Airline Passenger Satisfaction

7082CEM Coursework

BIG DATA MANAGEMENT & DATA VISUALIZATION

Student Name:GLEN T JOHN

Stident ID : 11980816

Index

1.Introduction

The dataset that I have used for this coursework is downloaded from Kaggle. This dataset **‘Airline passenger satisfaction’** contains information about different factors a passengers satisfaction for example seat comfort , leg space ,food and drinks .The dataset has 25 columns and 25976 rows so it is ideal for big data analysis.

The analysis that will be done will be using Pyspark for machine learning analysis .Kmeans clustering and linear regression would be performed on the dataset. Algorithms that can predict outcomes and learn from data are designed and studied as part of machine learning. This project discovers and selects appropriate analytical techniques, focuses on an analysis of huge data, and explains the findings. .Exploratory Data analysis would be performed using Tableau .

In nutshell I would be analyzing satisfaction of passengers for different factors .

2.Background & Implementation

2.1 Background study

2.1.1. Spark

Apache Spark is a piece of software used for distributed data processing. It has a lot of data processing capacity. The SPARK Application Programming Interface supports Python, R, Scala, and Java (API). Higher-level libraries for SQL queries, graph processing, and machine learning are also included in Spark. Because of its libraries, Python was chosen for this study, and Pyspark—a mixture of Python and Spark—was also used.

2.1.2. Pyspark

Python programme can be executed using the capabilities of Apache Spark using the PySpark Python Spark library. The Scala-based Apache Spark library was created. A new tool named Pyspark was created to support Spark with Python. In comparison to conventional systems, PySpark applications operate 100 times faster. Python's extensive library set makes it the language of choice for most data scientists and analysts today. For them, integrating Python and Spark is quite advantageous.

2.1.3 Hadoop

Large datasets are processed using Hadoop, an Apache open-source framework developed in Java. An application built with the Hadoop framework operates in a setting that supports distributed computing and storage over a cluster of machines. Each giving local compute and storage, it can scale from one machine to thousands.

2.1.4 Tableau

Tableau is an interactive data visualization application that enables you to create relevant and interactive dashboards and spreadsheets to acquire business insights for better business development. It has several uses, including complex visualization, quick analytics, simple sharing, and interactive presentations.

2.1.5 Jupyter notebook

Documents with live code, mathematics, graphics, and text can be created and shared using Jupyter Notebook. Machine learning, data cleaning and transformation, statistical modelling, data visualization, and many other applications are all feasible.

2.2 Software Installation

2.2.1 Tableau

The tool Tableau Software makes it possible for users to see and comprehend data. From the official Tableau website, this programme can be downloaded.

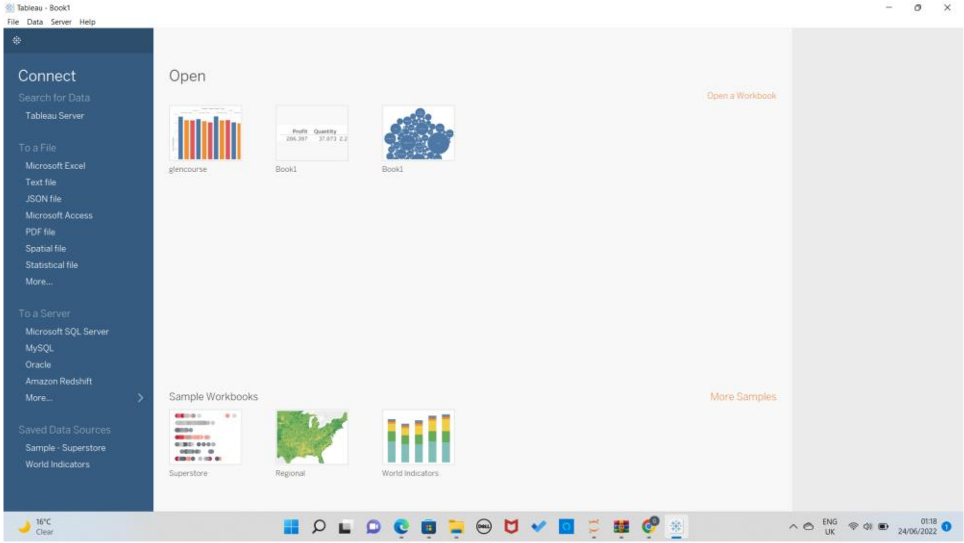


Fig-1 Tableau Screenshot

2.2.2 Pyspark

For downloading pyspark ,I had to download Apache spark, anaconda distribution , hadoop and jdk version 8.Higher version of jdk does not support apache spark.Further putting the environment variables as shown in the figure.

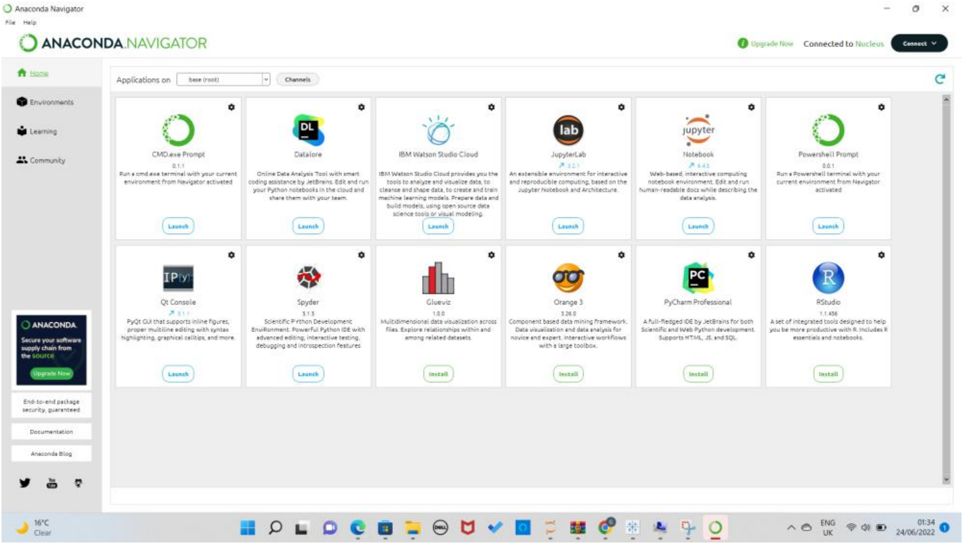
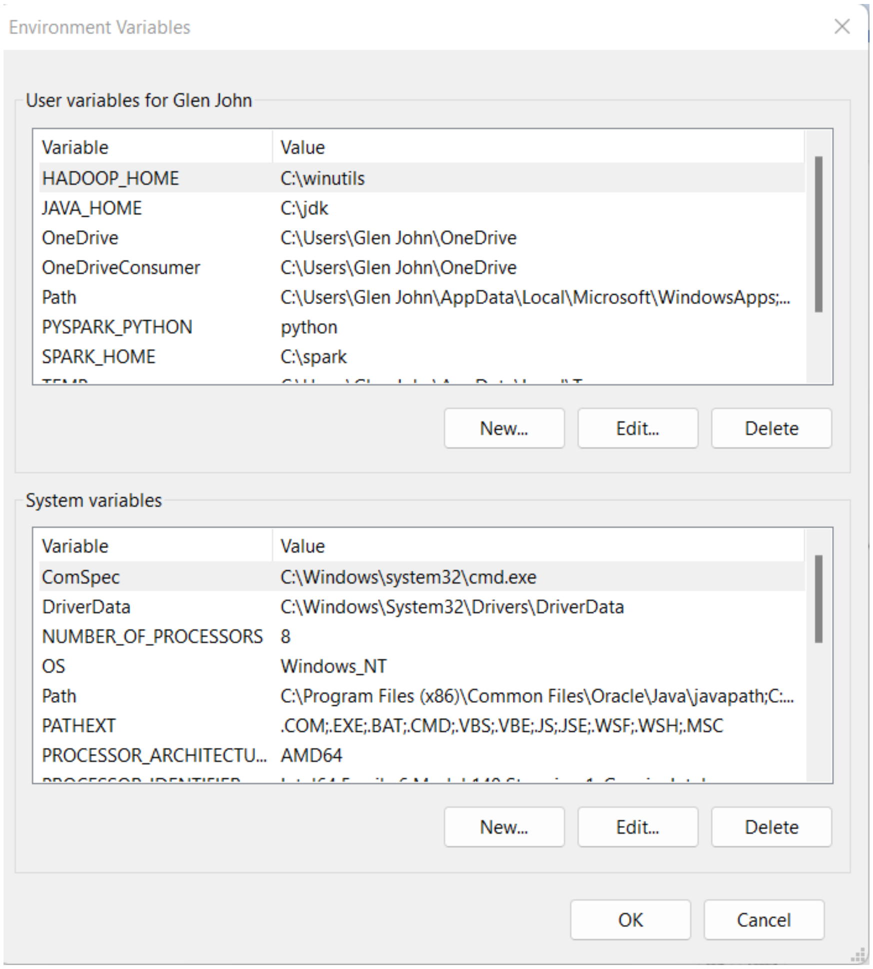


