REPORT ON EVALUATION OF BIG DATA CLOUD PLATFORM AND DATA ANALYSIS WITH PYTHON

BY

FELIX OPARAH

SECTION ONE: INTRODUCTION

1.1 Background

All over human endeavours, both structured and unstructured data are generated in high volumes and velocities daily. Thus, analysing and management of such quantum of data in order to make meaningful decisions, predictions and planning, in both the private and public sectors become pertinent. Several approaches, techniques, computer algorithms, computer software, and platforms have been developed in recent times in a bid to unravel this phenomenon, while others are still evolving.

One of such platforms developed is the cloud. The cloud which refers to servers, software and databases that can be accessed over the internet based on the cloud computing technology, has become a vital aspect of data analytics. Contemporary cloud data warehouses aid users analyse big data at high levels of data granularity. Some of the cloud data warehouses were investigated in this study.

Similarly, one of such computer software developed for the analysis of data is the Python Programming language. The Python Programming Language was adopted in this study for analysis of big datasets so as to make informed decisions in business and other organisations.

1.2 Objective of investigation

The objectives of this study are:

a) To evaluate a sample of 4 big data supported cloud platforms offering data warehouse implementation based on 8 criteria which are important to the success of any organization in this area. These cloud platforms are:

i. BigQuery (Google)
ii. Azure (Microsoft)
iii. Keboola
iv. Red Hat OpenShift

b). To analyse big dataset by implementing Exploratory Data Aanalysis (EDA), Classification and Prediction, using Python programming language.

1.3 Evaluation criteria

The criteria for evaluating the cloud platforms adopted in this study and the associated categories are as shown in the table below:

Table 1: Evaluation Criteria table

CATEGORIES		CRITERIA	
1	Performance at scale	i	Is there the possibility that compute resources will meet future scaling needs?
		ii	Are data compression, indexing and pruning supported by the platform?
2	Elasticity	i	Can compute and storage scale independently for the central processing unit, memory space, capacity and performance?
		ii	Which controls exist to avoid cost overruns ?
3	Ease of use	i	Does the technology need infrastructure management (servers, storage etc) ?
		ii	Is the platform able to offer Infrastructure as a service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS)?
4	Cost efficiency	i	Does the platform charge for data capacity or consumed-compressed capacity?
5	Ability to support structured and semi-structured data	i	Is semi-structured data supported by the platform?

Source: Author

1.4 Methodology

The methodology of investigation in this study was both analytical and quantitative. Features and capabilities of each of the various cloud databases were investigated and analysed. These were equally compared among the various platforms to determine their strengths and weaknesses, so as to provide appropriate insights regarding their overall effectiveness. Furthermore, Python programming (on Jupyter) was used to process some selected big dataset. EDA, Classification and Prediction were done using the datasets. This helped to demonstrate the learning outcomes of course: Processing Big Data.

SECTION TWO: PLATFORM INVESTIGATION

In this section, the four selected platforms were evaluated based on the criteria stated in 1.3 above.

2.1 Performance at scale

Performance of a platform can be described as the ability of a platform to provide for smooth and speedy data querying and Extract, Load Transform/Extract, Transform & Load (ETL/ELT) processes. A good platform should be able to enable users analyse big data at high granularity.

Criteria 1: Is there the possibility that compute resources will meet future scaling needs?

Under this criterion, the platforms were evaluated to see whether compute resources will meet future scaling needs.

i. BigQuery (Google). It functions based on SaaS principles and can be scaled up to process up to hundreds of petabytes. It can be set up in a matter of minutes and does not require the user to provide or manage servers. BigQuery is suitable for organisations who want to preserve their data capabilities now and in the future.
ii. Azure (Microsoft): This database can be easily scaled to fit user needs. More compute or storage can be added to satisfy performance needs without migrating data to more powerful machines. Vertical and horizontal scaling are available.
iii. Keboola provides scalable, secure storage for both structured and unstructured data. It is powered by Snowflakes which (SaaS) data warehouse and offers high level scalable services for data storage, compute, and analytics.
iv. Red Hat OpenShift. Red Hat Openship has a feature called Autoscaling which enables scaling of applications as need be and based on certain specifications.

Criteria 2: Are data compression, indexing and pruning supported by the platform?

i. BigQuery (Google). BigQuery divides table data into smaller units called partitions based on time units such as date or time. Such columnar data are scanned to filter using the partition keys. Partitions that don't match the filter are skipped. By this process, data are pruned. Partitioning can speed up a query that filters on the partition key while older partitions are automatically deleted. Partitioning is a way of indexing data and could improve database performance by reducing the amount of data to be queried.

ii. Azure (Microsoft): This platform provide for row and page compression for rowstore tables and indexes. They also support columnstore archival compression for columnstore tables and indexes.

iii. Keboola. Keboola has unique features which enable data to be pruned, compressed, and indexed. The primary index defines the sort order on the table, which can include one or many fields. Primary indexes are mandatory for fact tables and optional for dimension tables.

iv. Red Hat OpenShift. Openshift can prune data and remove older versions that are no longer required. It can also compress data however, allowing compression can influence application performance and might prove unproductive when data to be processed is already compressed or encrypted.

2.2 Elasticity.

Elasticity is the ability of the cloud to quickly expand or decrease computer processing, memory and storage resources to adapt to changing demands of an organization. This will ensure that an organization does not pay for unused capacity or idle resources. Considering this criterion, the following questions will be answered for all the platforms:

Criteria 3: Can compute and storage scale independently for the central processing unit, memory space, capacity and performance?

i. BigQuery (Google). BigQuery is highly elastic for CPU, memory and storage capacity. It scales to any size, swiftly and effortlessly.

ii. Azure (Microsoft): Azure employs serverless offering which also encompasses elasticity in CPU, memory and storage capacities. The SQL Server Stretch Database offering is an example of high elasticity in Azure. It is automatic making it possible to ensure elasticity without any human interventions.

iii. Keboola. Keboola Connection offers elasticity in CPU, memory and storage for both structured and unstructured data. It helps you swiftly expand computer processing, memory, and storage resources according to your needs.

iv. Red Hat OpenShift offers elasticity of cloud database resources.

Criteria 4: What controls exist to avoid cost overruns?

i. Bigquery: BigQuery adopts two(2) pricing models for running queries: on-demand pricing and flat rate pricing. The first is structured such that you pay for the number of bytes processed by each query you make while using flat rate pricing makes you pay for a stipulated slot or query processing capacity. You can directly reduce cost under On-demand pricing while you can do same under fixed rate pricing by purchasing the number of slots you actually need.

ii. Azure (Microsoft): Workload architecture can be optimized for cost savings in Azure. The cost management concentrates on establishing budgets, monitoring cost allocation pattern and adopting controls. Overspending or unbudgeted spending is reduced through performance monitoring, resource sizing and safely terminating idle resources. Different models of Azure can be implemented based on need and affordability. These include AWS, Azure for Windows Server, Azure SQL Server, Azure Hybrid Benefit, etc.

iii. Keboola has differentiated pricing models to fit user requirements and avoid cost overrun. Keboola pricing is based on usage unlike some other platforms that you pay for more connectors. Total spending in minimized as Keboola is free for certain users.

iv. Red Hat OpenShift: Red Hat Openshift has an SaaS that is added to the subscription at no additional subscription cost. This allows you view your costs all over on-premise and public cloud environment. Cost control for Red Hat OpenShift offers Information Technology and financial managers a distinctive view of the costs associated with the application running on the platform and others public cloud platforms.

2.3 Ease of use

Contemporary data cloud warehousing platforms are becoming more user friendly and aims at the utilization of resources to productive data processing. Hence, a good platform should easily provide for clusters, installations and hardware while maintaining great user experience. It should provide for performance, handling of semi structured data, assigning resources to users and integrates programming languages.

Given the above, the following criteria will be evaluated.

Criteria 5: Does the technology need infrastructure management (servers, storage etc)?

i. BigQuery (Google). As a cloud database platform, Bigquery enables infrastructure management such as servers, storage, processing units, etc. It is part of Google cloud that provide infrastructure as a service. It provides serverless, scalable infrastructure; thus It eliminates the cost of procuring and managing on-premise hardware. BigQuery automatically allocates storage when data is loaded into it.
ii. Azure (Microsoft): Microsoft Azure is one of the biggest cloud platforms which provide IaaS. It offers infrastructure management essentially storage, compute and networking resources on-demand. Usually, this is done on pay as you go basis.
iii. Keboola is a Platform as a Service that offers data integration with pre-built connectors. These connectors enable the integration of SaaS applications and data storage. Furthermore, Keboola is a SaaS platform that enables full infrastructure management. In addition to data connectors, it provides for storage, transformation backends, databases and server resources.
iv. Red Hat OpenShift provides IaaS, PaaS and SaaS offerings that integrate into a cloud computing environment with the associated infrastructure, platform and applications that any user requires.Red Hat's cloud infrastructure products allows their customer build and manage an IaaS cloud. It also offers storage and container arrangement platforms

Criteria 6: Is the platform able to offer Infrastructure as a service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS)?

i. BigQuery (Google). BigQuery provides Software as a Service (SaaS) data warehouse platform.
ii. Azure (Microsoft): It offers Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS).
iii. Keboola. Keboola is a SaaS platform.
iv. Red Hat OpenShift is mainly a Platform as a Service (Paas) platform.

2.4 Cost efficiency

The patterns of pricing for cloud data warehouse prototypes are not usually uniform across different platforms. They are based on different factors such as speed, size of data, usage, etc. Thus, the following question shall be evaluated for the different platforms:

Criteria 7: Does the vendor charge for data set capacity or consumed-compressed capacity?

i. BigQuery (Google). BigQuery uses on-demand model whereby it charges for both storage capacity and for data scanned by queries. It uses a columnar data structure and apportions charges based on data size per column. Storage capacity charges are equally based on active and long term data capacities.
ii. Azure (Microsoft) majorly based on pay as you go" pricing model, which charges users based on actual usage. billed per second, with no long-term commitment. There are other types of pricing like spot pricing and storage pricing. The storage pricing is based on scalable storage capacity charged per gigabyte.
iii. Keboola. Keboola is based on a free pricing version limited to 250GB storage capacity and a paid version on a 'Pay as you go' pricing model.
iv. Red Hat OpenShift. The pricing is structured differently for cloud services and self-managed services on own infrastructure. However, it has differentiated pricing based on 4vCPU and minimum worker load is required.

