Big Data

CMP3749M

19701707

Douglas Carrie

Task 1 – narrative

Understanding customers behaviour better at American Express

Small intro maybe

Methods

One analysis method is Descriptive analysis, it involves trying to present some kind of discovery from data that has already been collected. American Express use this method frequently to analyse all different types of data to draw conclusions. For example, finding how many new users have joined this year or what is the average spending per customer per day for different countries. These descriptive answers will be used to gain a real analysis over the situation being examined.

Another analysis method is Diagnostic analysis. Its used to identify an underlying pattern in data. American Express uses this approach to detect fraud within transactions. They also use data fusion, a method of combining multiple different data sources together to give a wider picture of a pattern which may not have been visible from just a single source. American Express uses data like card membership information, spending details, and merchant information to stop transactions from going through which most probable are fraudulent. On average AmEx has stopped 2 billion dollars' worth of potential fraud annually.

One analysis method is Predictive analysis, it involves trying to predict some kind of information or future outcome from historic. American Express uses vast amounts of historical data about its customers, especially transaction history and usage frequency, to determine which of its accounts will close. They expect 24% of all Australian accounts to close due to inactivity within 4 months. With this discovery, based on customer data, American express can spend more effort on its Australian market and take measures to keep existing customers.

Prescriptive analysis is used in Big Data to build on predictions to then advice actions. This can be extremely useful especially for American Express. They use this analysis method for other businesses which use Amex's service for transaction payments. Their service called American Express Advance use Big Data collected from customer transactions and other similar businesses transactions to then provide the business using the service to gain useful insights like purchasing patterns and potential customers. American Express have a great advantage over other banks and business merchants due to it's a closed loop system allowing them to access and predict far greater data.

Advantages and Disadvantages to each method

Descriptive:

<u>Advantages:</u>

- It takes into account of both types of data, quantitative and qualitative.
- It is quick to carry out descriptive investigations, meaning costs are not high and time is low compared with other investigations.

<u>Disadvantages:</u>

- Questions must be well formed, if not the answers may not be completely reliable, this could make it difficult to carry out a credible investigation.
- Data collected for descriptive analysis is collected at random which makes it impossible to obtain valid data that would represent the entire population.

Diagnostic:

Advantages:

- Can turn complex data into easily understandable information, makes it easier for all stakeholder to visualise and gain insights
- Can uncover useful information which may not have been discovered without searching for patterns in data

Disadvantages:

- Analysis focusses on past occurrences which could falsely predict future ones
- Analysis method may not be enough for certain businesses, may require more advanced solution (predictive)

Predictive:

Advantages:

- Can turn complex data into easily understandable information, makes it easier for all stakeholder to visualise and gain insights
- Can uncover useful information which may not have been discovered without searching for patterns in data

Disadvantages:

- Analysis focusses on past occurrences which could falsely predict future ones

3. Critical reflection including fundamental limitations for methods I mention

include : table to summarise your discussion.

https://www.sciencepubco.com/index.php/ijet/article/view/18451#:~:text=Google's%20streaming%20services%2C%20YouTube%2C%20are,system%20that%20offers%20consistent%20memory.

https://www.getsmarter.com/blog/career-advice/big-data-analysis-techniques/

https://digital.hbs.edu/platform-digit/submission/american-express-using-data-analytics-to-redefine-traditional-banking/

https://digital.hbs.edu/platform-digit/submission/amex-providing-data-driven-insights-for-businesses/

https://englopedia.com/advantages-and-disadvantages-of-descriptive-research/

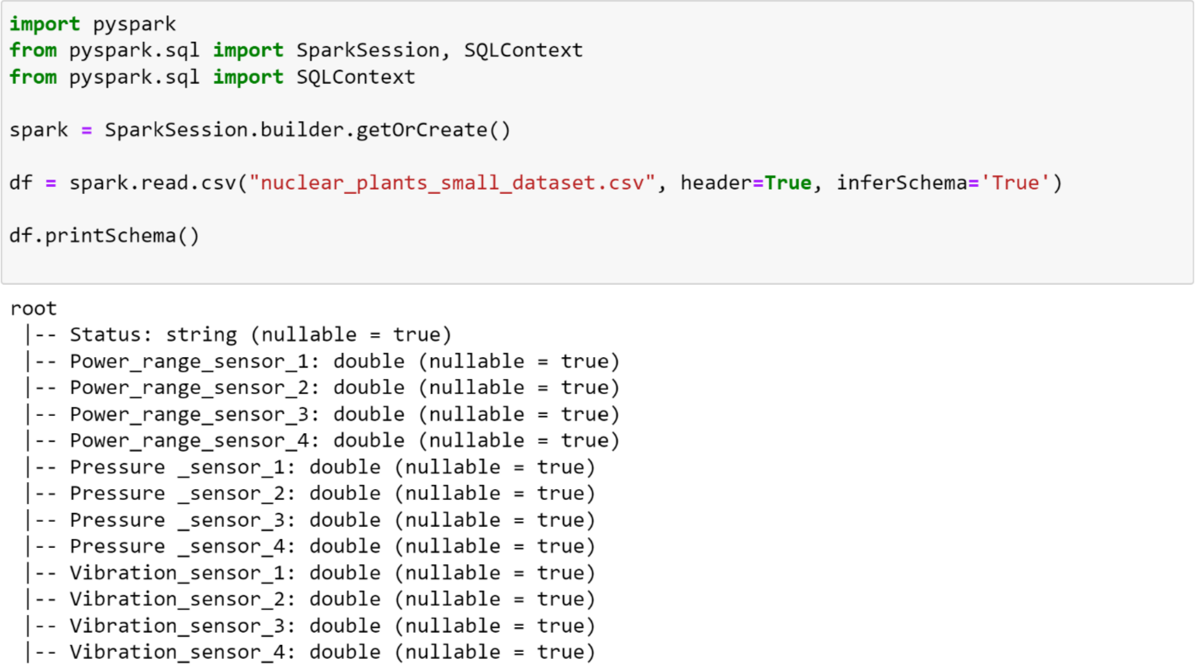
https://whatagraph.com/blog/articles/diagnostic-analytics