Fig 2 - Anaconda Navigator



User variables for Glen John

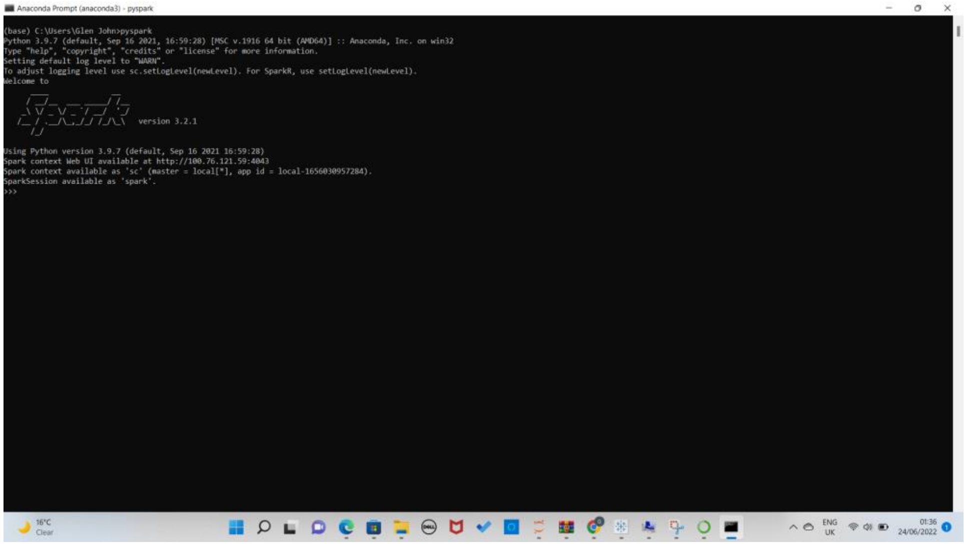
Variable	Value
HADOOP_HOME	C:\winutils
JAVA_HOME	C:\jdk
OneDrive	C:\Users\Glen John\OneDrive
OneDriveConsumer	C:\Users\Glen John\OneDrive
Path	C:\Users\Glen John\AppData\Local\Microsoft\WindowsApps;...
PYSPARK_PYTHON	python
SPARK_HOME	C:\spark

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTU...	AMD64

Fig –3 Setting environment variables

Running Pyspark command in anaconda prompt.

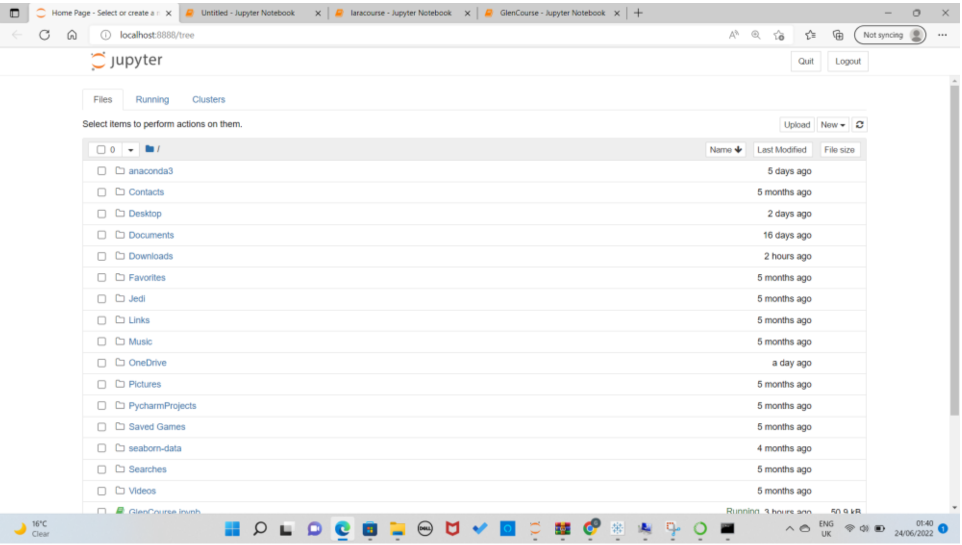


```
(base) C:\Users\Glen John>pyspark
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Python version 3.9.7 (default, Sep 16 2021 16:59:28)
Spark context Web UI available at http://100.76.121.59:4043
Spark context available as 'sc' (master = local[*], app id = local-1656030957284).
SparkSession available as 'spark'.
>>>
```

Fig-4 anaconda prompt for running pyspark

2.2.3 Jupyter Notebook

Jupyter notebook can be launched from anaconda navigator.This is what it will look


Home Page - Select or create a n
Untitled - Jupyter Notebook
laracourse - Jupyter Notebook
GlenCourse - Jupyter Notebook
localhost:8888/tree
Not syncing
jupyter
Quit
Logout
Files
Running
Clusters
Upload
New
16°C
Clear
ENG
UK
01:40
24/06/2022

Select items to perform actions on them.

☐ 0 /	Name	Last Modified	File size
☐ anaconda3		5 days ago	
☐ Contacts		5 months ago	
☐ Desktop		2 days ago	
☐ Documents		16 days ago	
☐ Downloads		2 hours ago	
☐ Favorites		5 months ago	
☐ Jedi		5 months ago	
☐ Links		5 months ago	
☐ Music		5 months ago	
☐ OneDrive		a day ago	
☐ Pictures		5 months ago	
☐ PycharmProjects		5 months ago	
☐ Saved Games		5 months ago	
☐ seaborn-data		4 months ago	
☐ Searches		5 months ago	
☐ Videos		5 months ago	
☐ GlenCourse.ipynb	Running	3 hours ago	50.9 kB

Fig- 5 Jupyter Notebook

2.3 Dataset

The Dataset that I have chosen for the data analysis is **‘Airline passenger satisfaction’ .**The dataset gives information about the passenger satisfaction using different factors which are seat comfort , food and drinks ,departure delay time ,arrival delay time ,Inflight entertainment etc. The dataset has 25 columns and has 25976 rows .

Gender: Gender of the passengers (Female, Male)

Customer Type: The customer type (Loyal customer, disloyal customer)

Age: The actual age of the passengers

Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)

Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Flight distance: The flight distance of this journey

Inflight Wi-Fi service: Satisfaction level of the inflight Wi-Fi service (0:Not Applicable;1-5)

Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient

Ease of Online booking: Satisfaction level of online booking

Gate location: Satisfaction level of Gate location

Food and drink: Satisfaction level of Food and drink

Online boarding: Satisfaction level of online boarding

Seat comfort: Satisfaction level of Seat comfort

Inflight entertainment: Satisfaction level of inflight entertainment

On-board service: Satisfaction level of On-board service

Leg room service: Satisfaction level of Leg room service

Baggage handling: Satisfaction level of baggage handling

Check-in service: Satisfaction level of Check-in service

Inflight service: Satisfaction level of inflight service

Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure

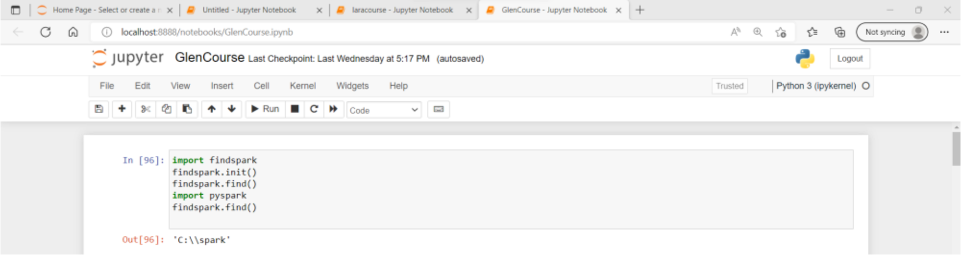
Arrival Delay in Minutes: Minutes delayed when Arrival

Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

2.4 Implementation

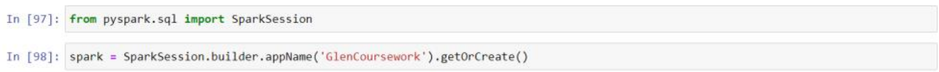
Importing Pyspark and Spark Session

To start pyspark in jupyter notebook I had to download findspark in anaconda prompt


Home Page - Select or create a
Untitled - Jupyter Notebook
laracourse - Jupyter Notebook
GlenCourse - Jupyter Notebook
localhost:8888/notebooks/GlenCourse.ipynb
Not syncing
jupyter
GlenCourse Last Checkpoint: Last Wednesday at 5:17 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [96]: import findspark
         findspark.init()
         findspark.find()
         import pyspark
         findspark.find()

Out[96]: 'C:\\spark'
```

Fig –6

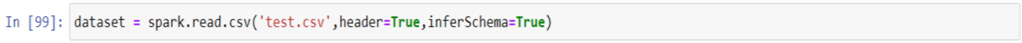


```
In [97]: from pyspark.sql import SparkSession

In [98]: spark = SparkSession.builder.appName('GlenCoursework').getOrCreate()
```

Fig-7

Loading of Dataset

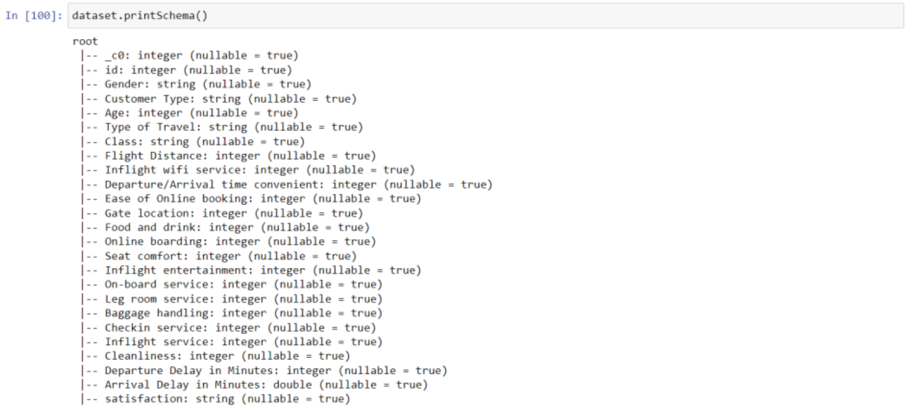
Loading of the dataset can be done with read.csv ,keeping the header true and inferSchema as true