2.5 Ability to support structured and semi-structured data

Using semi-structured data can be very beneficial in data analytics. However, some old-style data warehouse platforms do not have the capacity to handle such. Users waste resources on inefficient flattening and un-nesting, which increases the number of rows/cells in the table. Hence tables are made bigger which negatively impinges on cost and performance. However, a modern data warehouse will enable query of semi-structured data with the standard SQL and without complicated ETL processes. Thus, the platforms under review will be evaluated with the following criteria:

Criteria 8: Is semi-structured data supported by the platform?

i. BigQuery (Google). BigQuery supports the processing of semi-structured data using the JSON data type. Such data are encoded and processed independently. Queries can be run on the fields in the JSON data dot.notation and this makes it simple to handle. Structured data is equally supported by Bigquery.

ii. Azure (Microsoft). Azure support processing of semi structured data or non-relational (NoSQL) data using XML and JavaScript Object Notation (JSON) options. Structured data is equally supported

iii. Keboola is able to handle both structured and semi structured data in various forms as XML, JSON and even Metadata.

iv. Red Hat OpenShift. Supports both structured and semi-structured data with applications like Spark, Presto, Red Hat AMQ Streams (Kafka), etc. Red Hat OpenShift Container Storage provides support for all types of storage, including file, block, and object.

2.6 The comparation result.

The investigation of the different platform above shows that in general, platforms such as Bigquery and Azure are more robust than Keboola and Red Hat OpenShift. All the platforms measure well in Elasticity and Ease of use, however Bigquery and Azure have an edge here.

Similarly, on the ability to meet scaling needs, all the platforms have some good ability to meet these needs. All the platform are also able to handle both structured and semi-structured data. In the area of cost efficiency, different platforms have varying cost structures hence, cost efficiency is accessed in conjunction with the applicable service offering at various degrees. However, Keboola and Red Hat OpenShift price certain services lower than Bigquery and Azure. Some of the platforms like Keboola have some sort of free pricing regime limited to limited to 250GB storage capacity (in the case of Keboola). However, when choosing a platform for any business entity, the decision has to be more of a holistic one in view of technical constraints as well as cost implications.

SECTION THREE: BIG DATA PROCESSING AND ANALYSIS IMPLEMENTATION

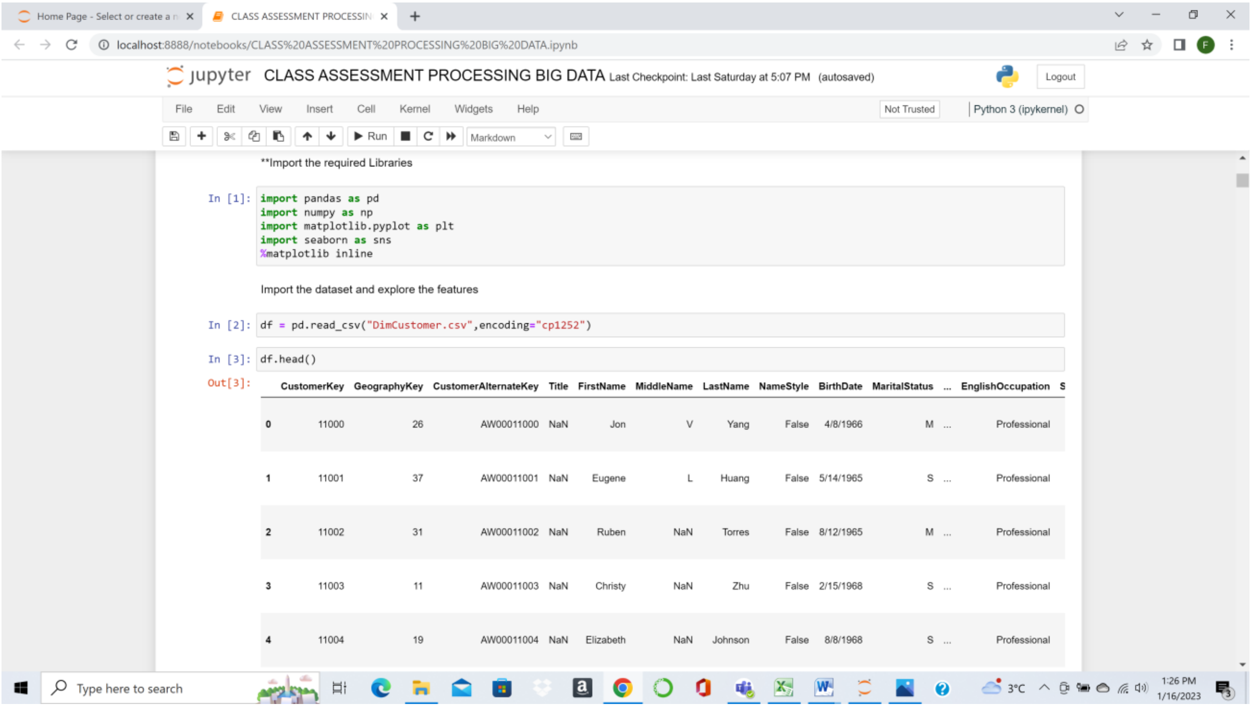
In this section, some of the learning outcomes of the course were demonstrated by way of analyzing some big dataset using Python on the Jupyter Notebook. Exploratory Data Analysis (EDA), Classification and Prediction were done in this project. Three datasets were used for the project: Dataset 1, Dataset 2, and Dataset 3. Dataset 1 was used for EDA only, Dataset2 for Classification and prediction while Dataset3 was used for prediction.

3.1. Exploratory Data Analysis

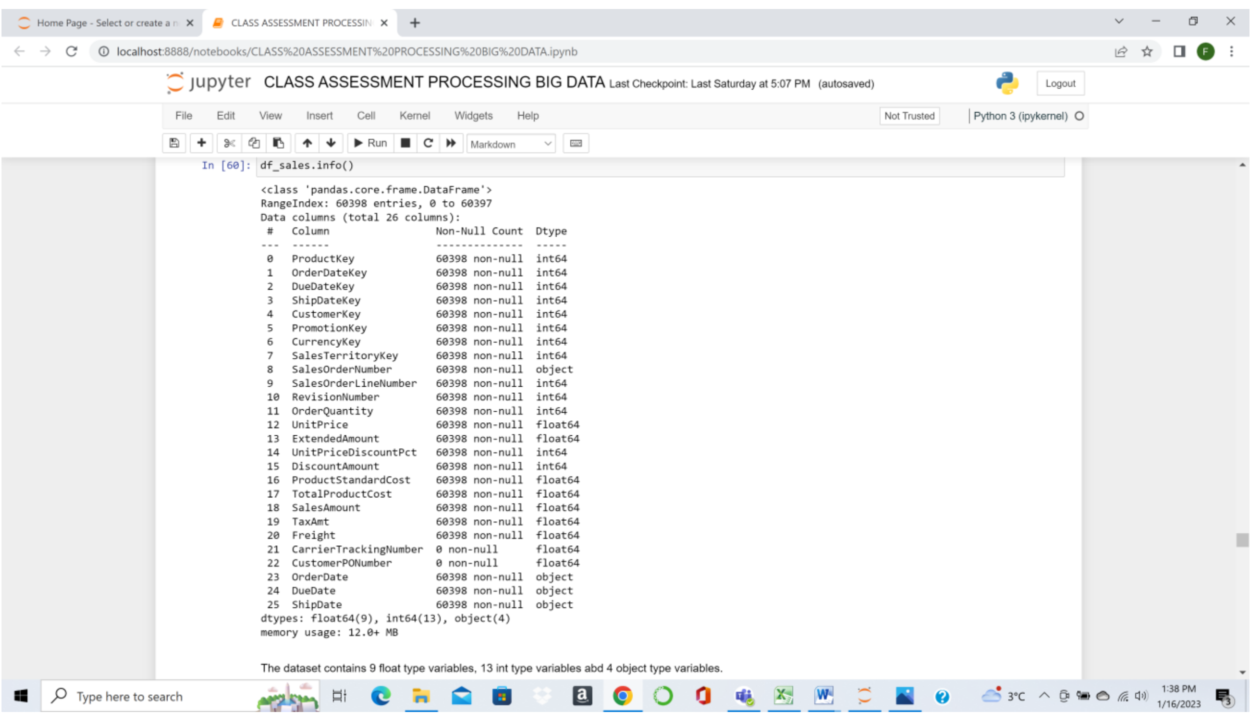
EDA was performed on the 3 datasets used for this analysis. Properties of the datasets were evaluated by looking at the head, tail, summary, etc. Some columns which are not needed were dropped through filtering and data cleaning. Data visualization was done through histogram, boxplot, scatter diagram, etc. Samples from EDA on dataset 1 is presented in this report.

Dataset 1 is named “DimCustomer”. It is about customers' bio details and yearly income. It consists of 29 variables and 18484 observations. Dataset 2 (“named FactCallCenter”) shows details of various Call Centers and their performance. It is made up 120 call centers with 14 variables. Similarly, Dataset 3 (FactinternetSales) is made up of 26 variabes and 60397 observations.

Dataset1 was loaded after importing the necessary Libraries for the analysis. The head function was used to call up the data in tables as given below.