Task 2 – Analysis (2500 words max strict) (90%)

Loading the data:

The smaller dataset 'nuclear_plants_small_dataset.csv' contains just under 1000 columns. This needs to be loaded into a pyspark dataframe, so that the data can be further examined. (Fig.1)

Fig.1



```
import pyspark
from pyspark.sql import SparkSession, SQLContext
from pyspark.sql import SQLContext

spark = SparkSession.builder.getOrCreate()

df = spark.read.csv("nuclear_plants_small_dataset.csv", header=True, inferSchema='True')

df.printSchema()
```

```
root
 |-- Status: string (nullable = true)
 |-- Power_range_sensor_1: double (nullable = true)
 |-- Power_range_sensor_2: double (nullable = true)
 |-- Power_range_sensor_3: double (nullable = true)
 |-- Power_range_sensor_4: double (nullable = true)
 |-- Pressure _sensor_1: double (nullable = true)
 |-- Pressure _sensor_2: double (nullable = true)
 |-- Pressure _sensor_3: double (nullable = true)
 |-- Pressure _sensor_4: double (nullable = true)
 |-- Vibration_sensor_1: double (nullable = true)
 |-- Vibration_sensor_2: double (nullable = true)
 |-- Vibration_sensor_3: double (nullable = true)
 |-- Vibration_sensor_4: double (nullable = true)
```

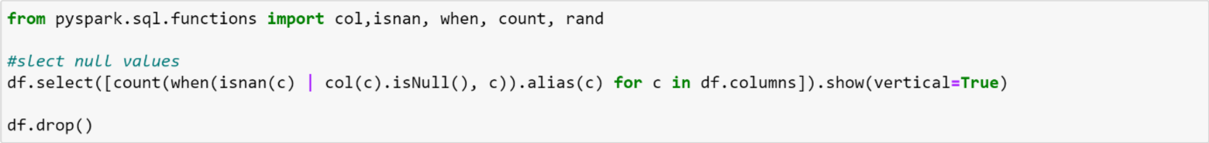
The 'nuclear_plants_small_dataset.csv' can now be targeted in python through the 'df' variable.

Task 1

The data must be in the appropriate form to analyse. All columns shouldn't contain any missing values and if they do we would want to remove the entire column as long as the data is missing at random. If the data was missing not at random then we would need to look into the reason why there are empty fields within the dataset, if this was the case then the column with missing data cannot simply be removed as it may result in a bias in the results.

In Pyspark, we can query 'df' dataframe to find all columns to find which rows contain no value at all, null. Also, we can query for any value which is not a number, NAN, this could be useful in finding values that should have been a number but are not.
Below shows how this is done in Pyspark (fig.2)

fig.2



```
from pyspark.sql.functions import col,isnan, when, count, rand

#slect null values
df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df.columns]).show(vertical=True)

df.drop()
```

The only row returned by this query is row 0 because it contains the header for the csv. The header contains words which fit the category, not a number (NAN). This means all numerical values for the sensors does not contain any missing values.

Task 2

Summary statistics of each feature for both groups

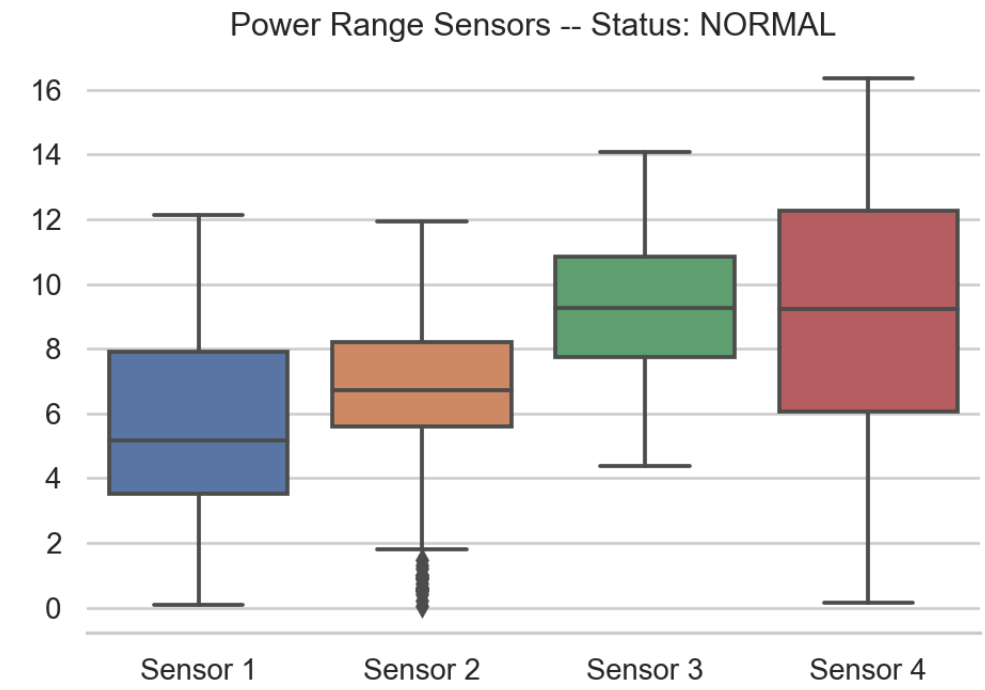
Status	Sensor	Minimum	Maximum	Mean	Median	Mode	Variance
Normal	Power range 1	0.0851	9.9463	5.602452811244976	5.178649		8.374354492436746
Abnormal		0.0082	9.649302	4.396694975903612	4.51355		6.201490118793131
Normal	Power range 2	0.0403	9.9298	6.844503413654616	6.71765		4.880531200853515
Abnormal		0.3891	9.9188	5.914042	5.932218		5.392428867
Normal	Power range 3	10.0012	9.9966	9.292054016064245	9.26285		4.173688700195111
Abnormal		10.019154	9.9994	9.164170212851408	9.47205		8.654818018354874
Normal	Power range 4	0.1547	9.9957	8.701398192771098	9.24085		20.053993554020884
Abnormal		0.0623	9.953772	6.00914597991968	5.3993		14.280946623673312

