max_depth = 4, random_state = 0)

classifier.fit(X_train, y_train)
```

Predict the Test set result

Calculated the predicted value by calling a method predict.

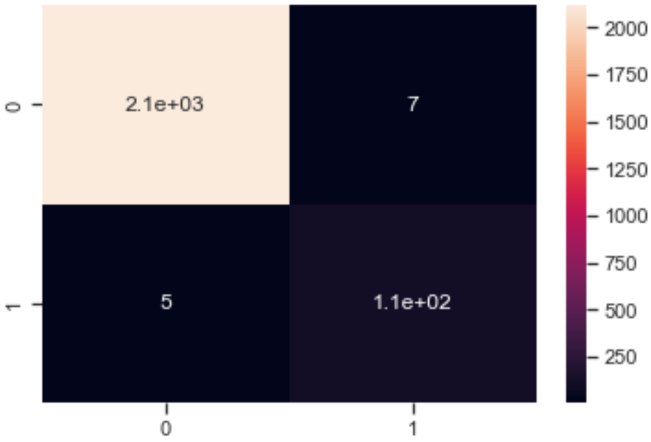
```
y_pred = classifier.predict(X_test)
```

Make the Confusion Matrix

Calculate cm by calling a method named as 'confusion_matrix'. Then called a method heatmap to plot confusion matrix.

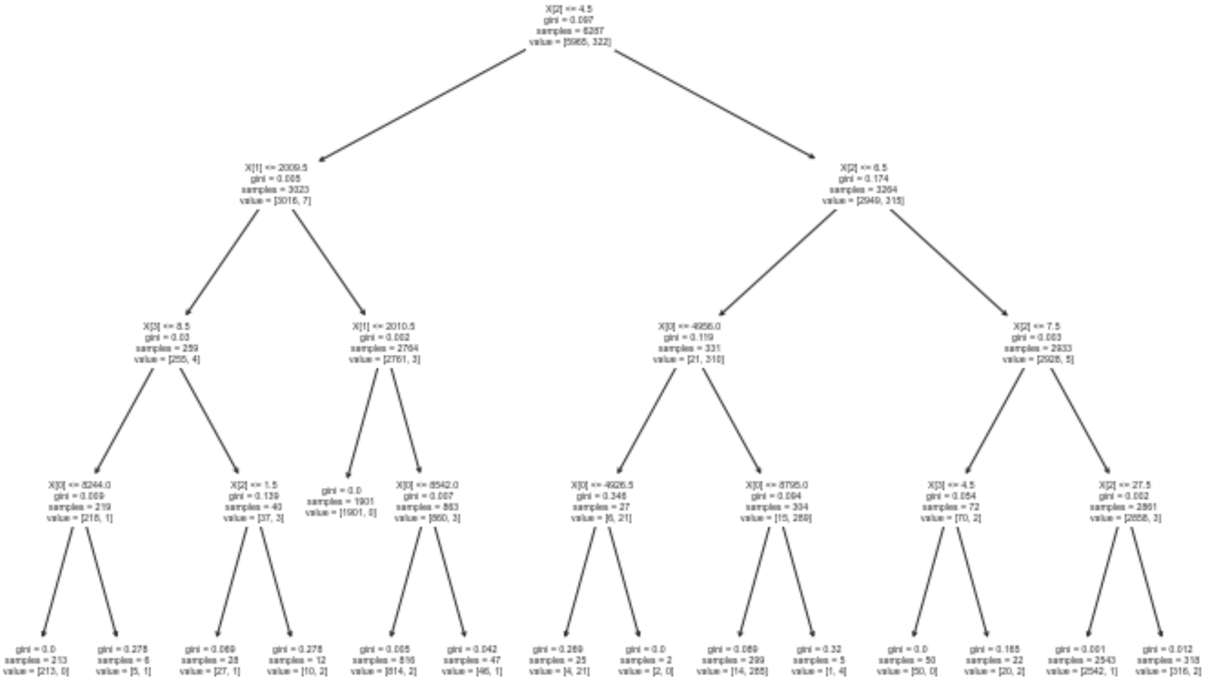
Function of print the classification_report based on y_test and y_predict

```
print(classification_report(y_test, y_pred))
```



[Figure.12] Heatmap to plot confusion matrix

Display the tree by calling a method plot_tree:



[Figure.13] Displayed the Tree

Implementation

Finally, I have created a model that uses decision tree algorithm to predict whether a drain will blocked by grease or not. The Accuracy of the model is 99.4%. The Government or maintenance companies can use this model to decide whether it should use grease removing instruments or not.

4.2. Random forests:

As we just observed, a main drawback of decision trees is that they tend to overfit the training data. Random forests are one way to address this problem. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely overfit on part of the data. If we build many trees, all of which work well and overfit in different ways, we can reduce the amount of overfitting by averaging their results. This reduction in overfitting, while retaining the predictive power of the trees, can be shown using rigorous mathematics.

To implement this strategy, we need to build many decision trees. Each tree should do an acceptable job of predicting the target, and should also be different from the other trees. Random forests get their name from injecting randomness into the tree build- ing to ensure each tree is different. There are two ways in which the trees in a random forest are randomized: by selecting the data points used to build a tree and by select- ing the features in each split test.

In my data, I Imported Random Forest Model.

