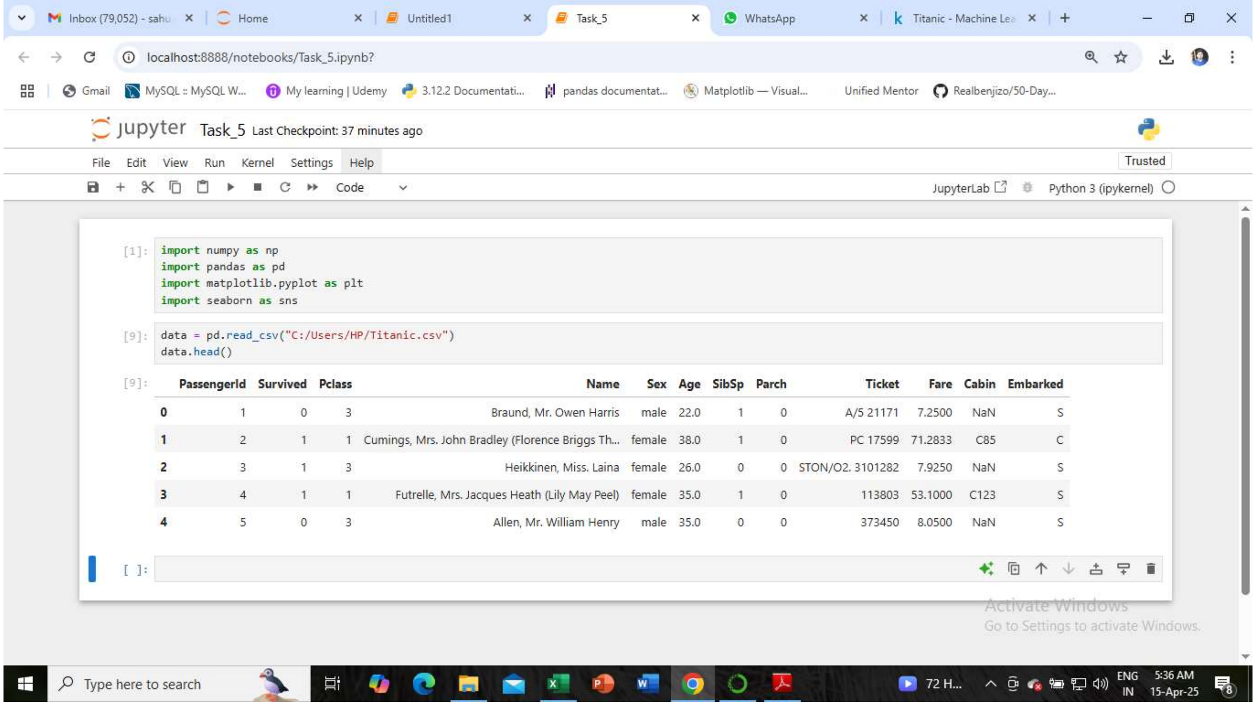
After importing different libraries and reading the dataset, we tried to see first few rows of the data using method head().



```
[1]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

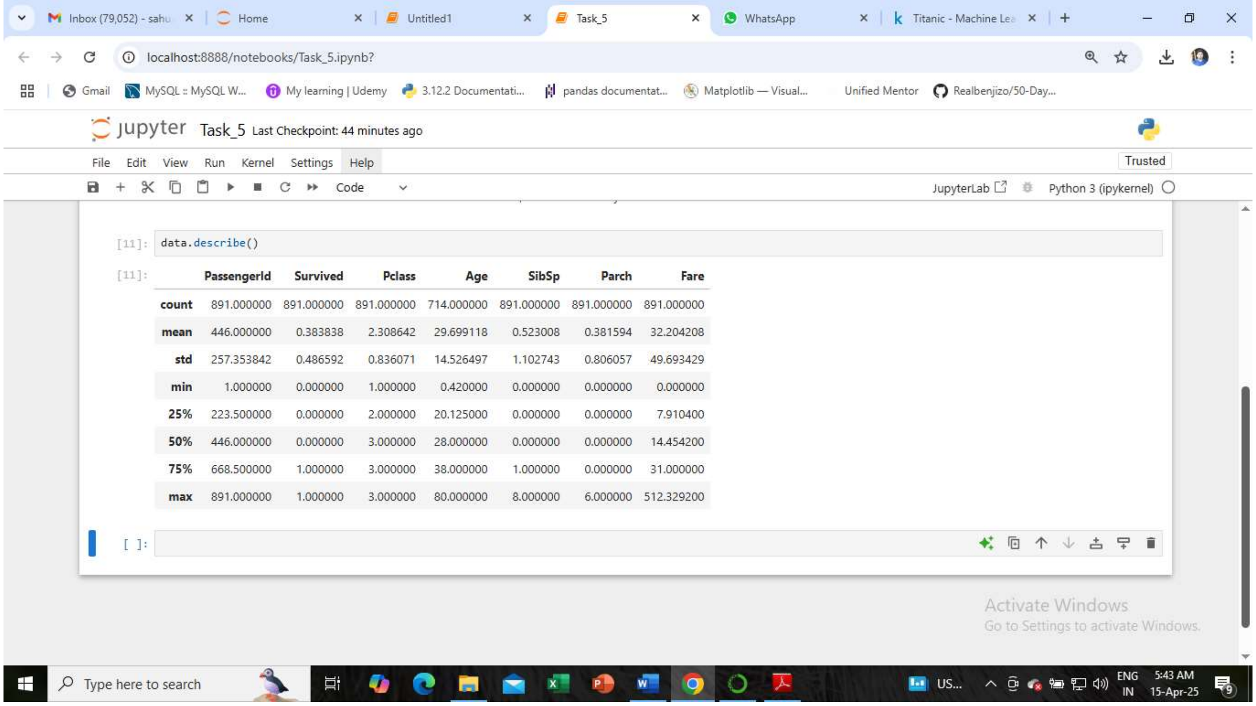
```
[9]: data = pd.read_csv("C:/Users/HP/Titanic.csv")
     data.head()
```

[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
[ ]:
```