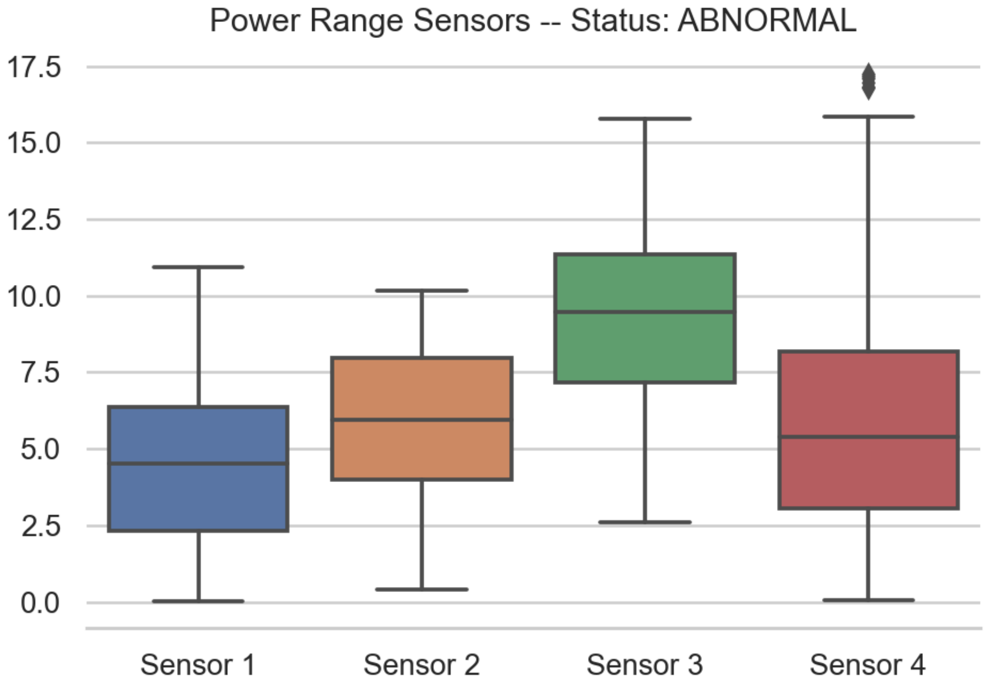
Status	Sensor	Minimum	Maximum	Mean	Median	Mode	Variance
Normal	Pressure 1	0.0248	56.8562	13.797525502008051	10.6348		138.37257347272865
Abnormal		0.131478	67.9794	14.600728132530124	12.596150000000002		134.42562444441973
Normal	Pressure 2	0.0104	9.2212	3.4156463855421695	3.113		4.806673026314872
Abnormal		0.008262	10.242738	2.7402695381526083	2.382689		4.014421746820416
Normal	Pressure 3	0.0774	12.6475	5.923352610441759	5.7394		6.499049116421752
Abnormal		0.001224	11.7724	5.5751150803212814	5.744256999999999		6.215756717729965
Normal	Pressure 4	0.0058	15.1085	5.586180120481923	4.25915		18.37268188992595
Abnormal		0.029478	16.55562	4.40782413253012	3.322575		15.669184817711217

Status	Sensor	Minimum	Maximum	Mean	Median	Mode	Variance
Normal	Vibration 1	0.0092	31.4981	8.441436947791166	7.4498999999999995		38.52794801577502
Abnormal		0.0	36.186438	7.887688803212859	6.53595		37.61341588622746
Normal	Vibration 1	0.0277	34.8676	9.699615863453822	8.70075		49.060657006045574
Abnormal		0.0185	34.331466	10.303569907630527	8.973099999999999		58.50550622001145
Normal	Vibration 1	0.0646	53.2384	19.437804417670666	16.4645		193.13624276565452
Abnormal		0.131784	36.911454	10.93815894779115	8.9872690		66.679536241711

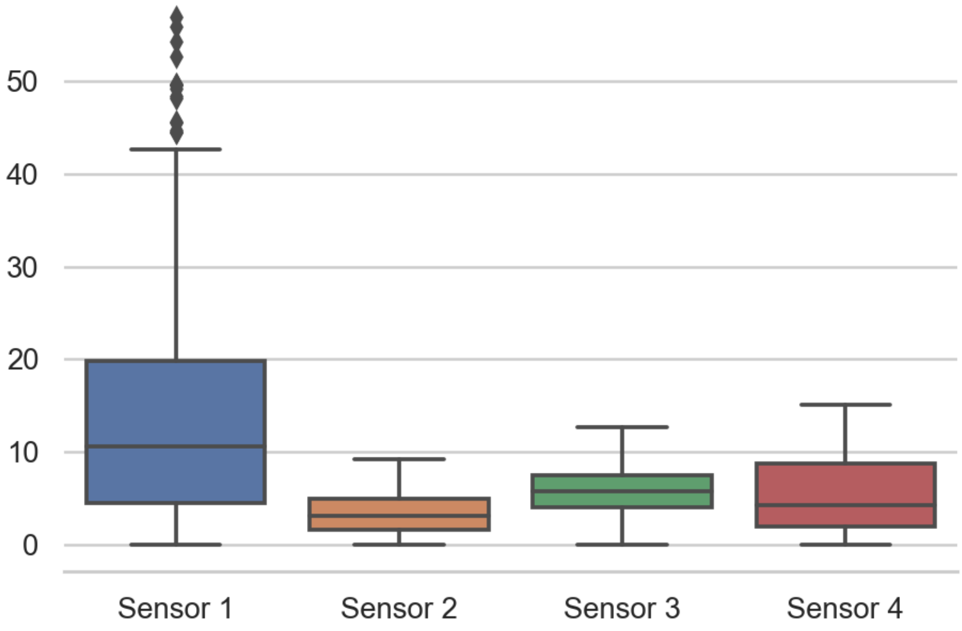
					00000001		
Normal	Vibration 1	0.0831	43.2314	10.925097590361439	9.48545		67.3093714686461
Abnormal		0.0092	26.4669	8.9420846746988	8.137599999999999		36.89342263539999