```
In [99]: dataset = spark.read.csv('test.csv',header=True,inferSchema=True)
```

Fig –8

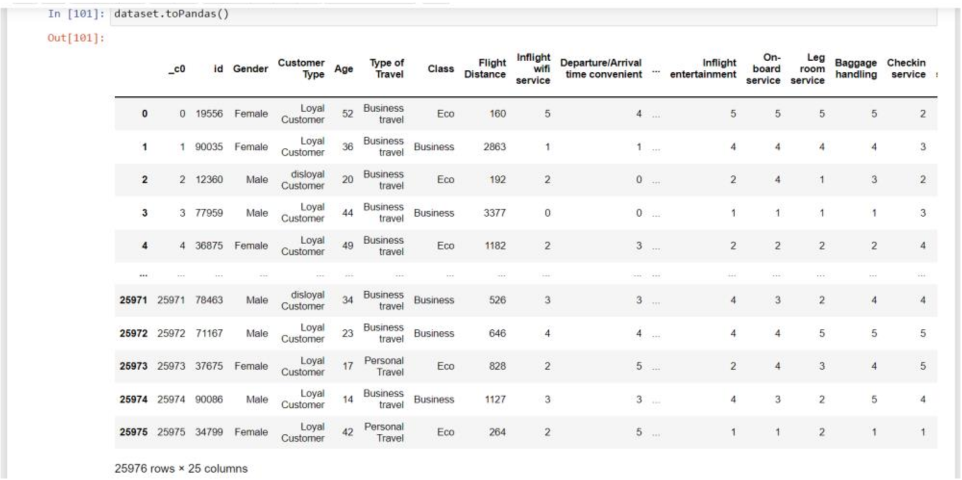
Checking the Schema of the Dataset



```
In [100]: dataset.printSchema()

root
 |-- _c0: integer (nullable = true)
 |-- id: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Customer Type: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Type of Travel: string (nullable = true)
 |-- Class: string (nullable = true)
 |-- Flight Distance: integer (nullable = true)
 |-- Inflight wifi service: integer (nullable = true)
 |-- Departure/Arrival time convenient: integer (nullable = true)
 |-- Ease of Online booking: integer (nullable = true)
 |-- Gate location: integer (nullable = true)
 |-- Food and drink: integer (nullable = true)
 |-- Online boarding: integer (nullable = true)
 |-- Seat comfort: integer (nullable = true)
 |-- Inflight entertainment: integer (nullable = true)
 |-- On-board service: integer (nullable = true)
 |-- Leg room service: integer (nullable = true)
 |-- Baggage handling: integer (nullable = true)
 |-- Checkin service: integer (nullable = true)
 |-- Inflight service: integer (nullable = true)
 |-- Cleanliness: integer (nullable = true)
 |-- Departure Delay in Minutes: integer (nullable = true)
 |-- Arrival Delay in Minutes: double (nullable = true)
 |-- satisfaction: string (nullable = true)
```

Fig-9

Loading the dataset



```
In [101]: dataset.toPandas()
Out[101]:
```

	_c0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service
0	0	19556	Female	Loyal Customer	52	Business travel	Eco	160	5	4	...	5	5	5	5	2
1	1	90035	Female	Loyal Customer	36	Business travel	Business	2863	1	1	...	4	4	4	4	3
2	2	12360	Male	disloyal Customer	20	Business travel	Eco	192	2	0	...	2	4	1	3	2
3	3	77959	Male	Loyal Customer	44	Business travel	Business	3377	0	0	...	1	1	1	1	3
4	4	36875	Female	Loyal Customer	49	Business travel	Eco	1182	2	3	...	2	2	2	2	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25971	25971	78463	Male	disloyal Customer	34	Business travel	Business	526	3	3	...	4	3	2	4	4
25972	25972	71167	Male	Loyal Customer	23	Business travel	Business	646	4	4	...	4	4	5	5	5
25973	25973	37675	Female	Loyal Customer	17	Personal Travel	Eco	828	2	5	...	2	4	3	4	5
25974	25974	90086	Male	Loyal Customer	14	Business travel	Business	1127	3	3	...	4	3	2	5	4
25975	25975	34799	Female	Loyal Customer	42	Personal Travel	Eco	264	2	5	...	1	1	2	1	1

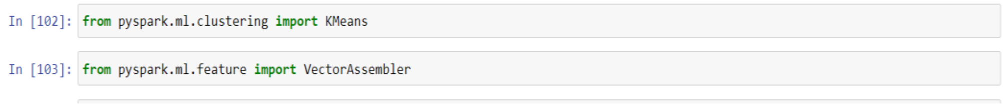
25976 rows × 25 columns

Fig-10

Kmeans Clustering

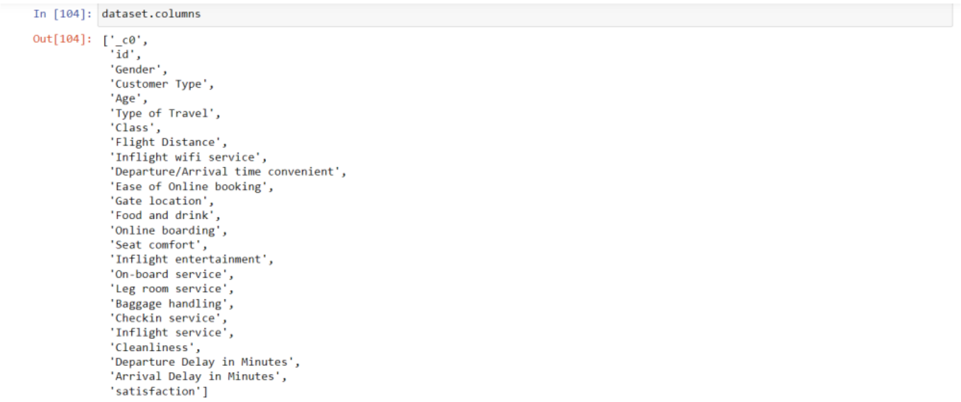
K-means clustering is one of the easiest and most popular unsupervised machine learning algorithms. A cluster is a collection of data elements that have been put together because they share certain traits. k, the desired number of centroids in the dataset, is the goal number. A centroid is the place that symbolises the cluster's actual or hypothetical centre. Each data point is sorted into a different cluster by reducing the in-cluster sum of squares. To put it another way, the K-means method minimises the size of the centroids while still finding k centroids and assigning each data point to the nearest cluster.

Import libraries for clustering



```
In [102]: from pyspark.ml.clustering import KMeans

In [103]: from pyspark.ml.feature import VectorAssembler
```

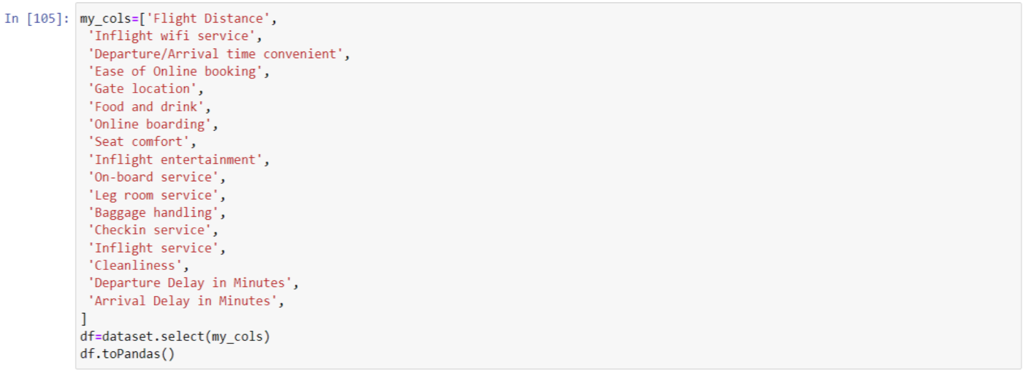
Fig-11



```
In [104]: dataset.columns

Out[104]: ['_c0',
           'id',
           'Gender',
           'Customer Type',
           'Age',
           'Type of Travel',
           'Class',
           'Flight Distance',
           'Inflight wifi service',
           'Departure/Arrival time convenient',
           'Ease of Online booking',
           'Gate location',
           'Food and drink',
           'Online boarding',
           'Seat comfort',
           'Inflight entertainment',
           'On-board service',
           'Leg room service',
           'Baggage handling',
           'Checkin service',
           'Inflight service',
           'Cleanliness',
           'Departure Delay in Minutes',
           'Arrival Delay in Minutes',
           'satisfaction']
```

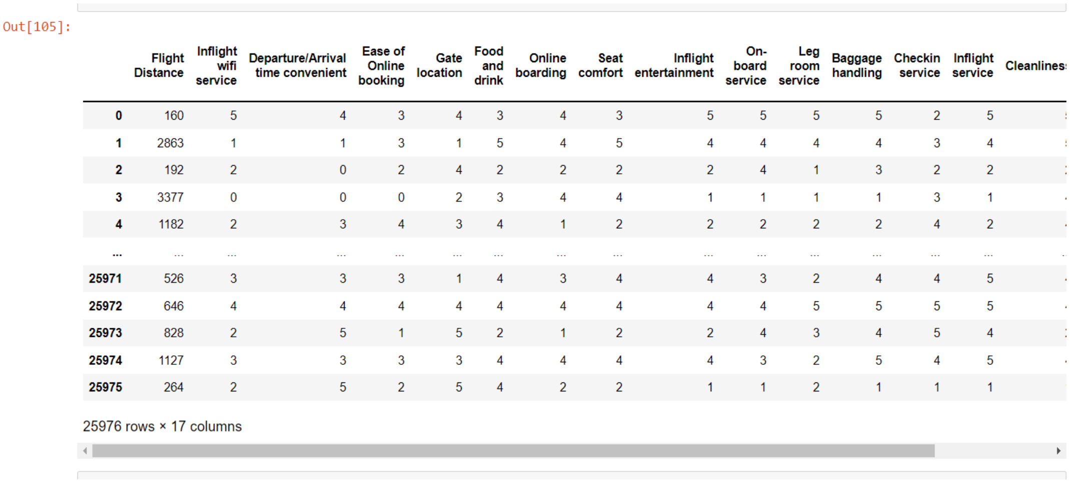
Fig-12

Necessary Columns to be selected for analysis of the dataset