Using describe(), we get to see different statistical factors of columns.



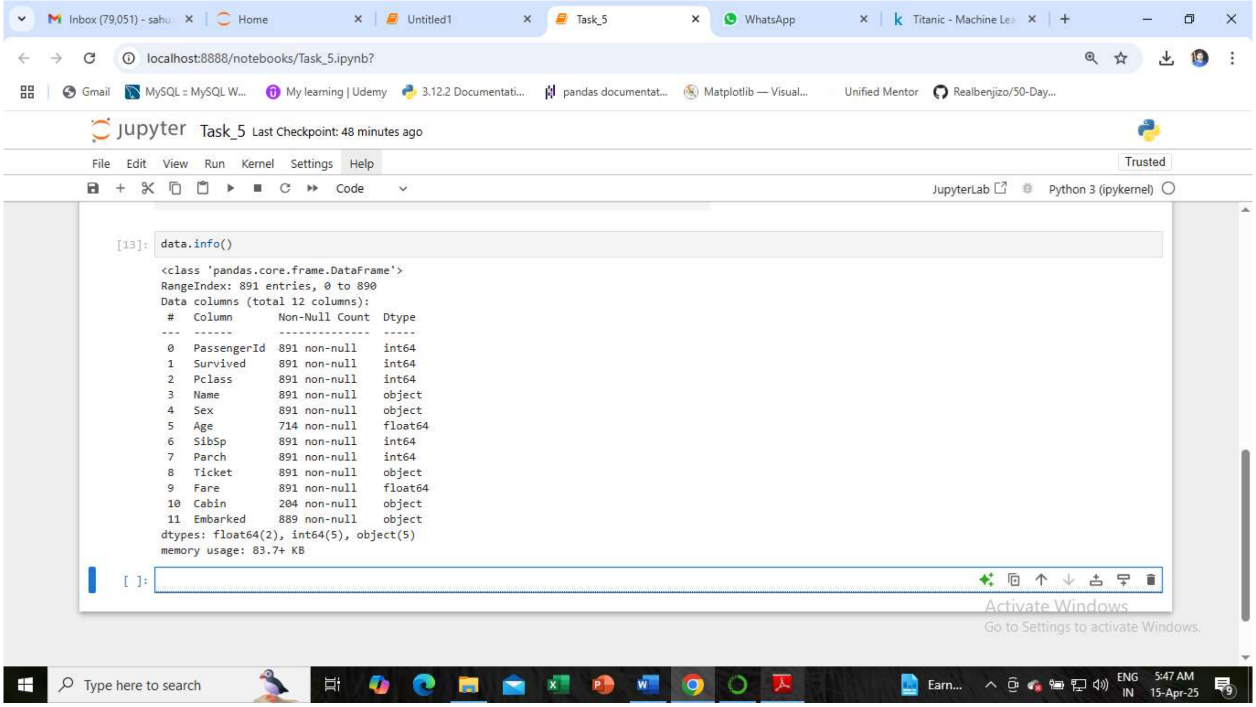
```
[11]: data.describe()
```

[11]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
[ ]:
```

Using info(), we get to see count of non-null values in each column; also, we can see the datatype for each column.



```
[13]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

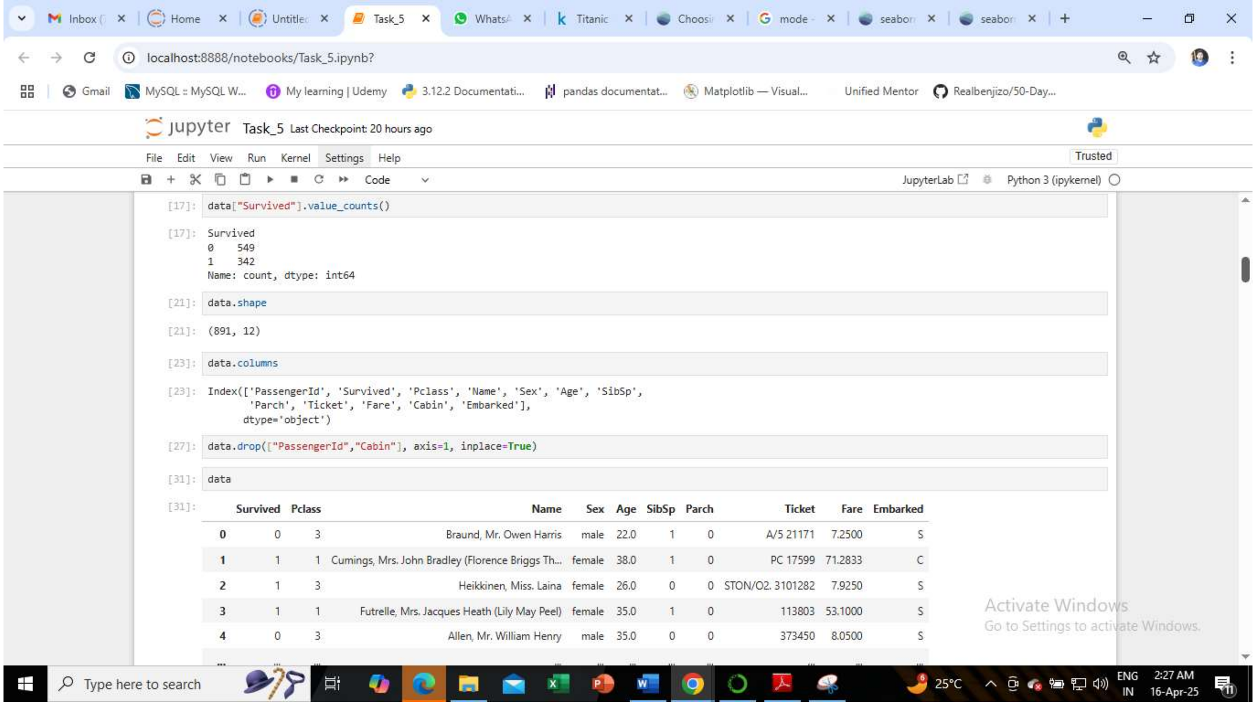
```
[ ]:
```

Using value_counts() in a column here gave us the counts of each category type in the column (like 0 and 1 are the categories here).

Using shape, we can check the number of rows and column of a dataframe.

Using column, we can see the column names.

Then we tried to drop 2 columns (Passenger Id and Cabin) as they were not giving any intelligence to our further analysis.