Boxplots of each feature for each group

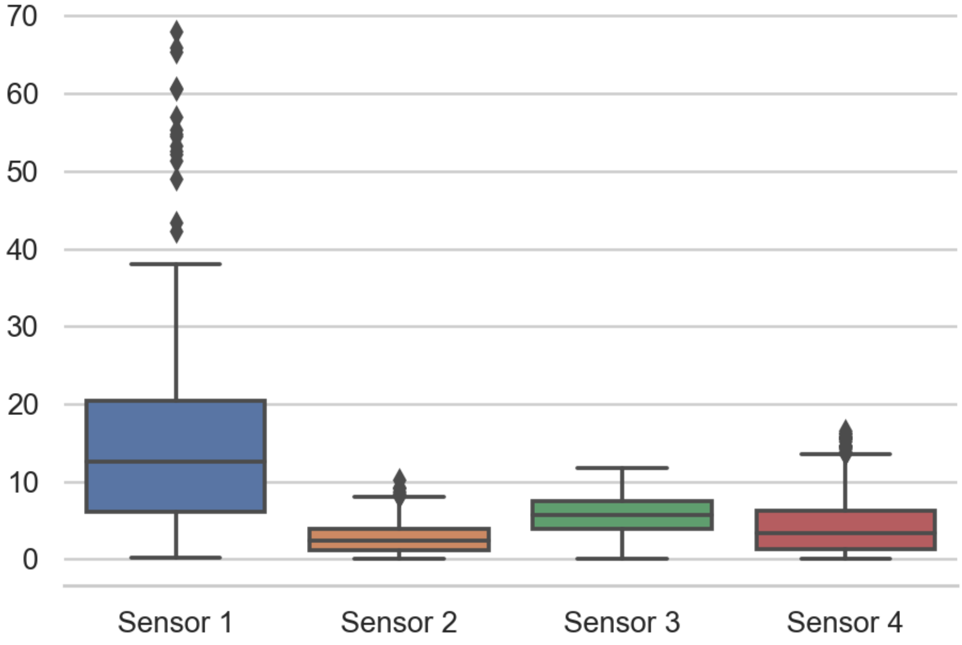


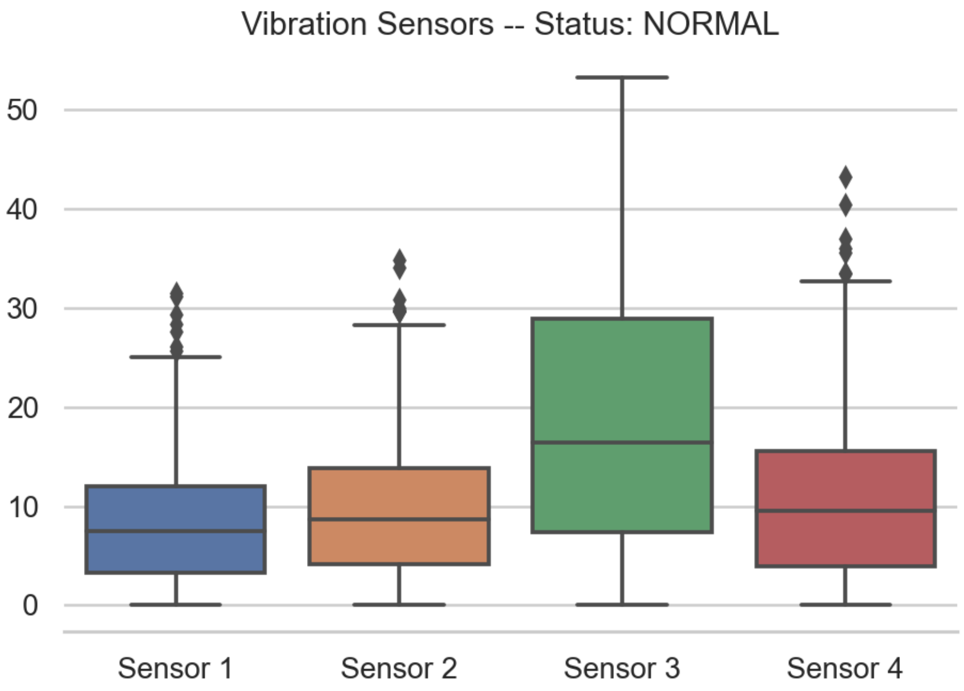


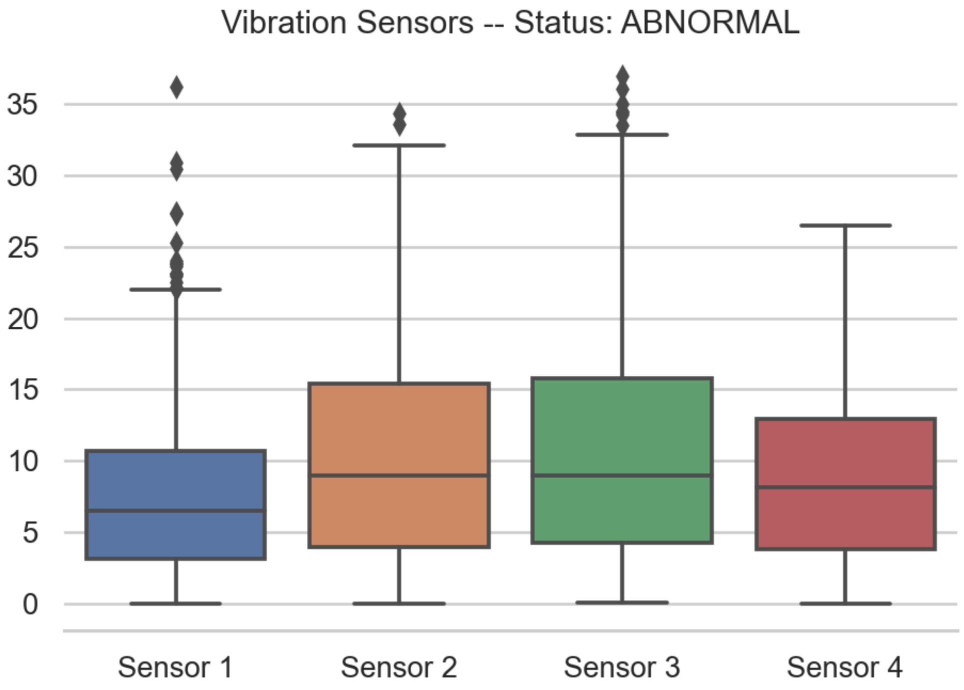
Pressure Sensors -- Status: NORMAL



Pressure Sensors -- Status: ABNORMAL

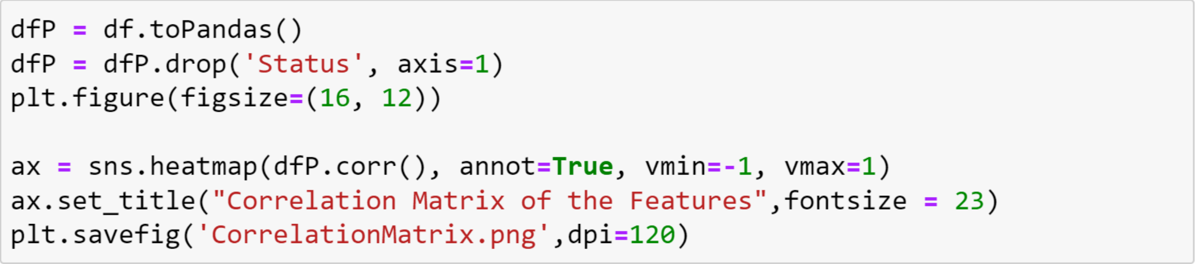



Vibration Sensors -- Status: NORMAL
50
40
30
20
10
0
Sensor 1
Sensor 2
Sensor 3
Sensor 4