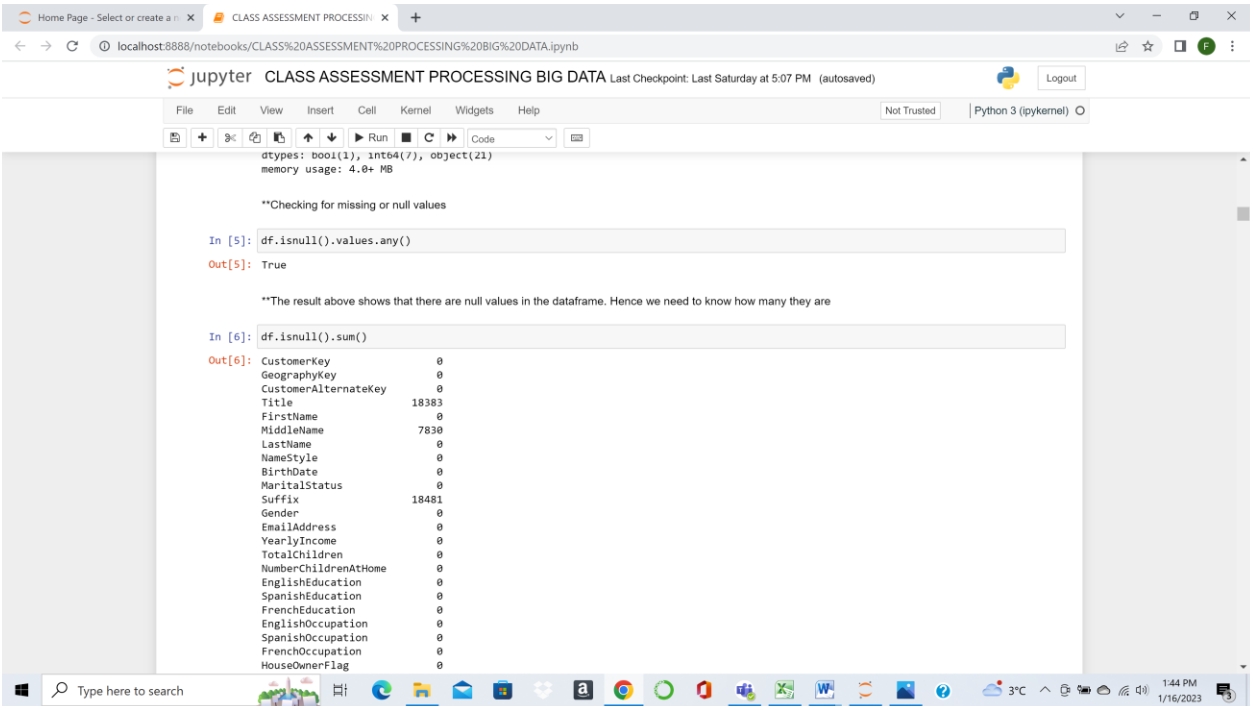


The characteristics of dataset1 were explored using the info() function.



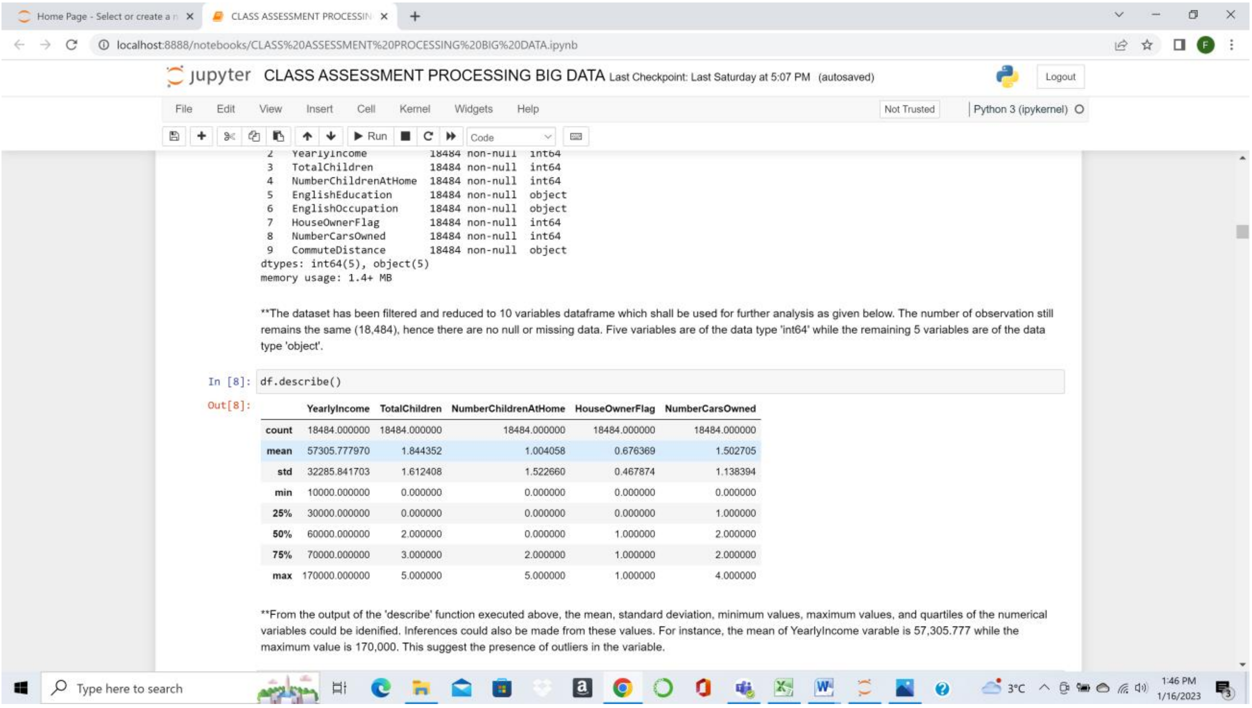
The result above shows the dataset contains 1 bool type variable, 7 int64 type variables and 21 object type variables.

Missing or null values were checked and removed:



The dataset was filtered and reduced to 10 variables dataframe which was used for further analysis as given below. The number of observation still remains the same (18,484), hence there are no more null or missing data. Five variables are of the data type 'int64' while the remaining 5 variables are of the data type 'object'.

The .describe() function was used to examine the statistical properties of the dataset such the mean, standard deviation, min, max, etc:



```
2   YearlyIncome          18484 non-null  int64
3   TotalChildren         18484 non-null  int64
4   NumberChildrenAtHome  18484 non-null  int64
5   EnglishEducation      18484 non-null  object
6   EnglishOccupation     18484 non-null  object
7   HouseOwnerFlag        18484 non-null  int64
8   NumberCarsOwned       18484 non-null  int64
9   CommuteDistance       18484 non-null  object
dtypes: int64(5), object(5)
memory usage: 1.4+ MB
```

**The dataset has been filtered and reduced to 10 variables dataframe which shall be used for further analysis as given below. The number of observation still remains the same (18,484), hence there are no null or missing data. Five variables are of the data type 'int64' while the remaining 5 variables are of the data type 'object'.

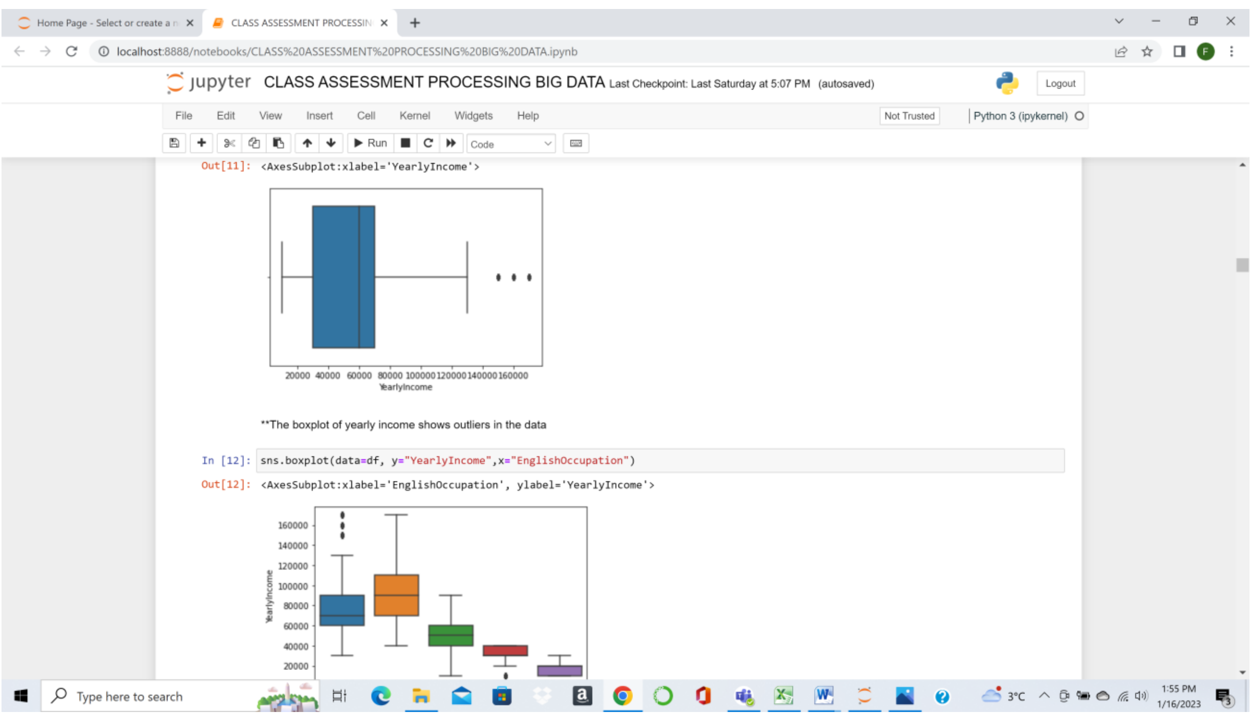
In [8]:
```
df.describe()
```

Out[8]:

	YearlyIncome	TotalChildren	NumberChildrenAtHome	HouseOwnerFlag	NumberCarsOwned
count	18484.000000	18484.000000	18484.000000	18484.000000	18484.000000
mean	57305.777970	1.844352	1.004058	0.676369	1.502705
std	32285.841703	1.612408	1.522660	0.467874	1.138394
min	10000.000000	0.000000	0.000000	0.000000	0.000000
25%	30000.000000	0.000000	0.000000	0.000000	1.000000
50%	60000.000000	2.000000	0.000000	1.000000	2.000000
75%	70000.000000	3.000000	2.000000	1.000000	2.000000
max	170000.000000	5.000000	5.000000	1.000000	4.000000

**From the output of the 'describe' function executed above, the mean, standard deviation, minimum values, maximum values, and quartiles of the numerical variables could be idenified. Inferences could also be made from these values. For instance, the mean of YearlyIncome varable is 57,305.777 while the maximum value is 170,000. This suggest the presence of outliers in the variable.