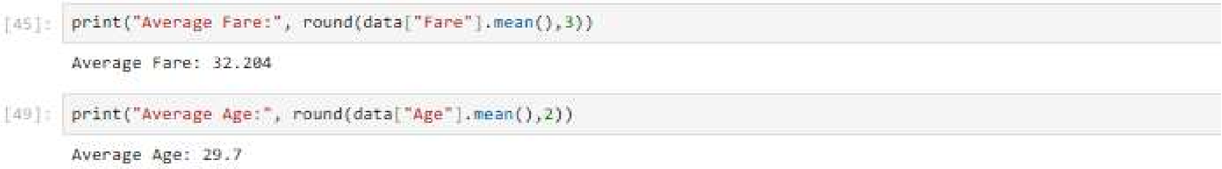


```
[17]: data["Survived"].value_counts()

[17]: Survived
      0    549
      1    342
      Name: count, dtype: int64

[21]: data.shape

[21]: (891, 12)

[23]: data.columns

[23]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
             'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
            dtype='object')

[27]: data.drop(["PassengerId","Cabin"], axis=1, inplace=True)

[31]: data
```

[31]:

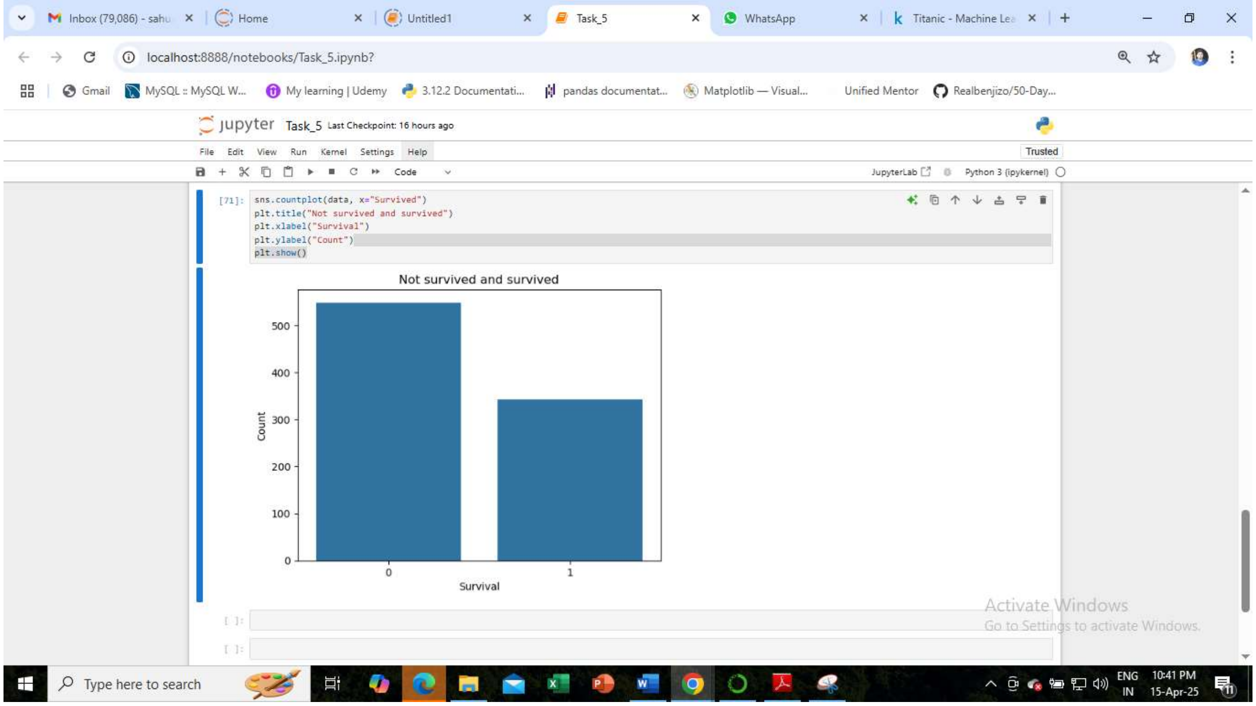
	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

Below screenshot showing mean() for Fare and Age using methods. The values were rounded off using round method.