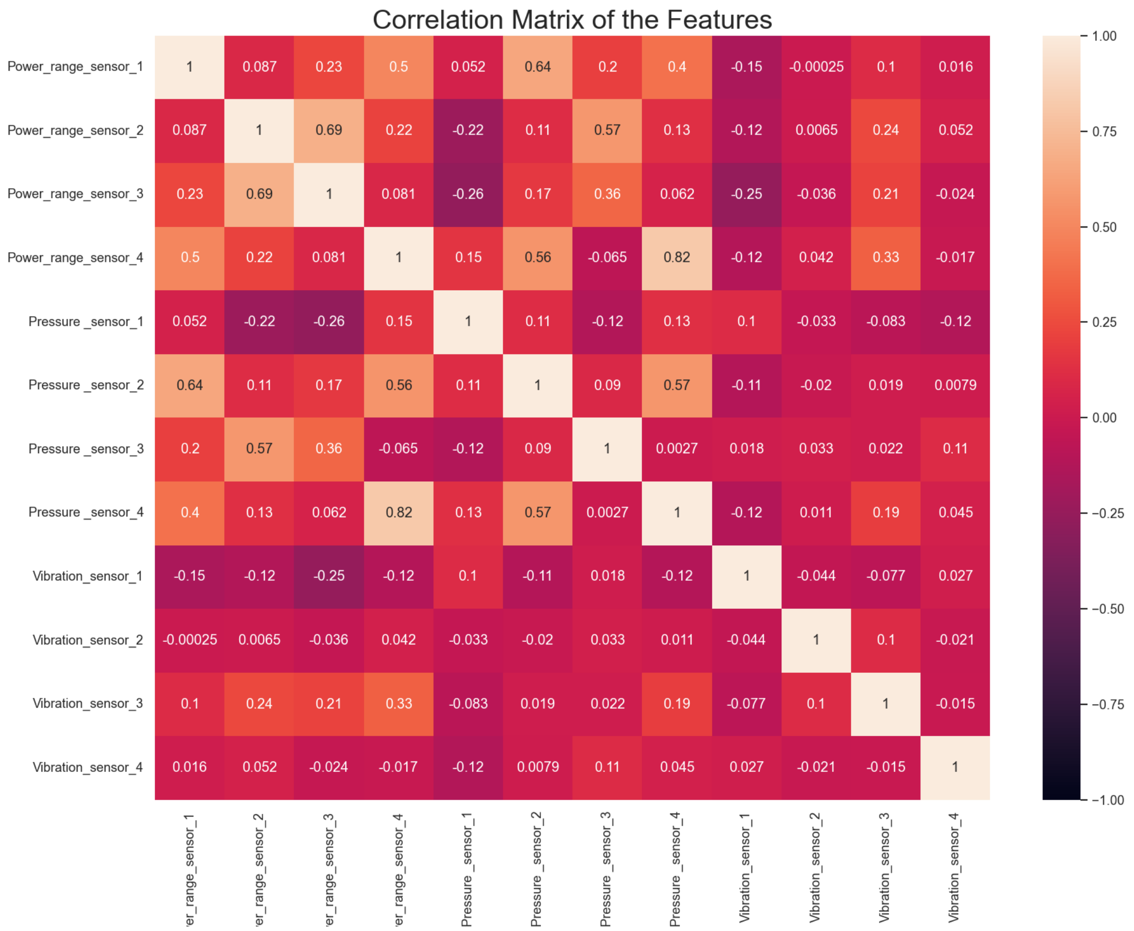

Vibration Sensors -- Status: ABNORMAL
35
30
25
20
15
10
5
0
Sensor 1
Sensor 2
Sensor 3
Sensor 4

Task 3

Correlation matrix:



```
dfP = df.toPandas()
dfP = dfP.drop('Status', axis=1)
plt.figure(figsize=(16, 12))

ax = sns.heatmap(dfP.corr(), annot=True, vmin=-1, vmax=1)
ax.set_title("Correlation Matrix of the Features",fontsize = 23)
plt.savefig('CorrelationMatrix.png',dpi=120)
```



Discuss your observations on the correlation matrix. Are there any features which are highly correlated? In any case, we will use all the features in the following tasks.

Power_range_sensor_4 and Pressure_sensor_4

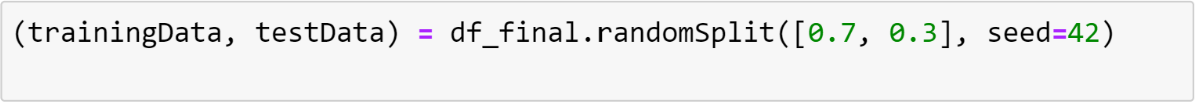
Power_range_sensor_2 and Power_range_sensor_3

Pressure_sensor_4 and Power_range_sensor_4

Task 4

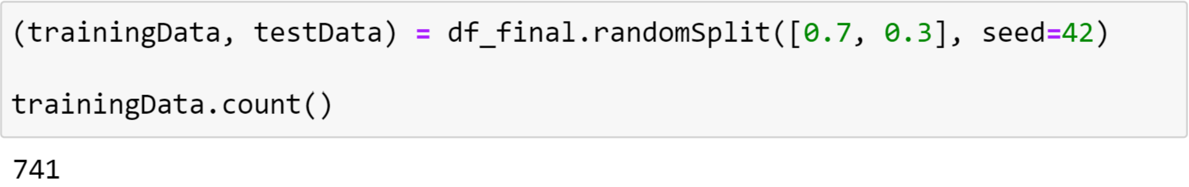
Splitting the data into test and train sets.