Data visualization was performed to have more insight into the properties of the dataset:

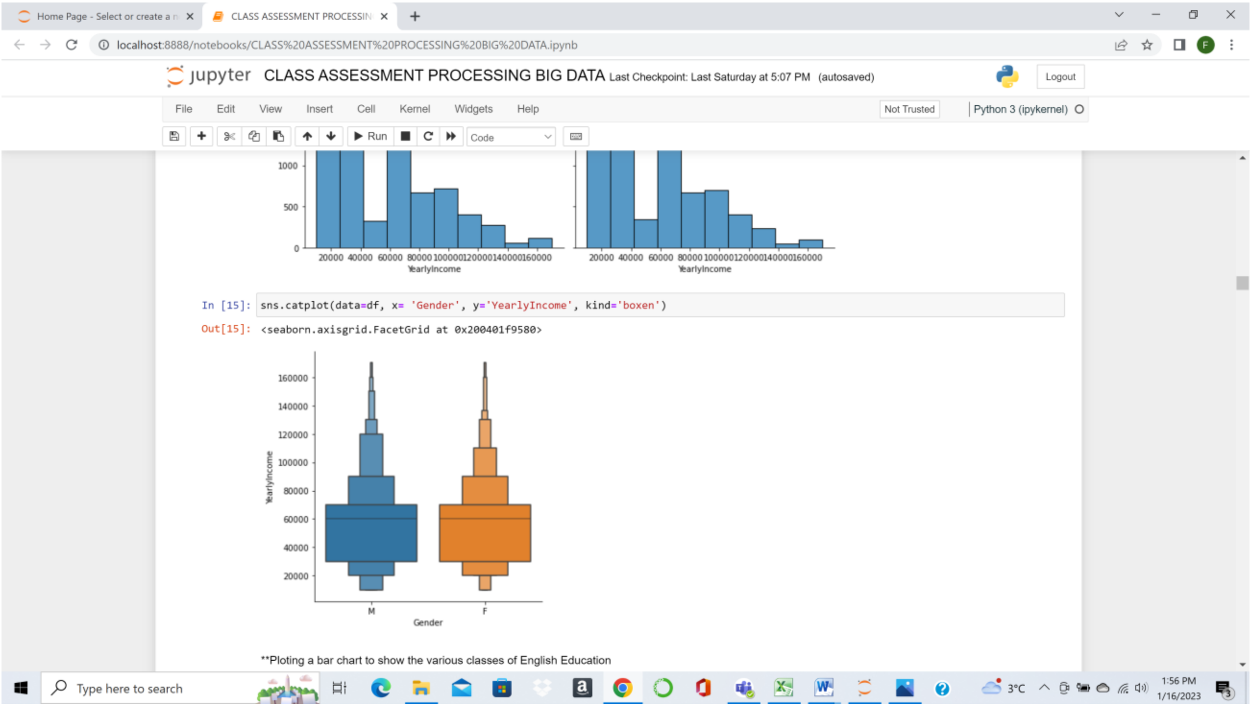


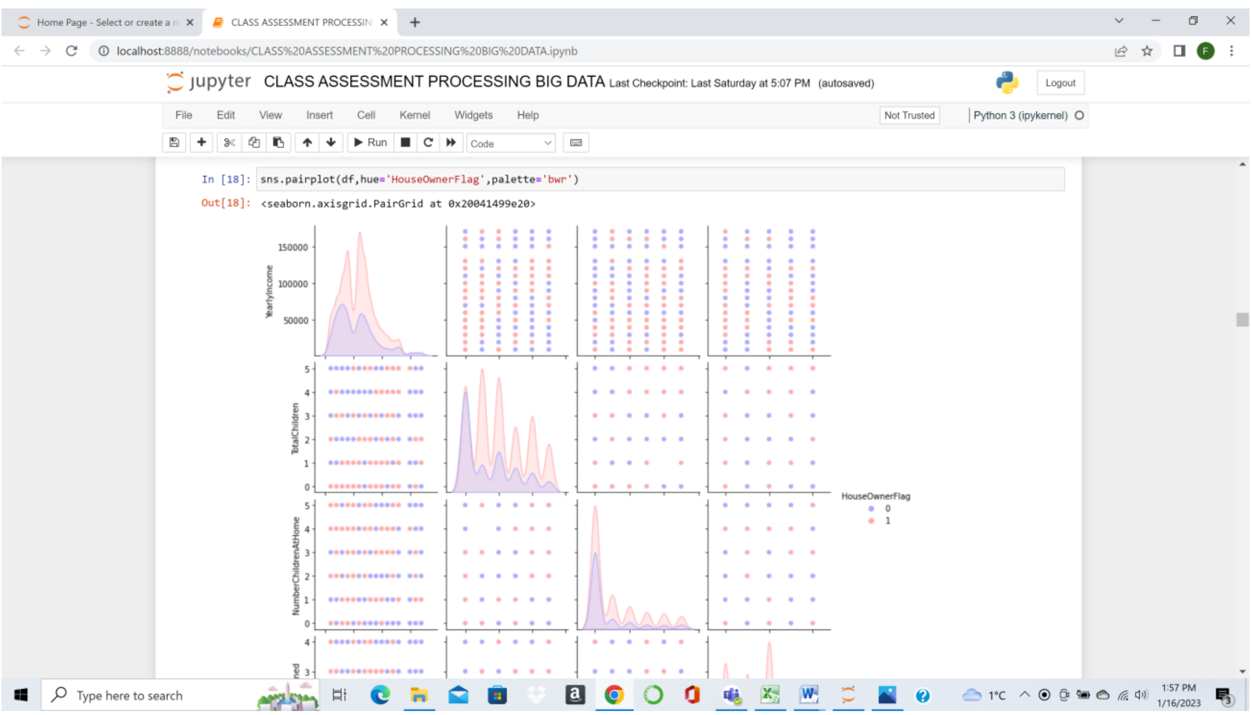
Out[11]: <AxesSubplot:xlabel='YearlyIncome'>