```
In [105]: my_cols=['Flight Distance',
           'Inflight wifi service',
           'Departure/Arrival time convenient',
           'Ease of Online booking',
           'Gate location',
           'Food and drink',
           'Online boarding',
           'Seat comfort',
           'Inflight entertainment',
           'On-board service',
           'Leg room service',
           'Baggage handling',
           'Checkin service',
           'Inflight service',
           'Cleanliness',
           'Departure Delay in Minutes',
           'Arrival Delay in Minutes',
          ]
          df=dataset.select(my_cols)
          df.toPandas()
```

Fig-13



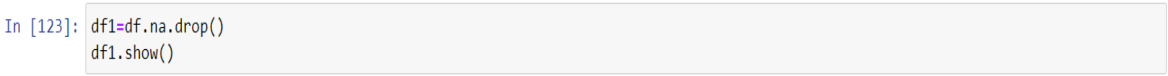
Out[105]:

	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checkin service	Inflight service	Cleanlines
0	160	5	4	3	4	3	4	3	5	5	5	5	2	5	
1	2863	1	1	3	1	5	4	5	4	4	4	4	3	4	
2	192	2	0	2	4	2	2	2	2	4	1	3	2	2	
3	3377	0	0	0	2	3	4	4	1	1	1	1	3	1	
4	1182	2	3	4	3	4	1	2	2	2	2	2	4	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25971	526	3	3	3	1	4	3	4	4	3	2	4	4	5	
25972	646	4	4	4	4	4	4	4	4	4	5	5	5	5	
25973	828	2	5	1	5	2	1	2	2	4	3	4	5	4	
25974	1127	3	3	3	3	4	4	4	4	3	2	5	4	5	
25975	264	2	5	2	5	4	2	2	1	1	2	1	1	1	

25976 rows × 17 columns

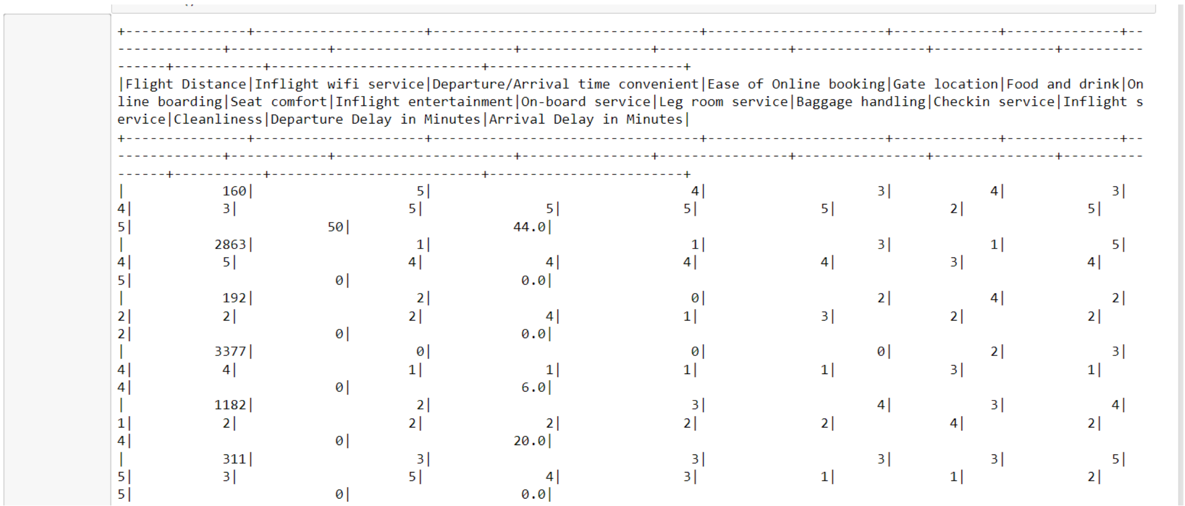
Fig-14

Dropping Unwanted Data



In [123]:
```
df1=df.na.drop()
df1.show()
```

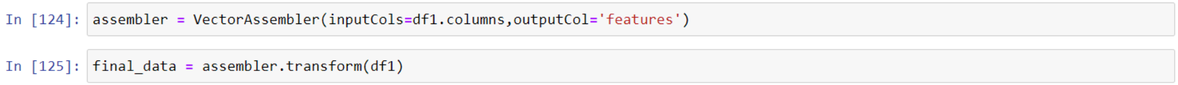
Fig-15



```
+---------------+---------------------+---------------------------------+----------------------+-------------+--------------+--
-------------+------------+----------------------+----------------+----------------+----------------+---------------+----------
------+-----------+--------------------------+------------------------+
|Flight Distance|Inflight wifi service|Departure/Arrival time convenient|Ease of Online booking|Gate location|Food and drink|On
line boarding|Seat comfort|Inflight entertainment|On-board service|Leg room service|Baggage handling|Checkin service|Inflight s
ervice|Cleanliness|Departure Delay in Minutes|Arrival Delay in Minutes|
+---------------+---------------------+---------------------------------+----------------------+-------------+--------------+--
-------------+------------+----------------------+----------------+----------------+----------------+---------------+----------
------+-----------+--------------------------+------------------------+
|            160|                    5|                                4|                     3|            4|             3|
4|           3|                     5|               5|               5|               5|              2|               5|
5|                        50|                    44.0|
|           2863|                    1|                                1|                     3|            1|             5|
4|           5|                     4|               4|               4|               4|              3|               4|
5|                         0|                     0.0|
|            192|                    2|                                0|                     2|            4|             2|
2|           2|                     2|               4|               1|               3|              2|               2|
2|                         0|                     0.0|
|           3377|                    0|                                0|                     0|            2|             3|
4|           4|                     1|               1|               1|               1|              3|               1|
4|                         0|                     6.0|
|           1182|                    2|                                3|                     4|            3|             4|
1|           2|                     2|               2|               2|               2|              4|               2|
4|                         0|                    20.0|
|            311|                    3|                                3|                     3|            3|             5|
5|           3|                     5|               4|               3|               1|              1|               2|
5|                         0|                     0.0|
```

Fig-16

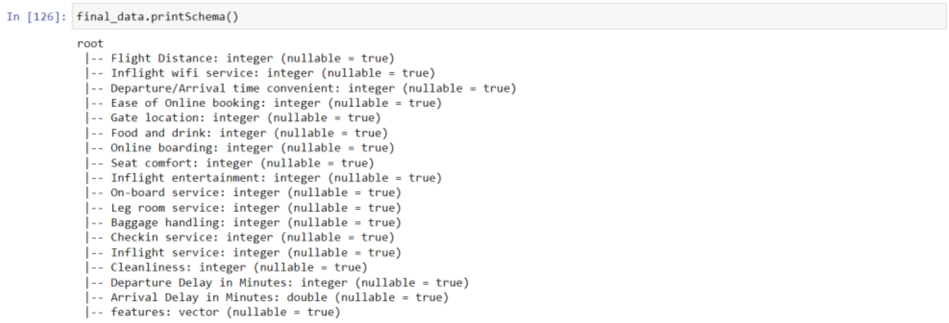
Creating a vector column with the help of multiple columns and transforming the data



```
In [124]: assembler = VectorAssembler(inputCols=df1.columns,outputCol='features')

In [125]: final_data = assembler.transform(df1)
```

Fig-17

Printing the Schema of the final data

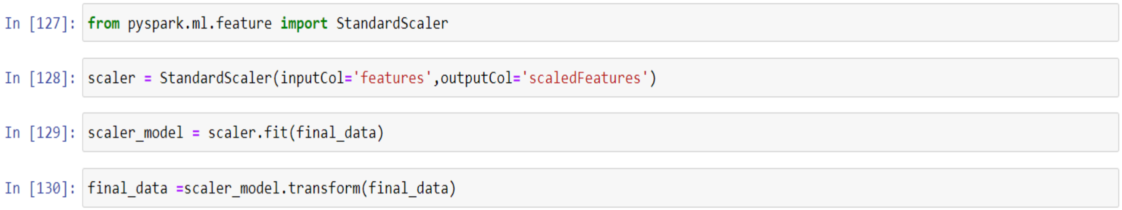


```
In [126]: final_data.printSchema()

          root
           |-- Flight Distance: integer (nullable = true)
           |-- Inflight wifi service: integer (nullable = true)
           |-- Departure/Arrival time convenient: integer (nullable = true)
           |-- Ease of Online booking: integer (nullable = true)
           |-- Gate location: integer (nullable = true)
           |-- Food and drink: integer (nullable = true)
           |-- Online boarding: integer (nullable = true)
           |-- Seat comfort: integer (nullable = true)
           |-- Inflight entertainment: integer (nullable = true)
           |-- On-board service: integer (nullable = true)
           |-- Leg room service: integer (nullable = true)
           |-- Baggage handling: integer (nullable = true)
           |-- Checkin service: integer (nullable = true)
           |-- Inflight service: integer (nullable = true)
           |-- Cleanliness: integer (nullable = true)
           |-- Departure Delay in Minutes: integer (nullable = true)
           |-- Arrival Delay in Minutes: double (nullable = true)
           |-- features: vector (nullable = true)
```

Fig-18

As shown in fig-18 ,a features column has been created

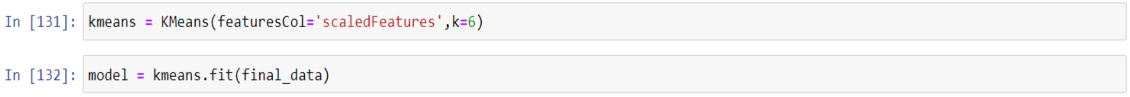
Next ,is scaling of the data and then fitting of the data into a scaler model