The data is shuffled using the randomSplit method from PySpark.



```
(trainingData, testData) = df_final.randomSplit([0.7, 0.3], seed=42)
```

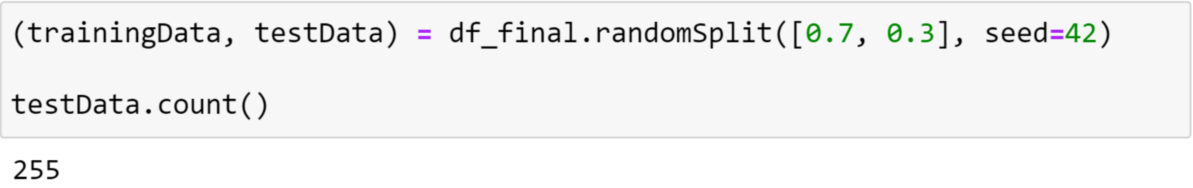
The training df contains 741 records using seed of 42



```
(trainingData, testData) = df_final.randomSplit([0.7, 0.3], seed=42)

trainingData.count()
```

```
741
```

The Testing df contains 255 records using seed of 42

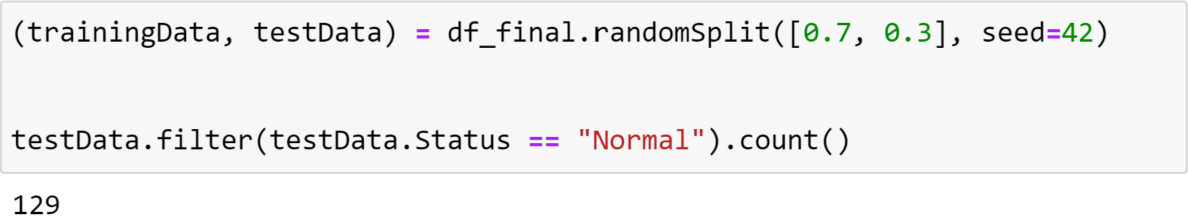


```
(trainingData, testData) = df_final.randomSplit([0.7, 0.3], seed=42)

testData.count()
```

```
255
```

Getting the amount of each group in the testing dataset

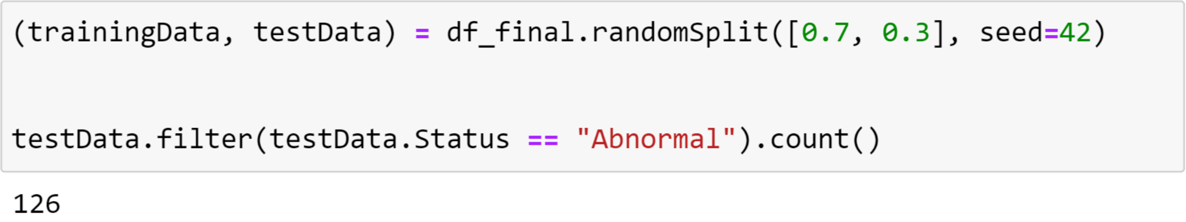
129 samples with Status Normal



```
(trainingData, testData) = df_final.randomSplit([0.7, 0.3], seed=42)


testData.filter(testData.Status == "Normal").count()
```

```
129
```