**The boxplot of yearly income shows outliers in the data

In [12]:
```
sns.boxplot(data=df, y="YearlyIncome",x="EnglishOccupation")
```

Out[12]: <AxesSubplot:xlabel='EnglishOccupation', ylabel='YearlyIncome'>

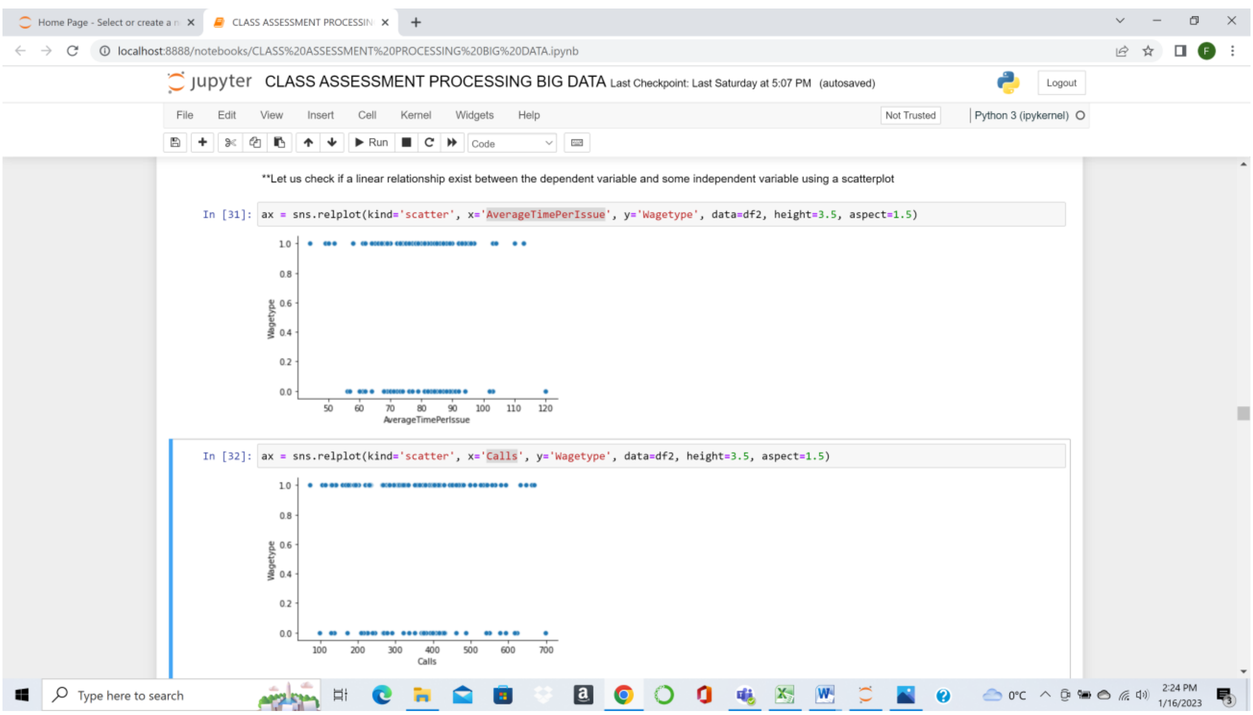




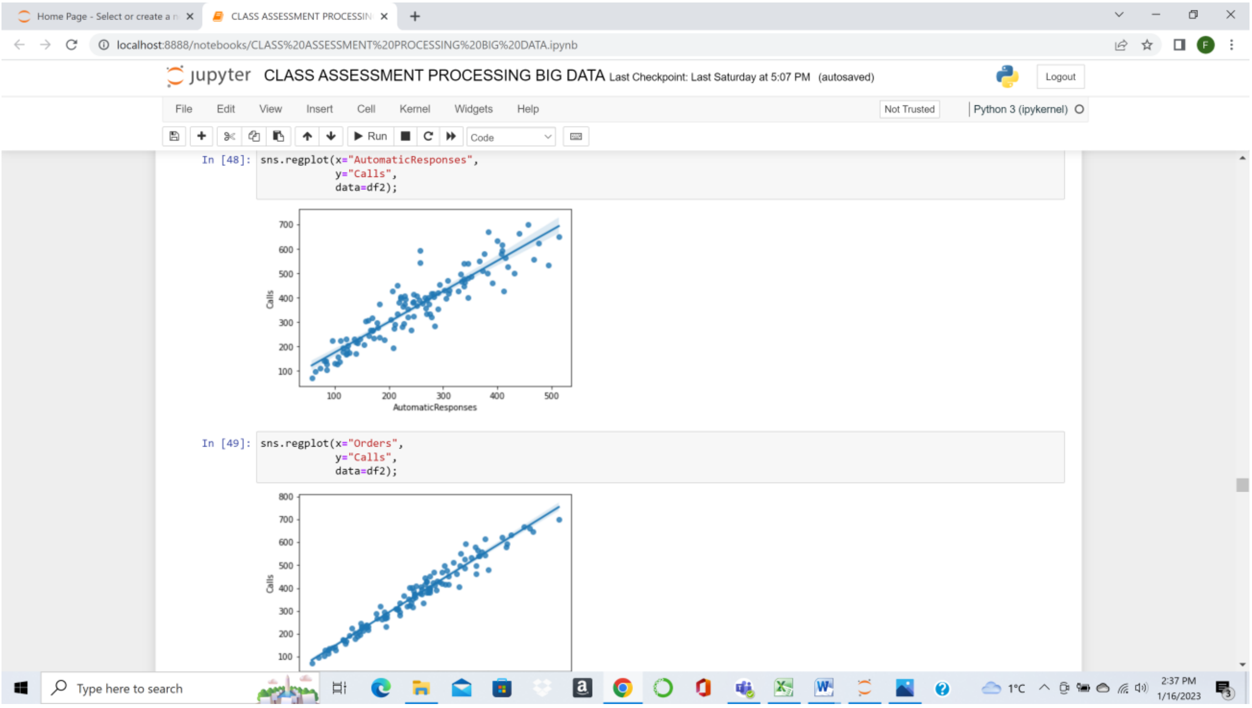
Summary of EDA and Justification.

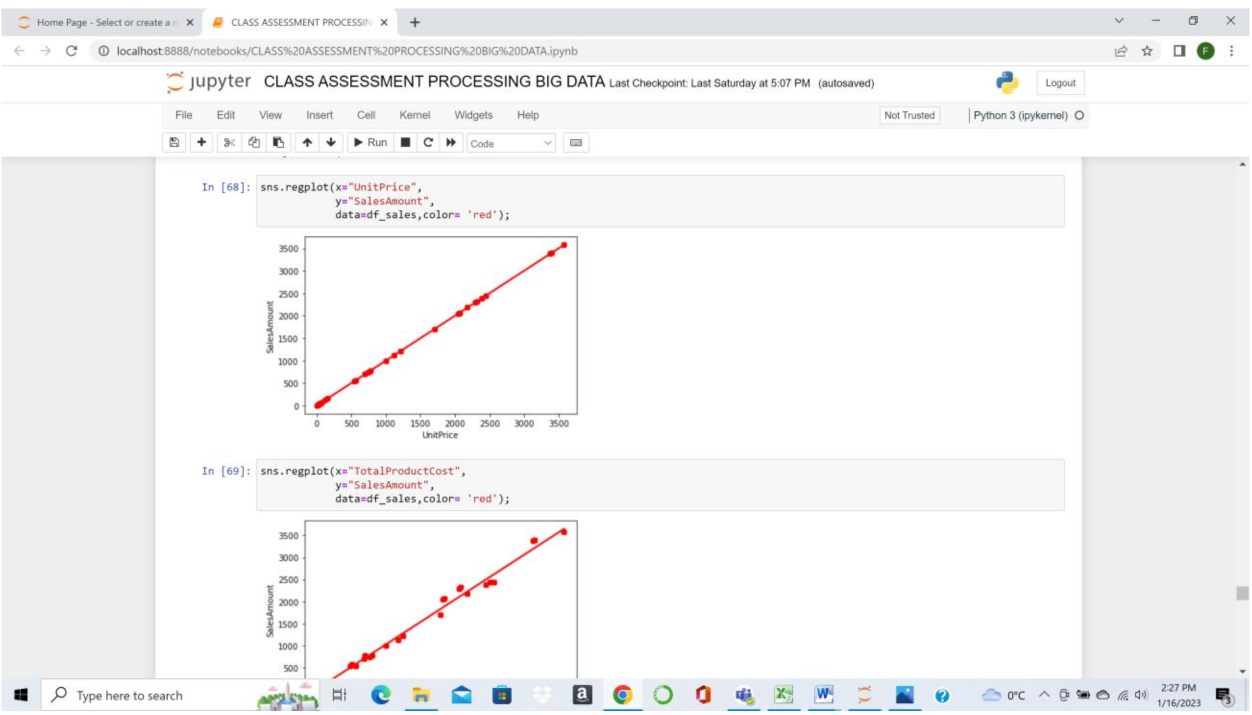
EDA enabled the understanding of the data, removal of inaccuracies, understanding of the patterns/trends and determine the appropriateness of the data for statistical analysis. EDA also helps to determine if the data meet the underlying assumptions of the statistical analysis to be performed.

After EDA was performed on all the 3 datasets, it was determined that Dataset2 is more appropriate for Classification using Logistic regression. Prediction with Linear regression was more appropriate using Dataset2 and Dataset3, while Dataset1 was least appropriate.. This is because, these more appropriate datasets conform to the assumptions of these techniques accordingly. For instance, in Dataset 2, most of the independent variables such as “AverageTimePerIssue”, “Calls” and “AutomaticResponses” do not show linear relationship with the dependent variable “Wagetype” to warrant a Linear Regression model. However, the relationship was consistent with logistic regression. This is depicted in the scatter plot below:



However, the dataset was appropriate for Linear regression when the variable “Calls” is being investigated as the dependent variable. On the other hand, in Dataset3, the relationship between the dependent variable and the independent variables was discovered to be linear during EDA as shown below:





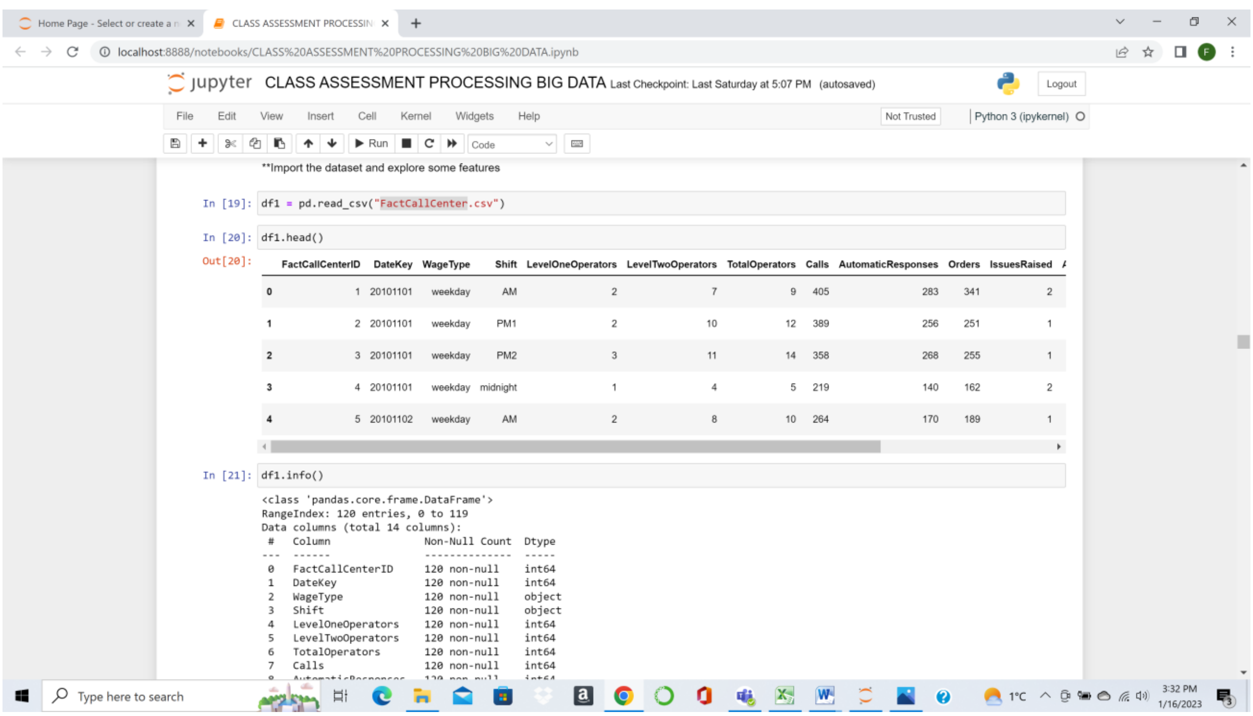
3.2 Classification

Classification helps to determine the possible class outcome in a Binary predictive model. Other classes of Classification include: Multi-Class Classification, Multi-Label Classification and Imbalanced Classification. The Binary Classification which normally follows a Bernoulli probability distribution has an outcome of 0 or 1. This was implemented in this study using a Logistic Regression model.

Dataset2 was used for this analysis. The choice of dataset2 is based on the insight gotten from the EDA performed, which shows that the dataset conformed to the basic assumptions of a Logistic Regression Model. One of such assumption is that the dependent variable must have 2 possible outcomes (0 or 1, yes or know, male or female, etc).

In this analysis, the dependent variable is WageType which has only two possible values- ‘weekday’ and ‘holiday’. This was converted to binary (0 and 1) during the analysis. The independent variables are: ‘LevelOneOperators’, ‘LevelTwoOperators’, ‘TotalOperators,’ ‘Calls’ ‘AutomaticResponses’, ‘Orders’, ‘IssuesRaised’ ‘AverageTimePerIssue’.

Dataset2 (named “FactCallCenter”) was imported and EDA performed.