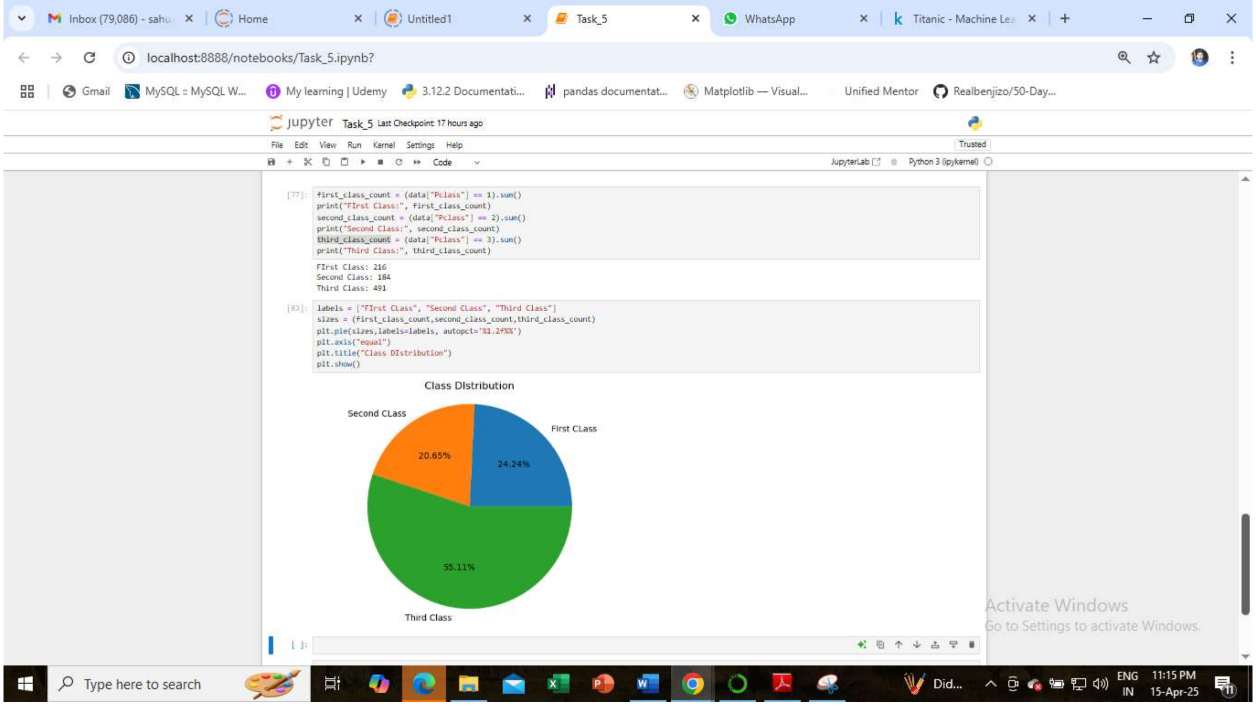
```
[45]: print("Average Fare:", round(data["Fare"].mean(),3))

      Average Fare: 32.204

[49]: print("Average Age:", round(data["Age"].mean(),2))

      Average Age: 29.7
```

This is a simple countplot which shows the plot for count of survived passengers and the count of the passengers who were unable to survive.



```
[71]: sns.countplot(data, x="Survived")
      plt.title("Not survived and survived")
      plt.xlabel("Survival")
      plt.ylabel("Count")
      plt.show()
```

This is a pie-plot where it shows the distribution of passengers as per the class (1, 2 &3)