126 samples with Status Abnormal



```
(trainingData, testData) = df_final.randomSplit([0.7, 0.3], seed=42)


testData.filter(testData.Status == "Abnormal").count()
```

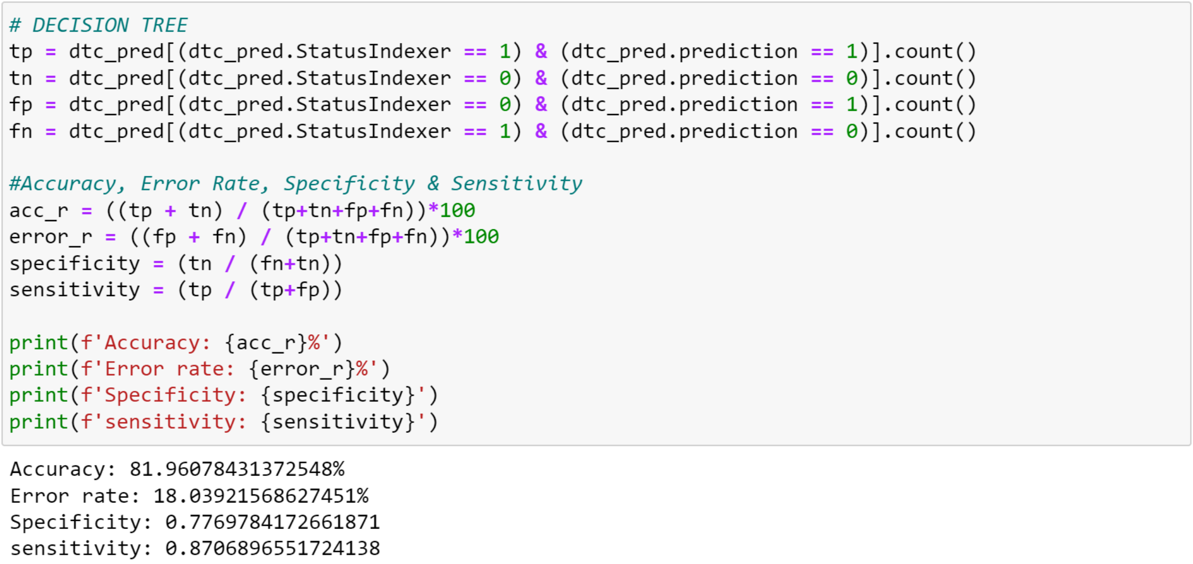
```
126
```

Task 5

Confusion matrix:

Decision Tree:

N = 255	Positive	Negative
Positive	101	108
Negative	15	31

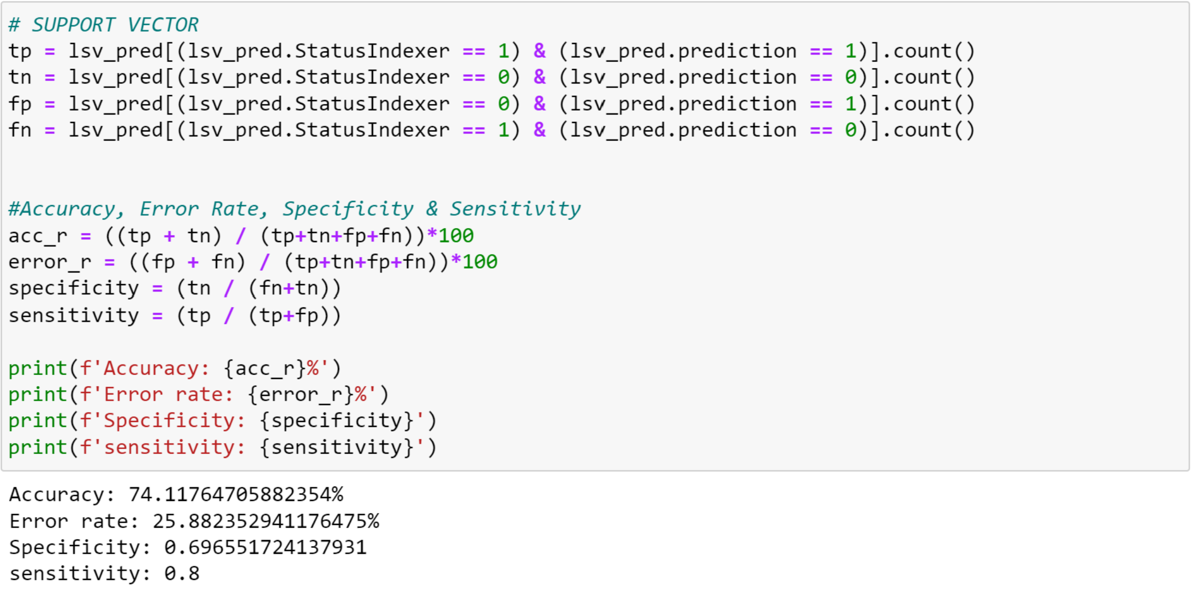


```
# DECISION TREE
tp = dtc_pred[(dtc_pred.StatusIndexer == 1) & (dtc_pred.prediction == 1)].count()
tn = dtc_pred[(dtc_pred.StatusIndexer == 0) & (dtc_pred.prediction == 0)].count()
fp = dtc_pred[(dtc_pred.StatusIndexer == 0) & (dtc_pred.prediction == 1)].count()
fn = dtc_pred[(dtc_pred.StatusIndexer == 1) & (dtc_pred.prediction == 0)].count()

#Accuracy, Error Rate, Specificity & Sensitivity
acc_r = ((tp + tn) / (tp+tn+fp+fn))*100
error_r = ((fp + fn) / (tp+tn+fp+fn))*100
specificity = (tn / (fn+tn))
sensitivity = (tp / (tp+fp))

print(f'Accuracy: {acc_r}%')
print(f'Error rate: {error_r}%')
print(f'Specificity: {specificity}')
print(f'sensitivity: {sensitivity}')
```

```
Accuracy: 81.96078431372548%
Error rate: 18.03921568627451%
Specificity: 0.7769784172661871
sensitivity: 0.8706896551724138
```