CLASS ASSESSMENT PROCESSING BIG DATA Last Checkpoint: Last Saturday at 5:07 PM (autosaved)

**Import the dataset and explore some features

```
In [19]: df1 = pd.read_csv("FactCallCenter.csv")
In [20]: df1.head()
```

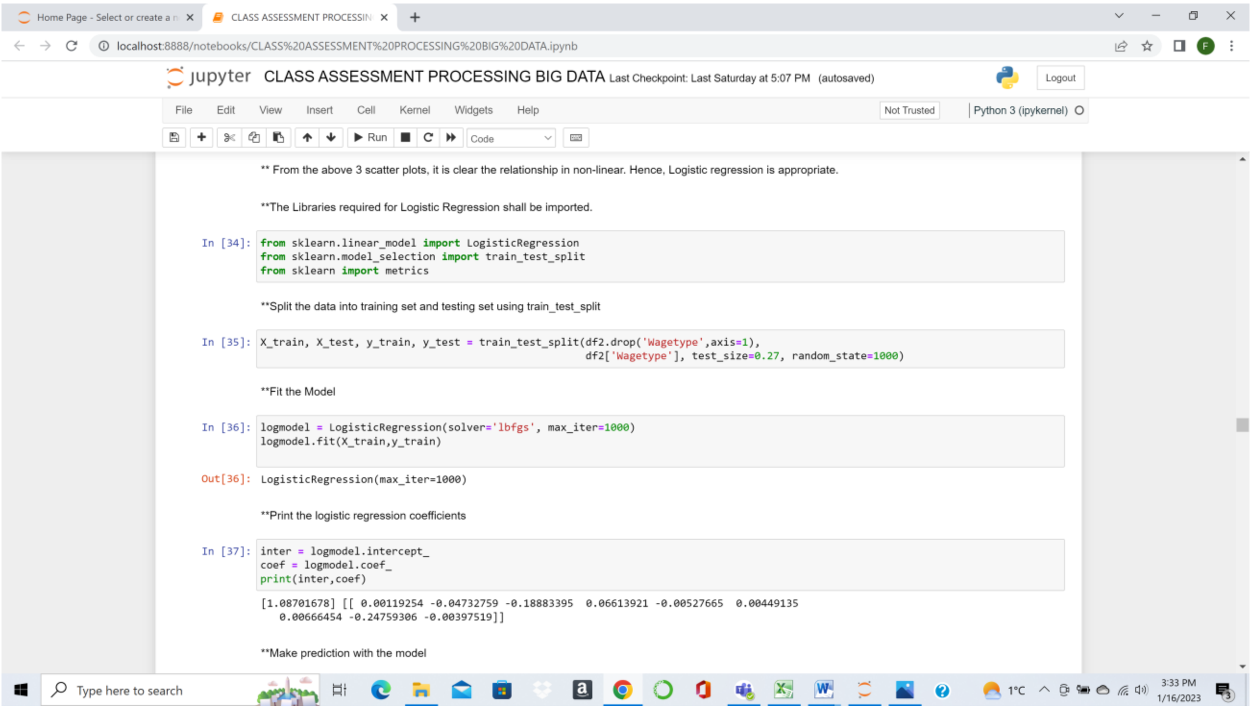
Out[20]:

	FactCallCenterID	DateKey	WageType	Shift	LevelOneOperators	LevelTwoOperators	TotalOperators	Calls	AutomaticResponses	Orders	IssuesRaised
0	1	20101101	weekday	AM	2	7	9	405	283	341	2
1	2	20101101	weekday	PM1	2	10	12	389	256	251	1
2	3	20101101	weekday	PM2	3	11	14	358	268	255	1
3	4	20101101	weekday	midnight	1	4	5	219	140	162	2
4	5	20101102	weekday	AM	2	8	10	264	170	189	1

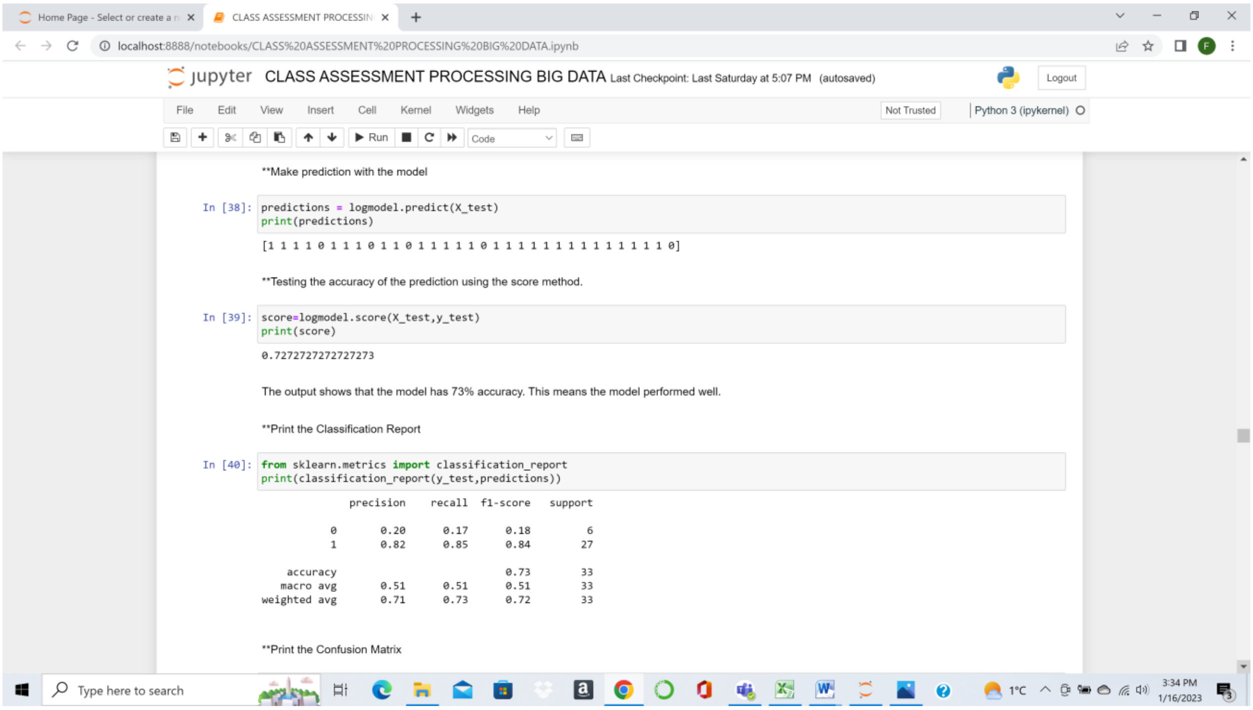
```
In [21]: df1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   FactCallCenterID     120 non-null    int64
 1   DateKey              120 non-null    int64
 2   WageType             120 non-null    object
 3   Shift                120 non-null    object
 4   LevelOneOperators    120 non-null    int64
 5   LevelTwoOperators    120 non-null    int64
 6   TotalOperators       120 non-null    int64
 7   Calls                120 non-null    int64
```

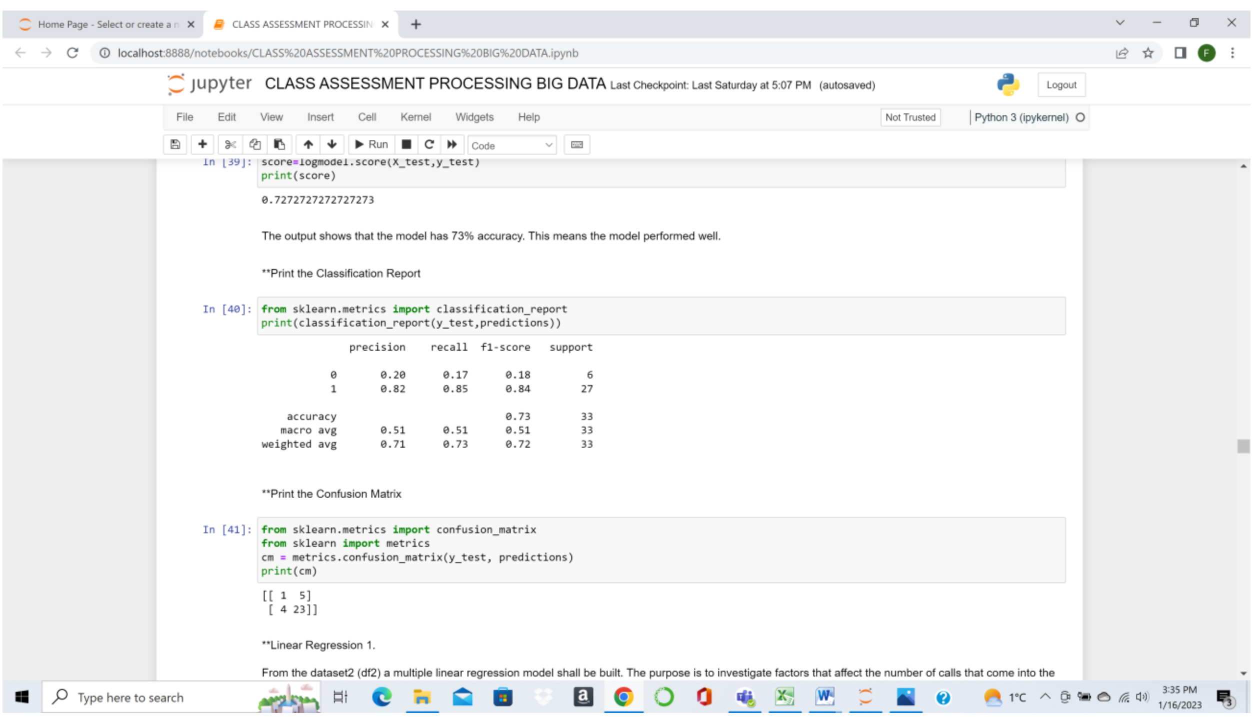
The necessary Libraries for Logistic Regression were imported. Data was trained and divided into training and test sets, with a test size of 0.27. The model was fit and model coefficients were obtained as shown below:



Prediction was made with the model with a predictive accuracy of 73%.



The classification report and the confusion matrix were also obtained.



Classification Summary

Classification was performed using Logistic Regression and Dataset2. The aim was to build a model that would properly make a binary class classification of the dependent variable (Wagetype) in the dataset. The choice of Logistic regression over other types of classification algorithm was due to the properties of data which conformed to the assumption of the Logistic regression model.