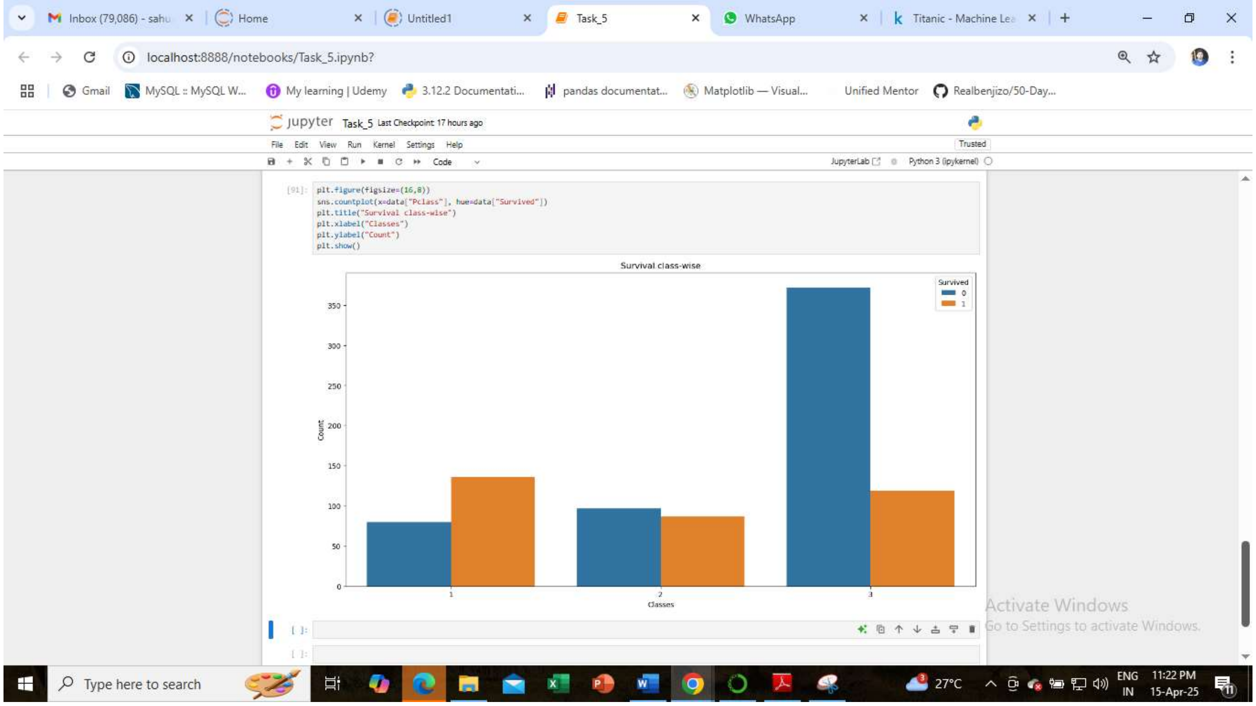
```
[77]: first_class_count = (data["Pclass"] == 1).sum()
      print("FIrst Class:", first_class_count)
      second_class_count = (data["Pclass"] == 2).sum()
      print("Second Class:", second_class_count)
      third_class_count = (data["Pclass"] == 3).sum()
      print("Third Class:", third_class_count)
```

```
FIrst Class: 216
Second Class: 184
Third Class: 491
```

```
[83]: labels = ["FIrst CLass", "Second CLass", "Third Class"]
      sizes = (first_class_count,second_class_count,third_class_count)
      plt.pie(sizes,labels=labels, autopct='%1.2f%%')
      plt.axis("equal")
      plt.title("Class DIstribution")
      plt.show()
```

Class DIstribution
Second CLass
First CLass
Third Class
20.65%
24.24%
55.11%
localhost:8888/notebooks/Task_5.ipynb?
jupyter Task_5 Last Checkpoint: 17 hours ago
Activate Windows
Go to Settings to activate Windows.
11:15 PM
15-Apr-25