Support Vector Machine:

N = 255	Positive	Negative
Positive	88	101
Negative	22	44

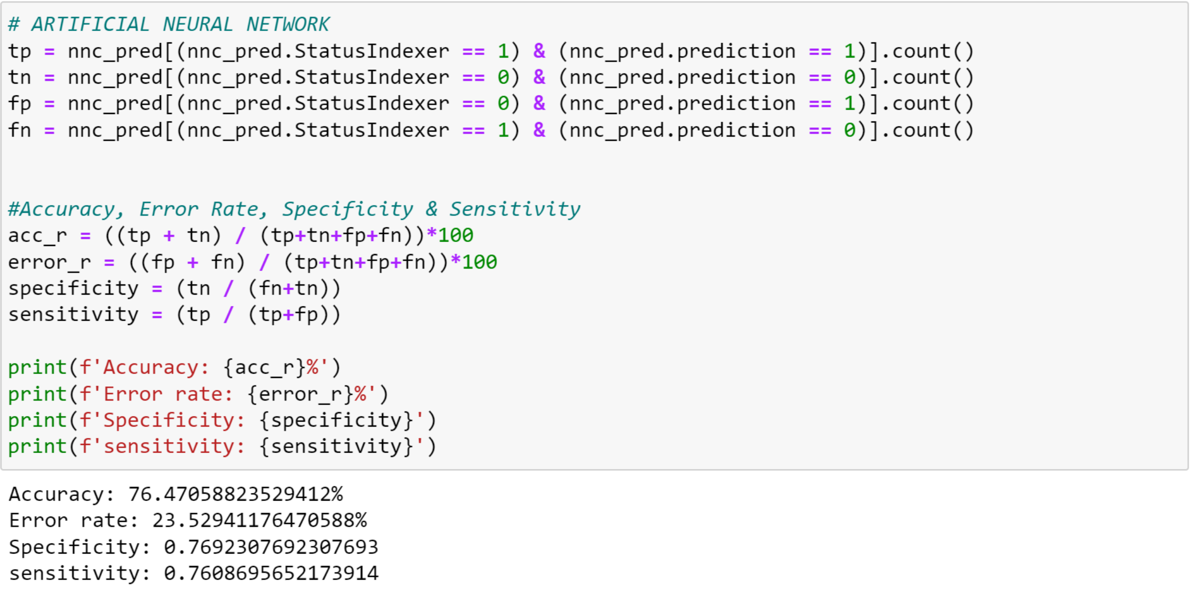


```
# SUPPORT VECTOR
tp = lsv_pred[(lsv_pred.StatusIndexer == 1) & (lsv_pred.prediction == 1)].count()
tn = lsv_pred[(lsv_pred.StatusIndexer == 0) & (lsv_pred.prediction == 0)].count()
fp = lsv_pred[(lsv_pred.StatusIndexer == 0) & (lsv_pred.prediction == 1)].count()
fn = lsv_pred[(lsv_pred.StatusIndexer == 1) & (lsv_pred.prediction == 0)].count()


#Accuracy, Error Rate, Specificity & Sensitivity
acc_r = ((tp + tn) / (tp+tn+fp+fn))*100
error_r = ((fp + fn) / (tp+tn+fp+fn))*100
specificity = (tn / (fn+tn))
sensitivity = (tp / (tp+fp))

print(f'Accuracy: {acc_r}%')
print(f'Error rate: {error_r}%')
print(f'Specificity: {specificity}')
print(f'sensitivity: {sensitivity}')
```

```
Accuracy: 74.11764705882354%
Error rate: 25.882352941176475%
Specificity: 0.696551724137931
sensitivity: 0.8
```

Artificial Neural Network:

N = 255	Positive	Negative
Positive	105	90
Negative	33	27



```
# ARTIFICIAL NEURAL NETWORK
tp = nnc_pred[(nnc_pred.StatusIndexer == 1) & (nnc_pred.prediction == 1)].count()
tn = nnc_pred[(nnc_pred.StatusIndexer == 0) & (nnc_pred.prediction == 0)].count()
fp = nnc_pred[(nnc_pred.StatusIndexer == 0) & (nnc_pred.prediction == 1)].count()
fn = nnc_pred[(nnc_pred.StatusIndexer == 1) & (nnc_pred.prediction == 0)].count()


#Accuracy, Error Rate, Specificity & Sensitivity
acc_r = ((tp + tn) / (tp+tn+fp+fn))*100
error_r = ((fp + fn) / (tp+tn+fp+fn))*100
specificity = (tn / (fn+tn))
sensitivity = (tp / (tp+fp))

print(f'Accuracy: {acc_r}%')
print(f'Error rate: {error_r}%')
print(f'Specificity: {specificity}')
print(f'sensitivity: {sensitivity}')
```

```
Accuracy: 76.47058823529412%
Error rate: 23.52941176470588%
Specificity: 0.7692307692307693
sensitivity: 0.7608695652173914
```

Task 6

Based on the results obtained in task 5, I would use the Decision Tree as the best means to classify the reactors status as it returns the highest accuracy of the three, or lowest error rate.

Task 7

The decision tree returns the lowest error rate of xxx.

Task 8

Bibliography

https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/