```
In [127]: from pyspark.ml.feature import StandardScaler

In [128]: scaler = StandardScaler(inputCol='features',outputCol='scaledFeatures')

In [129]: scaler_model = scaler.fit(final_data)

In [130]: final_data =scaler_model.transform(final_data)
```

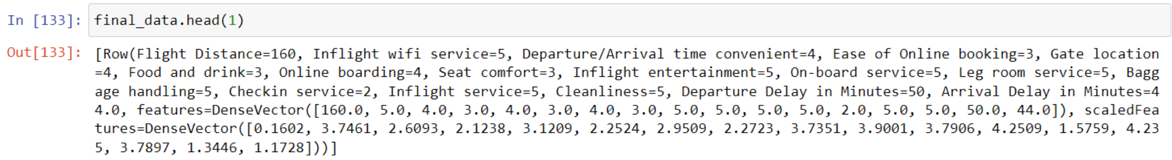
Fig –19

Assigning the K value and then fitting into the model



```
In [131]: kmeans = KMeans(featuresCol='scaledFeatures',k=6)

In [132]: model = kmeans.fit(final_data)
```

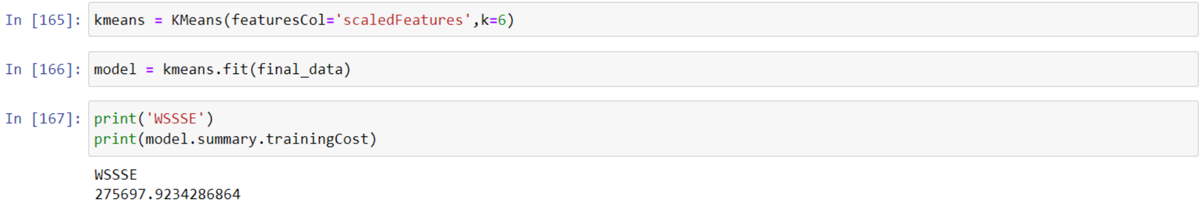
Fig-20



```
In [133]: final_data.head(1)

Out[133]: [Row(Flight Distance=160, Inflight wifi service=5, Departure/Arrival time convenient=4, Ease of Online booking=3, Gate location
          =4, Food and drink=3, Online boarding=4, Seat comfort=3, Inflight entertainment=5, On-board service=5, Leg room service=5, Bagg
          age handling=5, Checkin service=2, Inflight service=5, Cleanliness=5, Departure Delay in Minutes=50, Arrival Delay in Minutes=4
          4.0, features=DenseVector([160.0, 5.0, 4.0, 3.0, 4.0, 3.0, 4.0, 3.0, 5.0, 5.0, 5.0, 5.0, 2.0, 5.0, 5.0, 50.0, 44.0]), scaledFea
          tures=DenseVector([0.1602, 3.7461, 2.6093, 2.1238, 3.1209, 2.2524, 2.9509, 2.2723, 3.7351, 3.9001, 3.7906, 4.2509, 1.5759, 4.23
          5, 3.7897, 1.3446, 1.1728]))]
```

Fig-21

As shown in the output ,a scaled features columns(Dense Vector) is been created ,where the features column has been scaled.

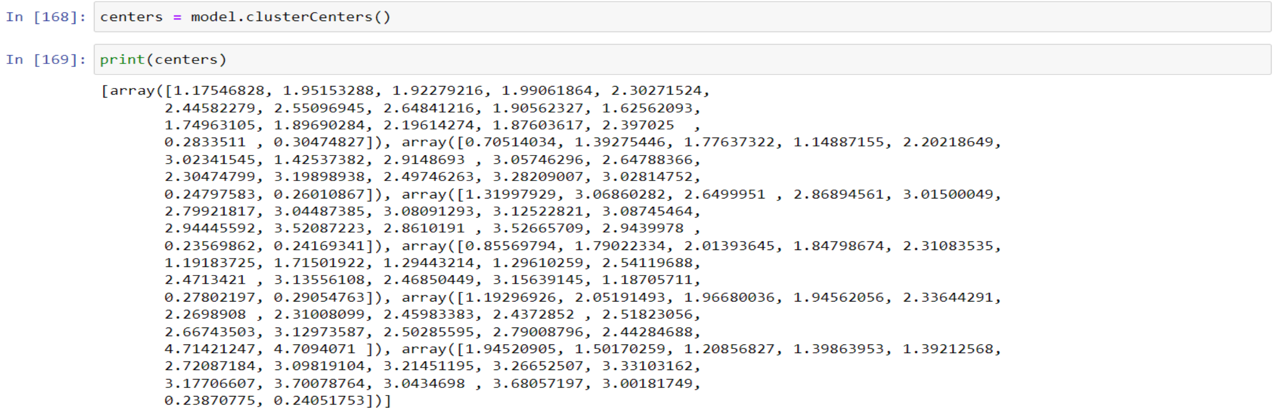


```
In [165]: kmeans = KMeans(featuresCol='scaledFeatures',k=6)

In [166]: model = kmeans.fit(final_data)

In [167]: print('WSSSE')
          print(model.summary.trainingCost)

          WSSSE
          275697.9234286864
```

Fig-22

Fig-22 shows the sum of set squares as 275697 ,which is very high .So that means the K value has to be increased .The points are very far away from the centroid

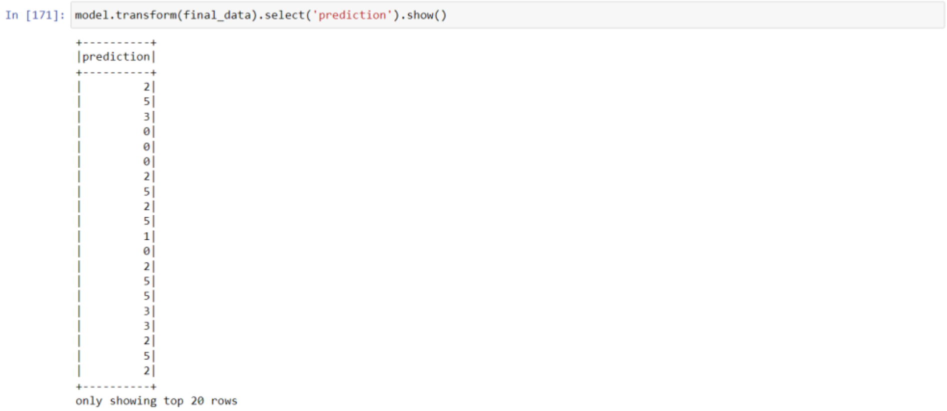
Printing the centroids(centers)



```
In [168]: centers = model.clusterCenters()

In [169]: print(centers)

          [array([1.17546828, 1.95153288, 1.92279216, 1.99061864, 2.30271524,
                 2.44582279, 2.55096945, 2.64841216, 1.90562327, 1.62562093,
                 1.74963105, 1.89690284, 2.19614274, 1.87603617, 2.397025  ,
                 0.2833511 , 0.30474827]), array([0.70514034, 1.39275446, 1.77637322, 1.14887155, 2.20218649,
                 3.02341545, 1.42537382, 2.9148693 , 3.05746296, 2.64788366,
                 2.30474799, 3.19898938, 2.49746263, 3.28209007, 3.02814752,
                 0.24797583, 0.26010867]), array([1.31997929, 3.06860282, 2.6499951 , 2.86894561, 3.01500049,
                 2.79921817, 3.04487385, 3.08091293, 3.12522821, 3.08745464,
                 2.94445592, 3.52087223, 2.8610191 , 3.52665709, 2.9439978 ,
                 0.23569862, 0.24169341]), array([0.85569794, 1.79022334, 2.01393645, 1.84798674, 2.31083535,
                 1.19183725, 1.71501922, 1.29443214, 1.29610259, 2.54119688,
                 2.4713421 , 3.13556108, 2.46850449, 3.15639145, 1.18705711,
                 0.27802197, 0.29054763]), array([1.19296926, 2.05191493, 1.96680036, 1.94562056, 2.33644291,
                 2.2698908 , 2.31008099, 2.45983383, 2.4372852 , 2.51823056,
                 2.66743503, 3.12973587, 2.50285595, 2.79008796, 2.44284688,
                 4.71421247, 4.7094071 ]), array([1.94520905, 1.50170259, 1.20856827, 1.39863953, 1.39212568,
                 2.72087184, 3.09819104, 3.21451195, 3.26652507, 3.33103162,
                 3.17706607, 3.70078764, 3.0434698 , 3.68057197, 3.00181749,
                 0.23870775, 0.24051753])]
```

Fig-23

As shown in fig-23 ,there has been created 6 arrays of different points.