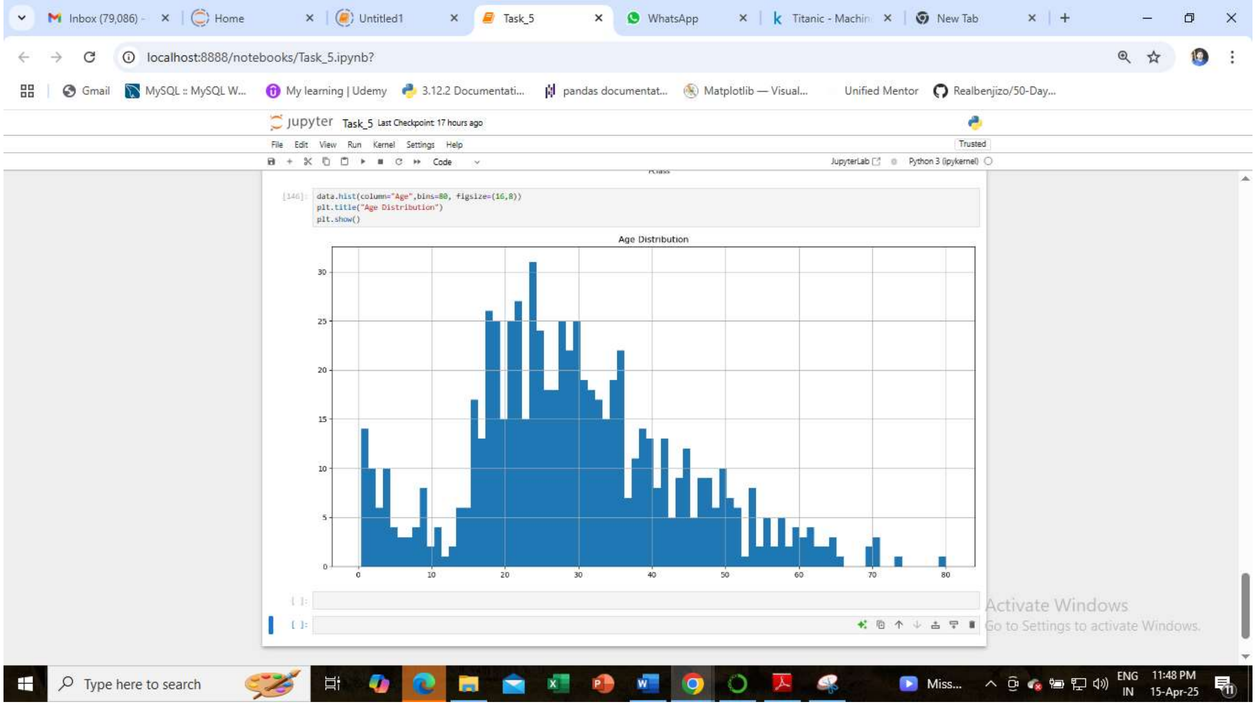
Again a countplot which also shows the survival with classes as well.



```
[91]: plt.figure(figsize=(16,8))
      sns.countplot(x=data["Pclass"], hue=data["Survived"])
      plt.title("Survival class-wise")
      plt.xlabel("Classes")
      plt.ylabel("Count")