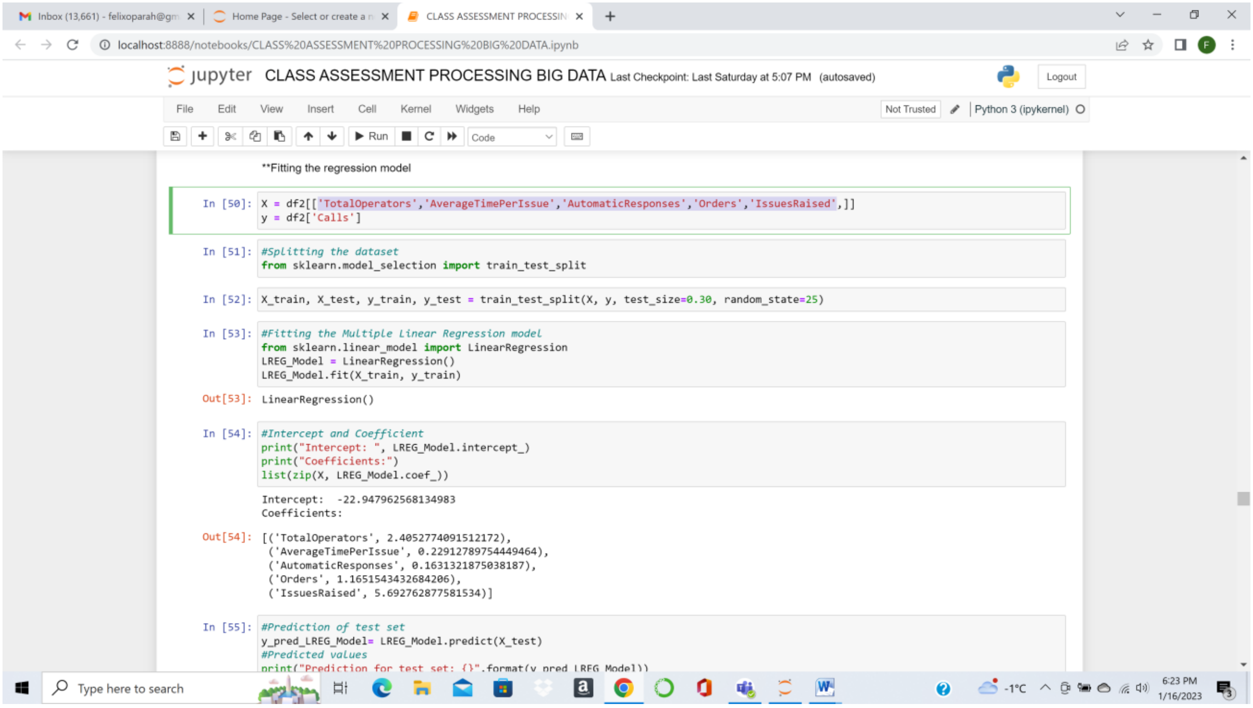
The data was trained and tested with a training size of 0.27. The result showed a relatively high accuracy rate of 73%.

3.3 Prediction

Multiple Linear Regression model was constructed to demonstrate prediction in Machine learning. Two models each from Dataset2 and Dataset3 were used and they showed impressive outcomes with high predictive ability.

Dataset2 was imported and all the necessary libraries were also imported accordingly. From the dataset, the study sought to examine factors that determine the volume of calls coming to the call centre and thereafter make prediction on the expected total number of calls. This would help the business managers prepare adequately so as to enhance customer service.

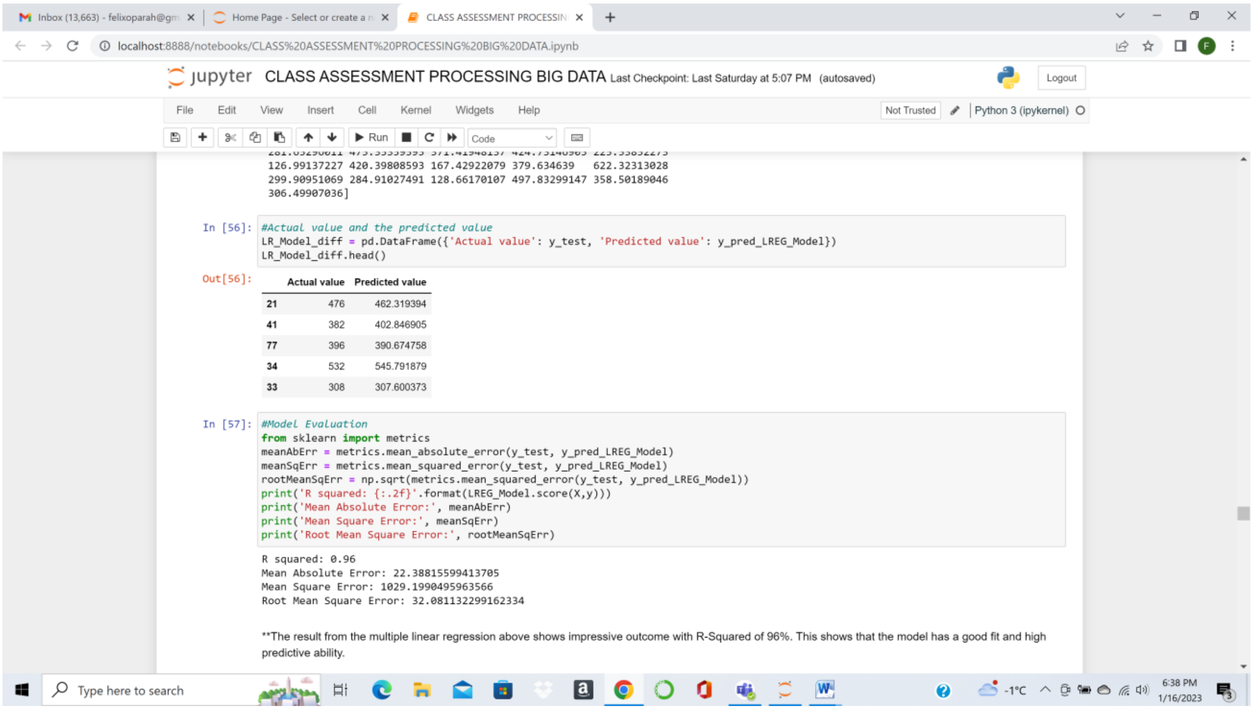
Consequently, the dependent variable is ‘Calls’ while the independent variables are: 'TotalOperators','AverageTimePerIssue','AutomaticResponses','Orders','IssuesRaised'. The data was trained with a data spilt into training and test data. Test size was 0.3 with a random state of 25. The coefficients of the explanatory variables were also obtained as shown below:



All the estimated coefficients have positive signs indicating a positive relationship between the dependent variable and the independent variable. For instance, TotalOperators with a coefficient of 2.41 shows that holding all other variables constant, one unit (one worker) increase in TotalOperators will result in 2.45 increase in calls.

Furthermore, the model was used for prediction. Actual values of the dependent variable were also compared with the predicted values.

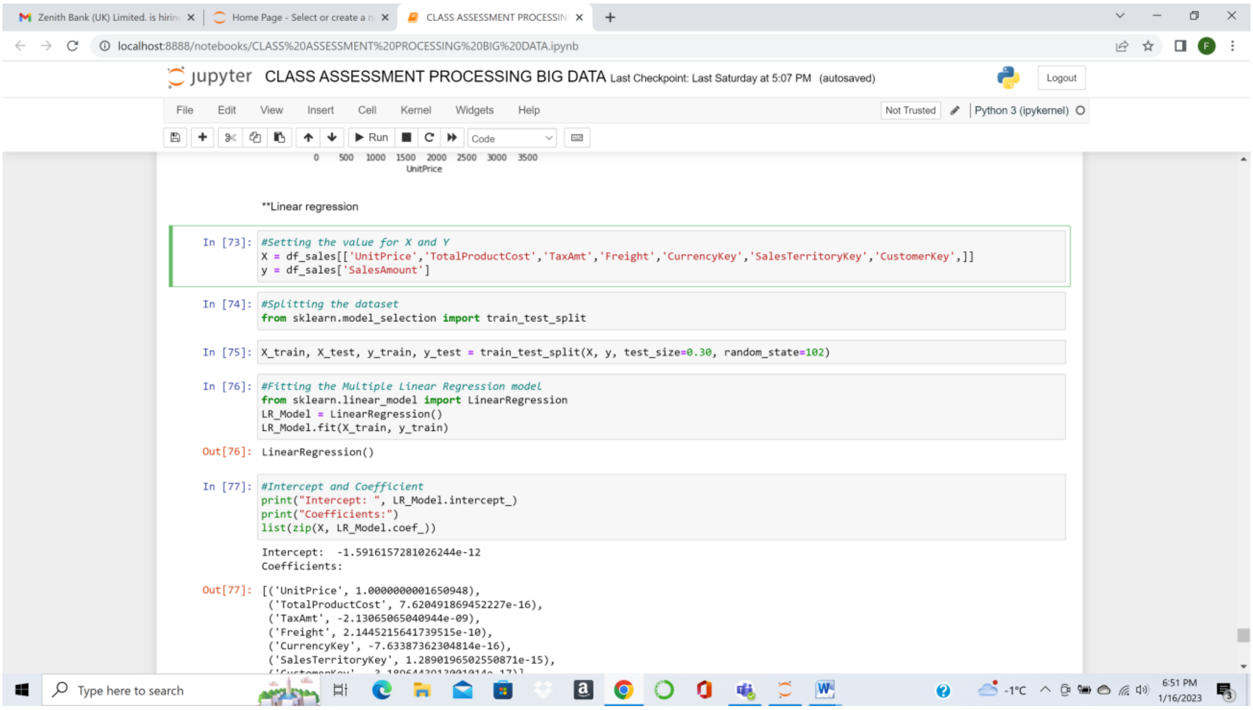
The model was also subjected to diagnostic analysis. From the results, the R-squared was 0.96 indicating that 96% variation in the dependent variable 'Calls' was explained by the model. It aslo indicates a high explanatory ability of the model as evidenced by the comparism of the Actual values with the predicted values. This result is shown below:



A second model of multiple regression analysis was constructed using dataset3. This is to compare the results with the outcome of the model with Dataset 2.