```
from sklearn.ensemble import RandomForestClassifier
```

Created a Gaussian Classifier by

```
clf = RandomForestClassifier(n_estimators = 100)
```

Trained the model using the training sets y_pred=clf.predict(X_test)

```
clf.fit(X_train,y_train)
```

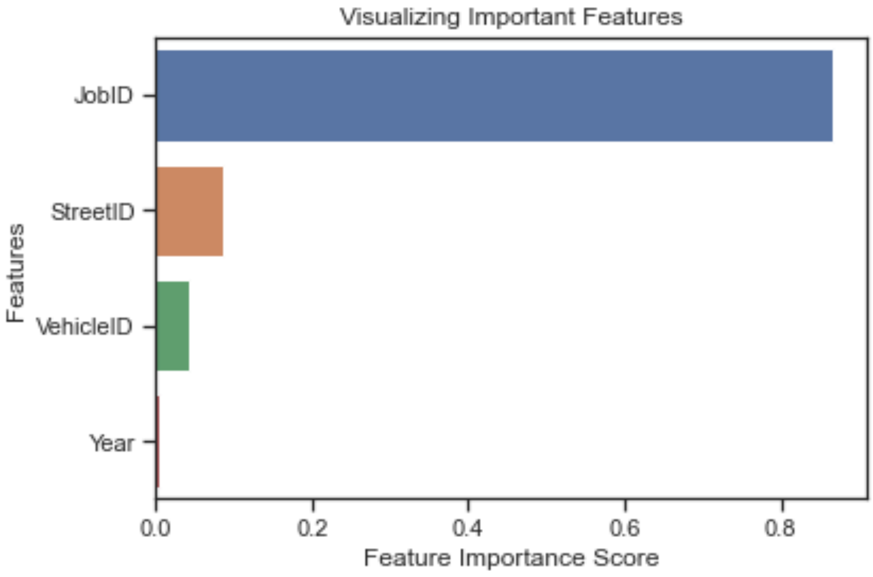
Maked the Confusion Matrix same as like the Decision tree.

Find Important Features in Scikit-learn
I found important features or selecting features in the Drain dataset. In scikit-learn, I can perform this task in the following steps:

- Create a random forests model.
  ```
  RandomForestClassifier()
  ```

- Use the feature importance variable to see feature importance scores.
  ```
  JobID        0.866630
  StreetID     0.085413
  VehicleID    0.042676
  Year         0.005282
  dtype: float64
  ```

- Visualize these scores using the seaborn library.



[Figure.14] Visualization of importance features

Implementation

So, I have created a model that uses Random Forest algorithm to predict whether a drain will blocked by grease or not. The Accuracy of the model is 99.5%. The Government or maintenance companies can use this model to decide whether it should use grease remover instruments or not and type of vehicle or type of work. Also, JobID is the essential features in this dataset.

Conclusion

The use of CRISP-DM methodology on Maintenance Cleaning Programme for Drains and Gullies's data made possible to achieve the main object proposed at the beginning. The predictions of the Grease's True/False value have been predicted by two different Machine Learning Models. All machine learning models accuracy is 99%.

This data was easy to understand, prepare and no need to normalize. Also, it was not need to analyze distribution.

Note: I had no basic knowledge of python coding, but I learned a lot while doing this work.

References

Ioannis Koumarelas, Lan Jiang, and Felix Naumann. 2020. Data Preparation for Duplicate Detection.

Jinglin Peng, Weiyuan Wu, Brandon Lockhart, Song Bian, Jing Nathan Yan, Linghao Xu, Zhixuan Chi, Jeffrey M. Rzeszotarski, Jiannan Wang. 2021, *DataPrep.EDA: Task-Centric Exploratory Data Analysis for Statistical Modeling in Python*

John A. Rice. 2007, *Mathematical Statistics and Data Analysis*, 3rd Edition, University of California, Berkeley

Matthew Kirk. 2017, Thoughtful Machine Learning with Python, A Test-Driven Approach

Jiawei Han, Micheline Kamber, Jian Pei. 2012, *Data mining*, 3rd Edition, Addison-Wesley.

Zhenyun Deng, Xiaoshu Zhu, Debo Cheng, Ming Zong, Shichao Zhang. 2015, Efficient kNN classification algorithm for big data

Andreas C. Müller, Sarah Guido, O'Reilly Media, Inc. 2016, Introduction to Machine Learning with Python

Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 2006, Introduction to Data Mining, 1st Edition, Chapter 4

Charles Severance. 2009, *Pyhton for Everybody, Exploring data using python 3*

Scharl, Julie. 2017, Atd-1 Avionics Phase 2: Post-Flight Data Analysis Report, Available at https://ntrs.nasa.gov/citations/20170007243

F.M. Dekking, C. Kraaikamp, H.P. Lopuhaa, L.E. Meester. 2005, A Modern Introduction to Probability and Statistics, Springer Texts in Statistics

https://www.programiz.com/python-programming

Harvard Style *UseIt,* Available at http://www.library.uq.edu.au/training/citation/harvard.html