      plt.show()
```

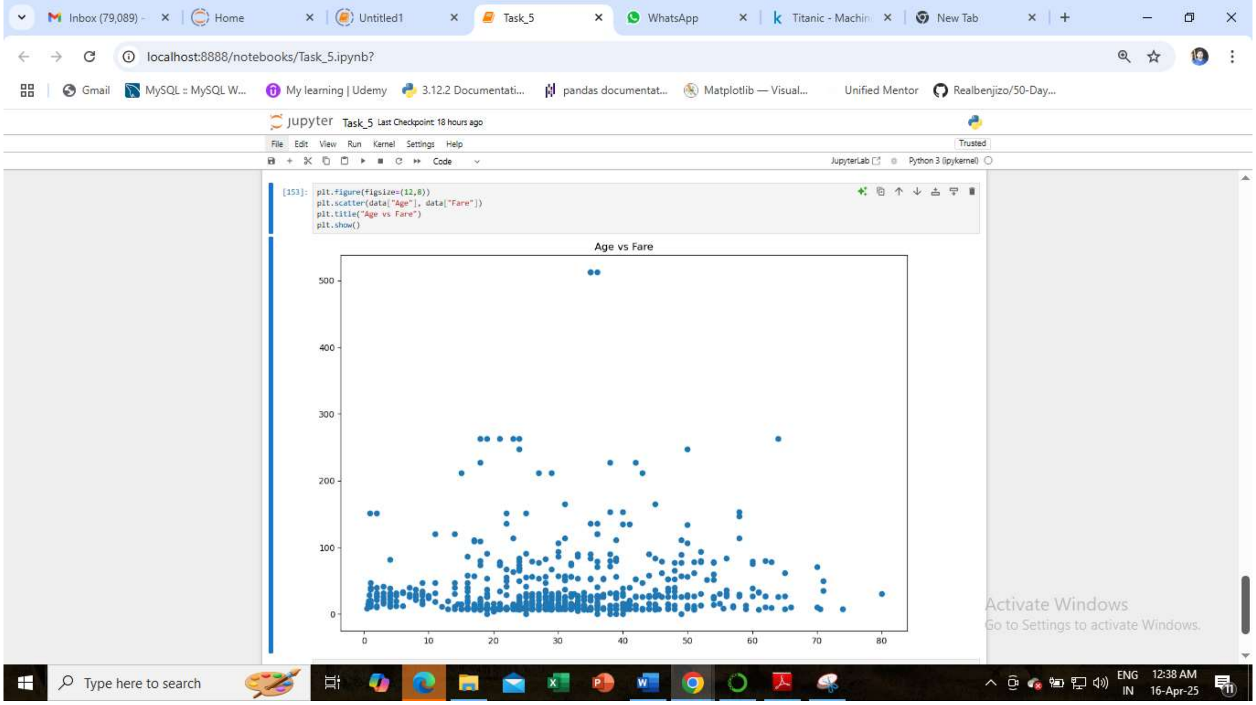
Survival class-wise
Survived
0
1
Count
0
50
100
150
200
250
300
350
1
2
3
Classes
localhost:8888/notebooks/Task_5.ipynb?
jupyter Task_5 Last Checkpoint: 17 hours ago
Activate Windows
Go to Settings to activate Windows.
11:22 PM
15-Apr-25