```
In [171]: model.transform(final_data).select('prediction').show()

+----------+
|prediction|
+----------+
|         2|
|         5|
|         3|
|         0|
|         0|
|         0|
|         2|
|         5|
|         2|
|         5|
|         1|
|         0|
|         2|
|         5|
|         5|
|         3|
|         3|
|         2|
|         5|
|         2|
+----------+
only showing top 20 rows
```

Fig-24

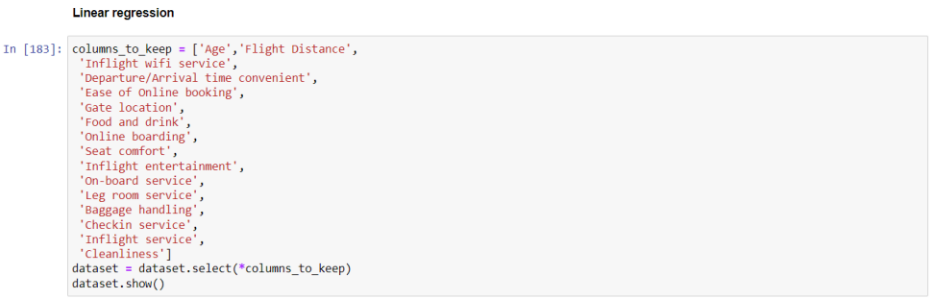
Fig-24 shows the prediction column ,each point is clustered with the respective clusters 0,1,2,3,4,5.

Linear Regression

Linear regression attempts to predict the relationship between two variables by fitting a linear equation to the observed data. While the second variable is viewed as a dependent variable, the first is considered an explanatory variable. A modeller might, for instance, compare people's weights to their heights using a linear regression model.

For the analysis I have used factors of the passenger satisfaction versus the age.

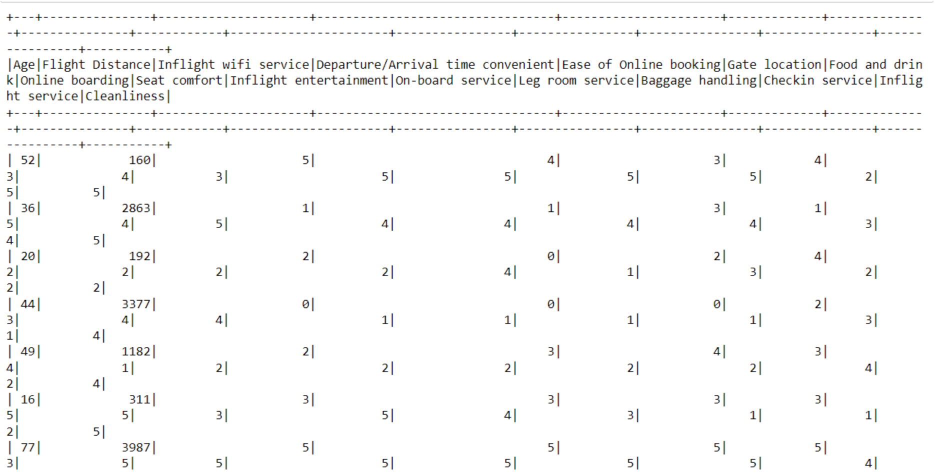
Following Columns to keep



Linear regression

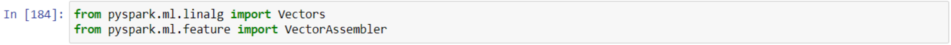
```
In [183]: columns_to_keep = ['Age','Flight Distance',
           'Inflight wifi service',
           'Departure/Arrival time convenient',
           'Ease of Online booking',
           'Gate location',
           'Food and drink',
           'Online boarding',
           'Seat comfort',
           'Inflight entertainment',
           'On-board service',
           'Leg room service',
           'Baggage handling',
           'Checkin service',
           'Inflight service',
           'Cleanliness']
          dataset = dataset.select(*columns_to_keep)
          dataset.show()
```

Fig-25



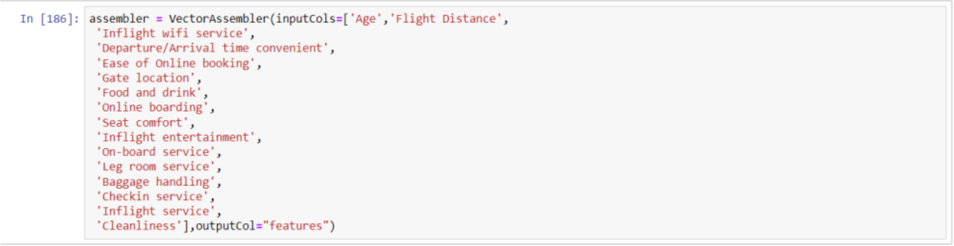
```
+---+---------------+---------------------+---------------------------------+----------------------+-------------+-------------
-+---------------+------------+----------------------+----------------+----------------+----------------+---------------+------
----------+-----------+
|Age|Flight Distance|Inflight wifi service|Departure/Arrival time convenient|Ease of Online booking|Gate location|Food and drin
k|Online boarding|Seat comfort|Inflight entertainment|On-board service|Leg room service|Baggage handling|Checkin service|Inflig
ht service|Cleanliness|
+---+---------------+---------------------+---------------------------------+----------------------+-------------+-------------
-+---------------+------------+----------------------+----------------+----------------+----------------+---------------+------
----------+-----------+
| 52|            160|                    5|                                4|                     3|            4|
3|              4|           3|                     5|               5|               5|               5|              2|
5|          5|
| 36|           2863|                    1|                                1|                     3|            1|
5|              4|           5|                     4|               4|               4|               4|              3|
4|          5|
| 20|            192|                    2|                                0|                     2|            4|
2|              2|           2|                     2|               4|               1|               3|              2|
2|          2|
| 44|           3377|                    0|                                0|                     0|            2|
3|              4|           4|                     1|               1|               1|               1|              3|
1|          4|
| 49|           1182|                    2|                                3|                     4|            3|
4|              1|           2|                     2|               2|               2|               2|              4|
2|          4|
| 16|            311|                    3|                                3|                     3|            3|
5|              5|           3|                     5|               4|               3|               1|              1|
2|          5|
| 77|           3987|                    5|                                5|                     5|            5|
3|              5|           5|                     5|               5|               5|               5|              4|
```

Fig-26



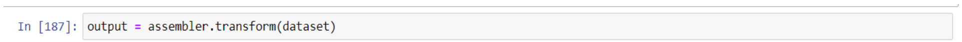
```
In [184]: from pyspark.ml.linalg import Vectors
          from pyspark.ml.feature import VectorAssembler
```

Fig-27



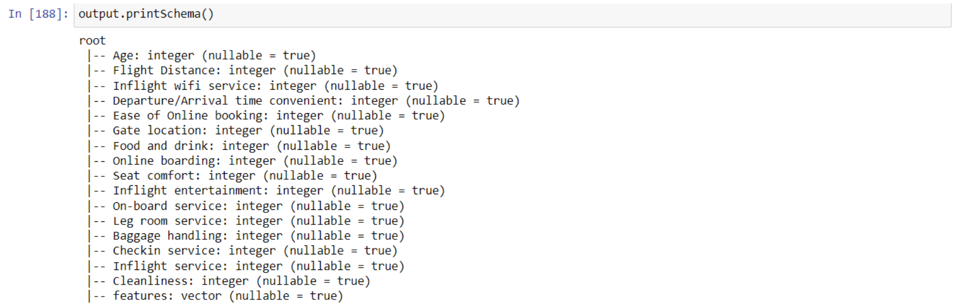
```
In [186]: assembler = VectorAssembler(inputCols=['Age','Flight Distance',
           'Inflight wifi service',
           'Departure/Arrival time convenient',
           'Ease of Online booking',
           'Gate location',
           'Food and drink',
           'Online boarding',
           'Seat comfort',
           'Inflight entertainment',
           'On-board service',
           'Leg room service',
           'Baggage handling',
           'Checkin service',
           'Inflight service',
           'Cleanliness'],outputCol="features")
```

Fig-28



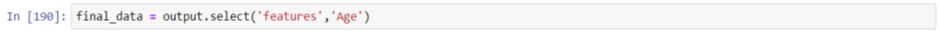
```
In [187]: output = assembler.transform(dataset)
```

Fig-29



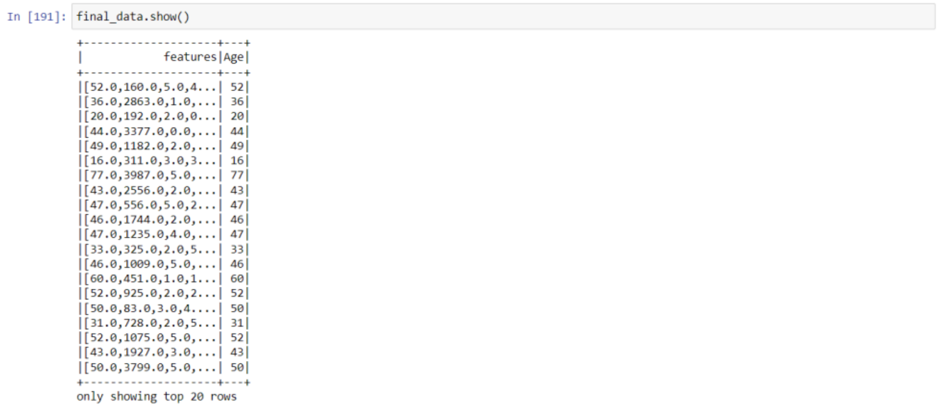
```
In [188]: output.printSchema()

root
 |-- Age: integer (nullable = true)
 |-- Flight Distance: integer (nullable = true)
 |-- Inflight wifi service: integer (nullable = true)
 |-- Departure/Arrival time convenient: integer (nullable = true)
 |-- Ease of Online booking: integer (nullable = true)
 |-- Gate location: integer (nullable = true)
 |-- Food and drink: integer (nullable = true)
 |-- Online boarding: integer (nullable = true)
 |-- Seat comfort: integer (nullable = true)
 |-- Inflight entertainment: integer (nullable = true)
 |-- On-board service: integer (nullable = true)
 |-- Leg room service: integer (nullable = true)
 |-- Baggage handling: integer (nullable = true)
 |-- Checkin service: integer (nullable = true)
 |-- Inflight service: integer (nullable = true)
 |-- Cleanliness: integer (nullable = true)
 |-- features: vector (nullable = true)
```

Fig30



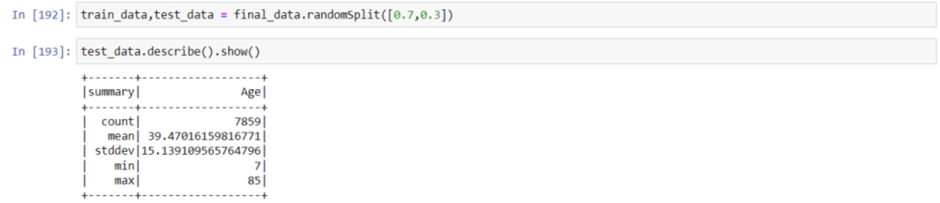
```
In [190]: final_data = output.select('features','Age')
```

Fig-31



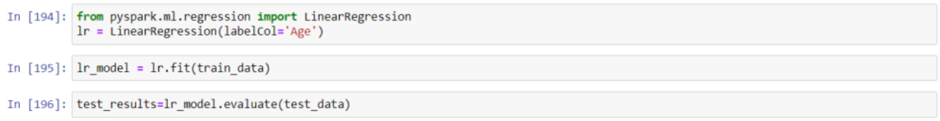
```
In [191]: final_data.show()