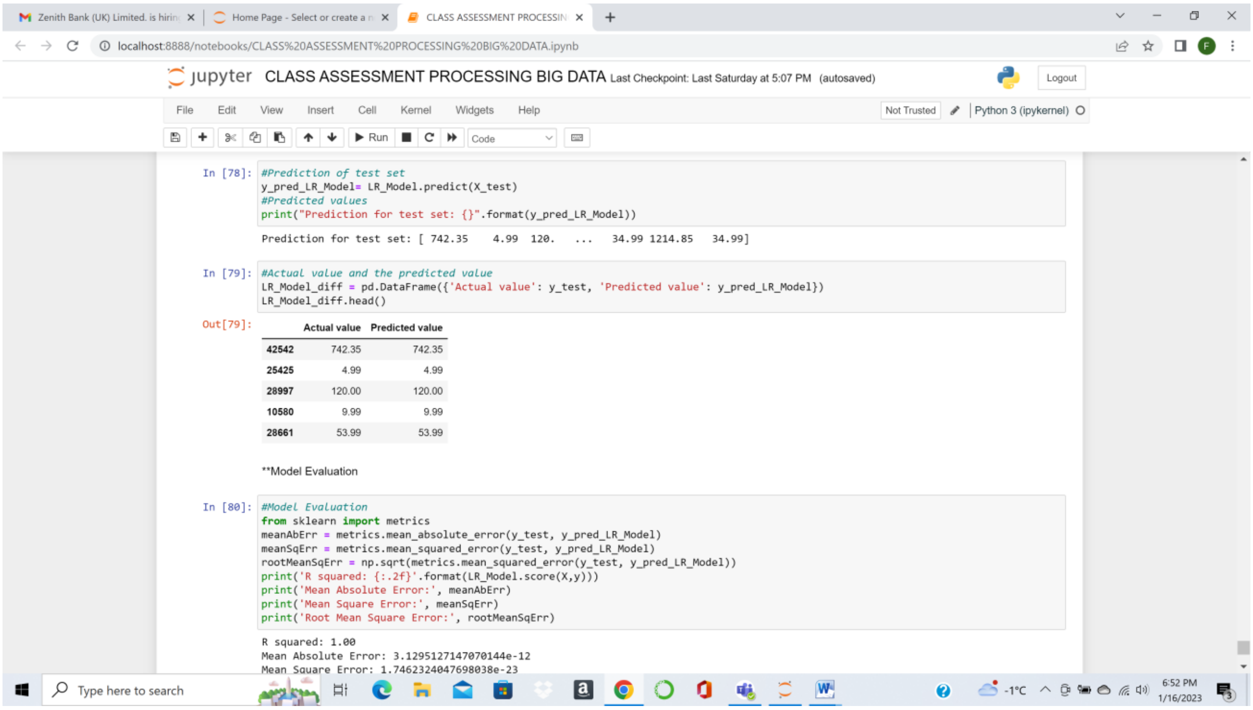
In dataset3, the objective was to estimate a model explaining the determinants of 'SalesAmount' and to make prediction accordingly. Thus, the dependent variable is 'SalesAmount' while the independent variables are 'UnitPrice', 'TotalProductCost', 'TaxAmt','Freight', 'CurrencyKey', 'SalesTerritoryKey' and 'CustomerKey'.

The technique adopted in analyzing dataset2 was also adopted and below are the major results:



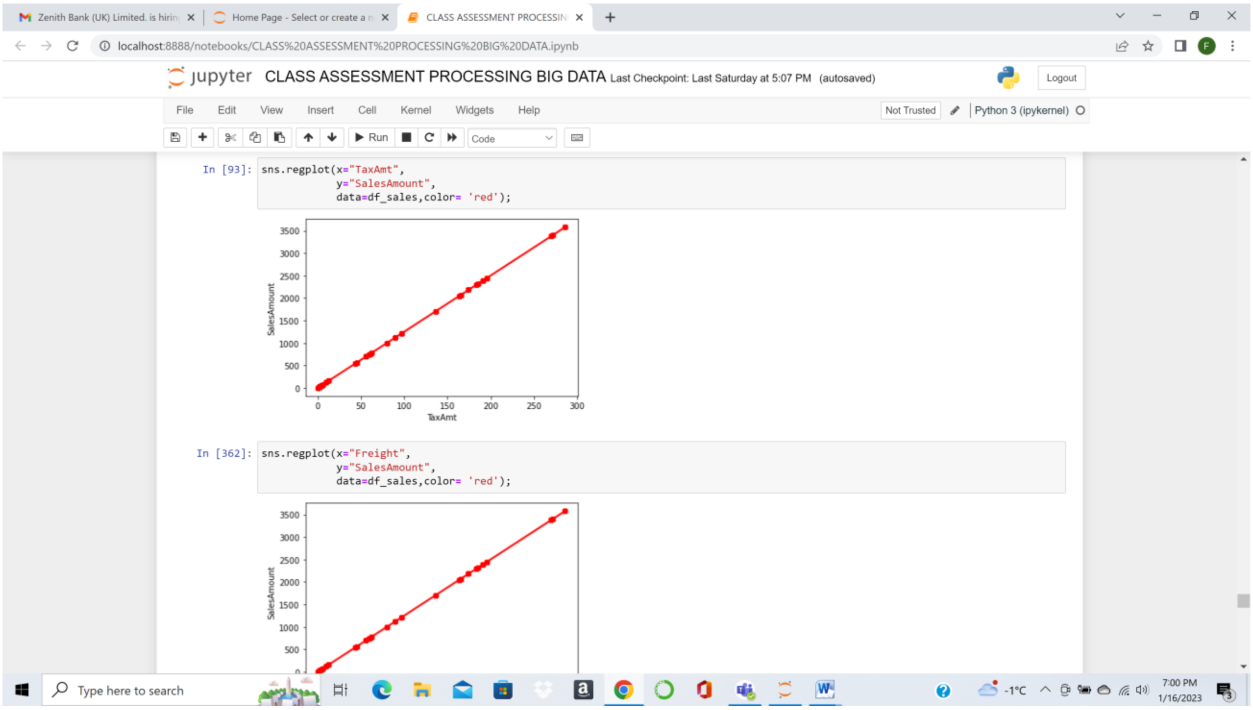
The estimated coefficients have mixed signs showing that while some variables are positively related to the dependent variable, others are negatively related to it. For instance, TotalProductcost with a coefficient of 7.62 implies that, a £1 increase in TotalProductcost will result in £7.62 increase in SalesAmount, holding all other factors constant. On the other hand, TaxAmt with a coefficient of -2.13 shows a negative relationship with the dependent variable.

The result of the prediction and model diagnostic analysis is given below:



The result of the multiple regression analysis above showed a very impressive outcome. The prediction was perfect with R-squared of 100%. This is evident as the predicted values are the same as the actual values.

It should be noted however, that the reason for the perfect prediction could be attributed to the perfect correlation between the dependent variable and the independent variables. The scatter plots showed the points lying on the line in almost all the cases as presented below:



Prediction summary

The aim of this section was to demonstrate the use of Machine learning algorithm to make prediction regarding future values of variables so as to aid planning and decision making. Multiple linear regression model was adopted over other models such as non-linear models based on the properties of the dataset visa-vis the key assumptions of the model. EDA on the datasets revealed a linear relationship between the dependent variable and the independent variables.

Two datasets were adopted for this analysis for comparism purposes. Evidence from the two datasets suggests strong predictive ability of the models. The first model has a predictive ability of 96% while the second model has predictive ability of 100%.

4. CONCLUSION

4.1 Summarisation

This study was designed to express the learning outcomes of the course Processing Big Data-7C5516 and was divided into two sections. Section one was the evaluation of Cloud big data processing platforms while section two was the use of Python programming language to analyse three big datasets.

In section one, 4 big data supported cloud platforms namely BigQuery, Azure, Keboola and Red Hat OpenShift were evaluated. Eight cloud data warehouse evaluation criteria were used for the evaluation. These criteria covers certain categories such as Performance at scale, Elasticity, Ease of Use, Cost Efficiency, Ability to support structured and semi-structured data, etc. The features of these platforms in terms of whether they offer SaaS, IaaS or PaaS were also examined. It was discovered in general that cloud platforms such as Bigquery and Azure are more robust than Keboola and Red Hat OpenShift. However, every platform has some area of strength which could distinguish it from others.

However, when choosing a platform for any business entity, the decision has to be more of a holistic one in the face of technical constraints as well as cost implications. In considering the features and functionality variations among the platforms, an optimal solution has to be implemented so as to avoid sub-optimal tradeoff between benefits and cost which will impinge on customer satisfaction. A good option could also be an integrated approach where certain platforms are chosen based on certain area of strength and a combination of platforms could then be made.

In section two, analysis was done on 3 datasets to demonstrate EDA, Classification and Prediction. In EDA, the datasets were examined for patterns, missing values and statistical properties/structures. They were also visualized by the use of charts and graphs. These enabled the understanding of the data and its evaluation so accessing its conformity with a-priori model assumptions.

Classification was done using Logistic regression. The result showed 73% accuracy which was impressive. Multiple Linear Regression model was employed for prediction using two separate datasets. The results were showed good fit, with R-Square of 96% for the first model and R-square of 100% for the second model. Thus, the predictive ability of the models was very high.

4.2 Experience Discussion

This assessment has helped me in a number of ways which I could enumerate as follows:

i. Through this assessment, I was able to research many cloud platforms. This enhanced my knowledge in cloud and data warehouse technologies. To a large extent, I now understand the various key cloud data warehouse platforms and how to evaluate them so as to make optimal choice when implementing a cloud data warehouse.
ii. With this knowledge and experience, I have acquired improved skills which will help me serve as a very good data cloud consultant to different businesses or organization. This is because I can give a well-informed advice to them in this area which will help them maximize their organizational goals.
iii. The data analysis exercise which I did with Python has deepened my knowledge of coding with Python on the Jupiter platform. My knowledge of Python codes has improved.
iv. It has also improved my skills on EDA and data visualization.
v. I have also learnt Machine learning algorithm in the area of Classification and Prediction. These skills will help me perform well as a seasoned Data Analyst when I finish my studies.

4.3 Future work

I suggest future work should be on relational database and NOSQL (including SQL/Python, NoSQL/Python integration such as MongoDB). A lot of companies still use them so it will be good to have a practical project on this.