+--------------------+---+
|            features|Age|
+--------------------+---+
|[52.0,160.0,5.0,4...| 52|
|[36.0,2863.0,1.0,...| 36|
|[20.0,192.0,2.0,0...| 20|
|[44.0,3377.0,0.0,...| 44|
|[49.0,1182.0,2.0,...| 49|
|[16.0,311.0,3.0,3...| 16|
|[77.0,3987.0,5.0,...| 77|
|[43.0,2556.0,2.0,...| 43|
|[47.0,556.0,5.0,2...| 47|
|[46.0,1744.0,2.0,...| 46|
|[47.0,1235.0,4.0,...| 47|
|[33.0,325.0,2.0,5...| 33|
|[46.0,1009.0,5.0,...| 46|
|[60.0,451.0,1.0,1...| 60|
|[52.0,925.0,2.0,2...| 52|
|[50.0,83.0,3.0,4....| 50|
|[31.0,728.0,2.0,5...| 31|
|[52.0,1075.0,5.0,...| 52|
|[43.0,1927.0,3.0,...| 43|
|[50.0,3799.0,5.0,...| 50|
+--------------------+---+
only showing top 20 rows
```

Fig-32



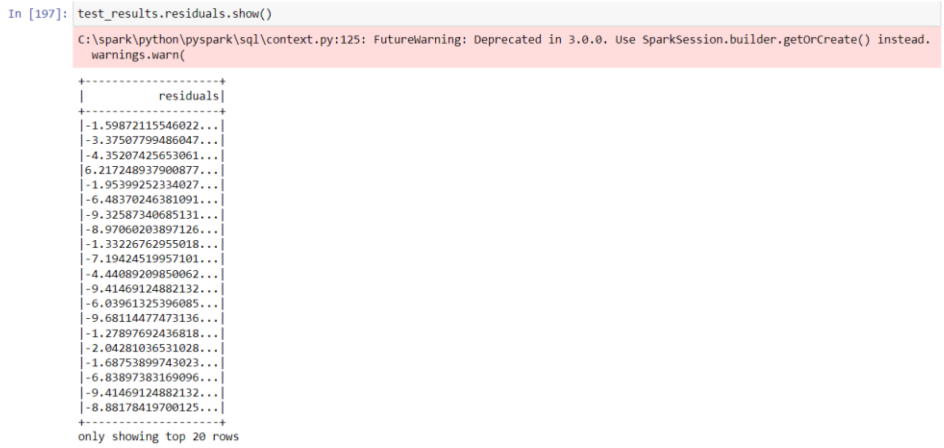
```
In [192]: train_data,test_data = final_data.randomSplit([0.7,0.3])

In [193]: test_data.describe().show()

+-------+------------------+
|summary|               Age|
+-------+------------------+
|  count|              7859|
|   mean| 39.47016159816771|
| stddev|15.139109565764796|
|    min|                 7|
|    max|                85|
+-------+------------------+
```

Fig-33



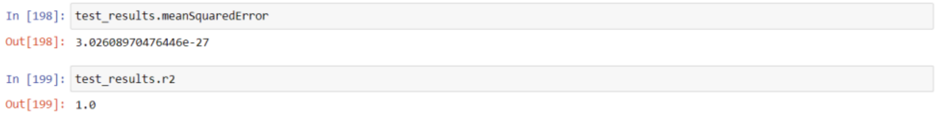
```
In [194]: from pyspark.ml.regression import LinearRegression
          lr = LinearRegression(labelCol='Age')

In [195]: lr_model = lr.fit(train_data)

In [196]: test_results=lr_model.evaluate(test_data)
```

Fig-34



```
In [197]: test_results.residuals.show()

C:\spark\python\pyspark\sql\context.py:125: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
  warnings.warn(

+--------------------+
|           residuals|
+--------------------+
|-1.59872115546022...|
|-3.37507799486047...|
|-4.35207425653061...|
|6.217248937900877...|
|-1.95399252334027...|
|-6.48370246381091...|
|-9.32587340685131...|
|-8.97060203897126...|
|-1.33226762955018...|
|-7.19424519957101...|
|-4.44089209850062...|
|-9.41469124882132...|
|-6.03961325396085...|
|-9.68114477473136...|
|-1.27897692436818...|
|-2.04281036531028...|
|-1.68753899743023...|
|-6.83897383169096...|
|-9.41469124882132...|
|-8.88178419700125...|
+--------------------+
only showing top 20 rows
```

Fig-35

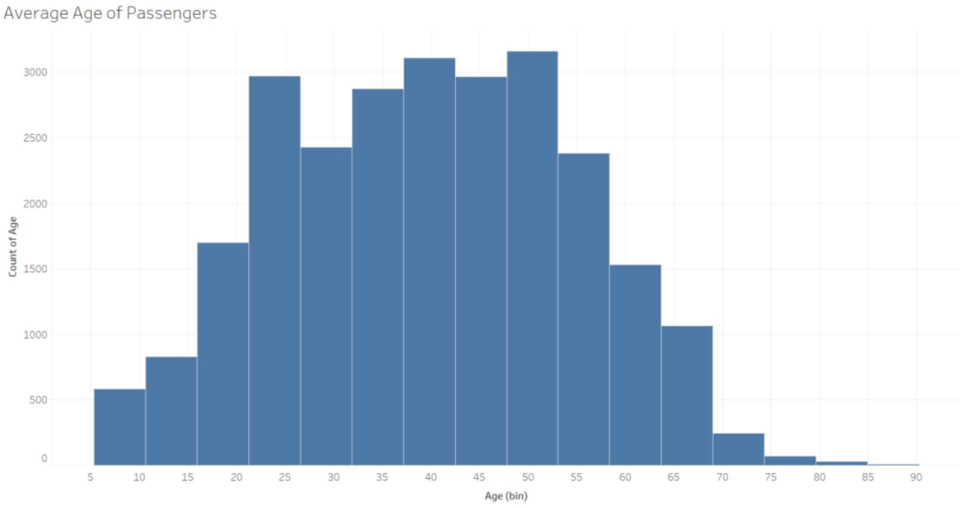


```
In [198]: test_results.meanSquaredError
Out[198]: 3.02608970476446e-27

In [199]: test_results.r2
Out[199]: 1.0
```

Fig-36

2.5 Data Visualizations

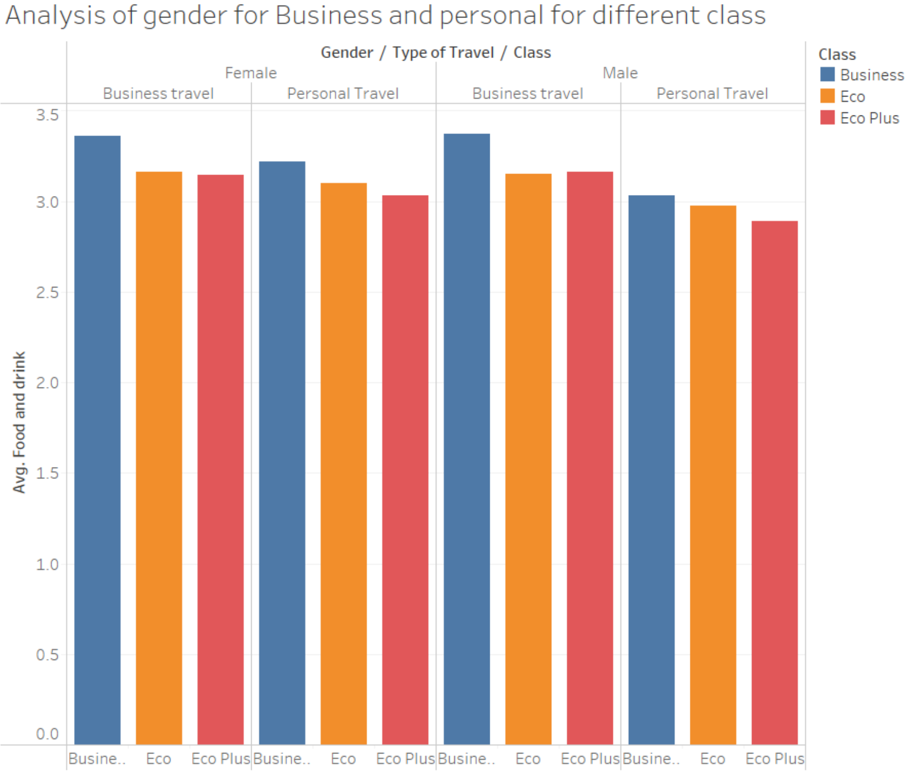
For reporting and analysing enormous amounts of data, Tableau is a superb business intelligence and data visualisation application. In June 2019, Salesforce bought Tableau, an American business that was founded in 2003. It assists users in producing a variety of graphs, maps, dashboards, and stories for the purpose of visualising and analysing data to aid in business decision-making.



The trend of count of Age for Age (bin). The view is filtered on Age (bin), which ranges from 5 to 85.

Fig-37

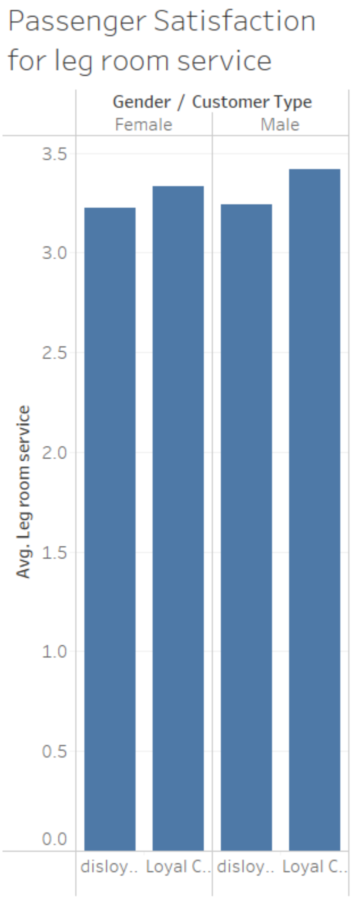
Fig-37 shows the average age of the passengers in the dataset .In the histogram on the x axis Age(bin) is taken and on the y-axis count of age is taken .We analyse from the histogram that the average age of the passenger were 40 – 50 years .


Analysis of gender for Business and personal for different class

Average of Food and drink for each Class broken down by Gender and Type of Travel. Color shows details about Class.

Fig-38

Fig-38 shows the bar graph of type of travel, gender , and class on the x-axis and avg food and drink rating. The ratings is marked from 0 to 5 .Both the female and male are divided into 2 groups business and personal, and bar chart shows different types of classes which are business, economy and economy plus. For female in business travel had the most average rating of 3.4 ,while economy and economy plus had same rating and in personal travel business had 3.3 rating followed by economy of 3.1 and economy with 3.0.Similar situation were found in male also ,for business travel 3.6 rating was given ,while economy and economy plus had the same rating of 3.2.For personal travel the ratings performed poorly with business class rating of 3.0 ,economy and economy plus with 2.9 and 2.7.

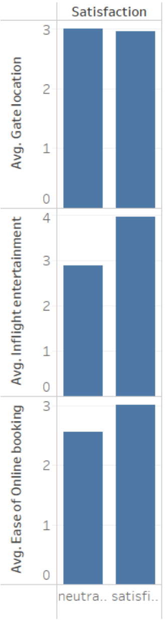


Average of Leg room service for each Customer Type broken down by Gender.

Fig-39

Fig 39 shows the bar chart between loyal and disloyal customers based on the gender. For female loyal customers gave higher rating of 3.3 average ,while disloyal customer gave rating of 3.1 .For male loyal customers gave higher rating of 3.4 while disloyal customer got 3.1 rating.

Satisfaction of passengers for Gate location ,Inflight entertainment , Ease of online booking



Average of Gate location, average of Inflight entertainment and average of Ease of Online booking for each Satisfaction.

Fig-40

Fig-40 shows 3 bar graphs for passengers satisfaction for the factors for ease of online booking , inflight entertainment , gate location .During the analysis it was found that passengers were satisfied and neutral for the location of the gate both of them got same rating of 3 .

For the inflight entertainment passengers were highly satisfied with services and gave a rating of 4 ,while dissatisfied customers gave 2.9.

For Ease of online booking satisfied passengers gave a rating of 3,while neutral or dissatisfied people gave a rating of 2.5.

3.Discussion

The discussion that follows focuses on this study's visualization and machine learning findings. I have used the dataset from Kaggle which is airline passenger satisfaction .The dataset gives information about the age ,customer type ,satisfaction ,inflight services ,seat comfort ,leg space ,departure and arrival delays .I have used Kmeans clustering machine learning algorithm to analyse the data with the help of pyspark and for data visualiztions I have used tableau.

To run pyspark in my computer I have used jupyter notebook ,where the necessary libraries were imported ,one of them was findspark . I have used only 17 columns for analysis of the clustering data.I have used every factor that depends on passenger satisfaction.During my analysis it was found that set sum of squares was really high ,which means the points were not enough close to the centroid ,so to decrease the value of WSSSE ,the k value has to be increased .As shown in the final prediction each rows were clustered in to respective clusters from 0-5 .

For analysis of the linear regression the mean square error came as 3.026 You may determine how closely a regression line resembles a set of points using the mean squared error (MSE).Since the value is small the model is a perfect fit for the analysis

4.Conclusion

Data was pre-processed and visualised as part of this project, followed by analysis and the application of the relevant machine learning models. The ideal solution for this project is PySpark. For big data management pyspark and tableau are really good .There is detailed documentation for every programme used in this report. The results are good overall, although there are some questions about the accuracy and quality of the data.

5.References

1. *Airline passenger satisfaction*. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

2. Biswal, A. (2021, March 9). *What is tableau: The ultimate guide to know all about tableau in 2021*. Simplilearn.com. https://www.simplilearn.com/tutorials/tableau-tutorial/what-is-tableau

3. Garbade, M. J. (2018, September 12). *Understanding K-means clustering in machine learning*. Medium. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

4. Pointer, I. (n.d.). *What is Apache spark? The big data platform that crushed Hadoop*. InfoWorld. https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html

5.YouTube. https://www.youtube.com/watchv=mbYNKM3hdPI&t=74s&ab_channel=DataScienceforEveryone

6. *Getting started — PySpark 3.3.0 documentation* [Video]. (n.d.). Apache Spark™ - Unified Engine for large-scale data analytics. https://spark.apache.org/docs/latest/api/python/getting_started/index.html

7. Thevapalan, A. (2021, August 1). *Creating visualizations with tableau made simple* [Video]. Medium. https://towardsdatascience.com/creating-visualizations-with-tableau-made-simple-ed86401e63b0

8. *Mean squared error: Definition and example*. (2021, May 13). Statistics How To. https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/

6.Appendix

Appendix A – Clustering Code

Importing the libraries

```
import findspark
findspark.init()
findspark.find()
import pyspark
findspark.find()


from pyspark.sql import SparkSession

spark =
SparkSession.builder.appName('GlenCoursework').getOrCreate()

dataset =
spark.read.csv('test.csv',header=True,inferSchema=True)

dataset.printSchema()

dataset.toPandas()

from pyspark.ml.clustering import KMeans

from pyspark.ml.feature import VectorAssembler

Dataset.columns

my_cols=['Flight Distance',
 'Inflight wifi service',
 'Departure/Arrival time convenient',
 'Ease of Online booking',
 'Gate location',
 'Food and drink',
 'Online boarding',
 'Seat comfort',
 'Inflight entertainment',
 'On-board service',
 'Leg room service',
 'Baggage handling',
 'Checkin service',
 'Inflight service',
 'Cleanliness',
 'Departure Delay in Minutes',
      'Arrival Delay in Minutes',
     ]
     df=dataset.select(my_cols)
     df.toPandas()

     df1=df.na.drop()
     df1.show()

     assembler =
     VectorAssembler(inputCols=df1.columns,outputCol='features')

     final_data = assembler.transform(df1)

     final_data.printSchema()

     from pyspark.ml.feature import StandardScaler

     scaler =
     StandardScaler(inputCol='features',outputCol='scaledFeatures
     ')

     scaler_model = scaler.fit(final_data)

     final_data =scaler_model.transform(final_data)

     kmeans = KMeans(featuresCol='scaledFeatures',k=6)

     model = kmeans.fit(final_data)

     final_data.head(1)

     kmeans = KMeans(featuresCol='scaledFeatures',k=6)

     print('WSSSE')
        print(model.summary.trainingCost) model =
kmeans.fit(final_data)

     centers = model.clusterCenters()

     print(centers)

     model.transform(final_data).select('prediction').show()
```

AppendixB –Data visualization

The trend of count of Age for Age (bin). The view is filtered on Age (bin), which ranges from 5 to 85.

Class.

Customer Type broken down by Gender.

booking for each
Satisfaction.

Flight Distance vs Seat comfort

Seat comfort.

As the flight distance increased the seat got started to get uncomfortable.

of Travel and Satisfaction.

From the above figure it is shown that business travel people were more satisfied than the passengers who travelled personally.

Appendex C –Linear Regression

```
import findspark
findspark.init()
findspark.find()
import pyspark
findspark.find()
```

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('GlenCoursework').getOrCreate()
```

```
dataset.printSchema()
```

```
from pyspark.ml.clustering import KMeans
```

```
from pyspark.ml.feature import VectorAssembler
```

```
assembler = VectorAssembler(inputCols=['Age','Flight Distance',
 'Inflight wifi service',
 'Departure/Arrival time convenient',
 'Ease of Online booking',
 'Gate location',
 'Food and drink',
 'Online boarding',
 'Seat comfort',
 'Inflight entertainment',
 'On-board service',
 'Leg room service',
 'Baggage handling',
 'Checkin service',
 'Inflight service',
 'Cleanliness'],outputCol="features")
```

```
output = assembler.transform(dataset)

final_data = output.select('features','Age')

final_data.show()

train_data,test_data = final_data.randomSplit([0.7,0.3])

test_data.describe().show()

from pyspark.ml.regression import LinearRegression
lr = LinearRegression(labelCol='Age')

lr_model = lr.fit(train_data)

test_results.residuals.show()

test_results.meanSquaredError

test_results.r2
```