A histplot which shows the distribution of age. It shows that most of the people aboard were of the age range of 20 and 40.



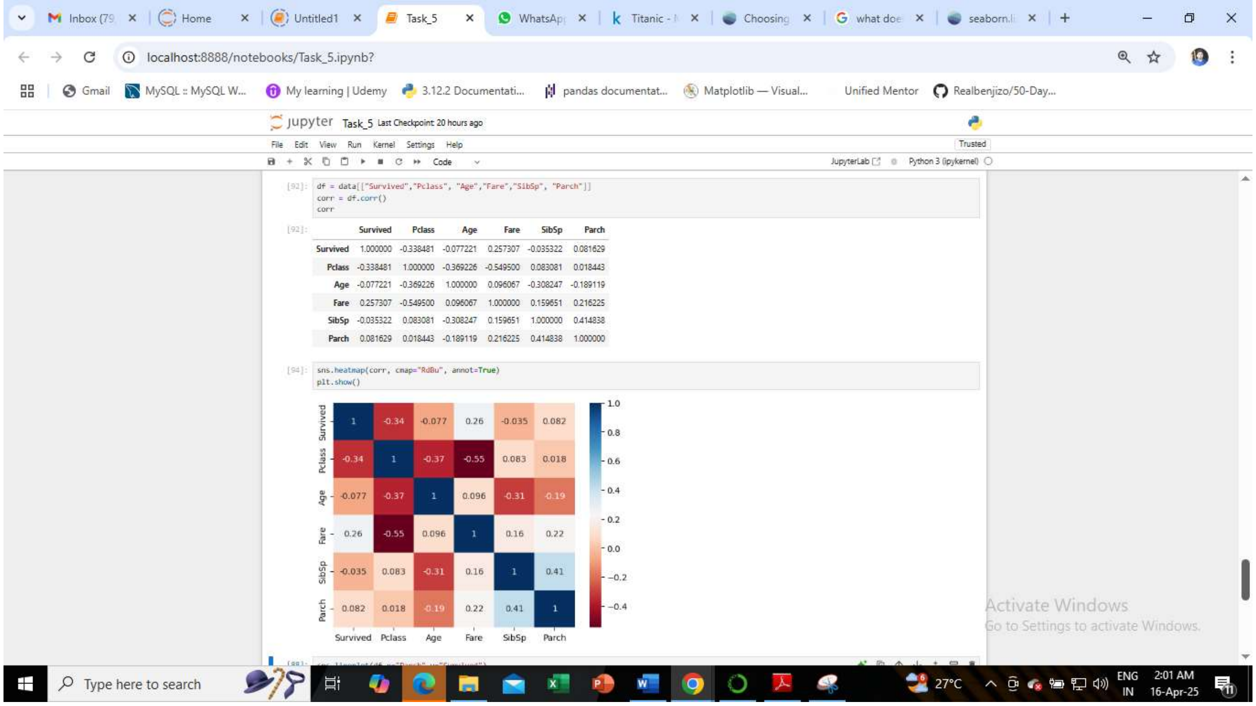
```
[146]: data.hist(column="Age",bins=80, figsize=(16,8))
       plt.title("Age Distribution")
       plt.show()
```

The scatterplot which shows that most of the people of all ages bought the tickets of price below 100 Dollars.



```
[153]: plt.figure(figsize=(12,8))
       plt.scatter(data["Age"], data["Fare"])
       plt.title("Age vs Fare")
       plt.show()
```

The heatmap below shows that some of the column values have positive correlation with few other column values while others have a negative correlation. Like, for example, Pclass have a negative correlation with the Survived.

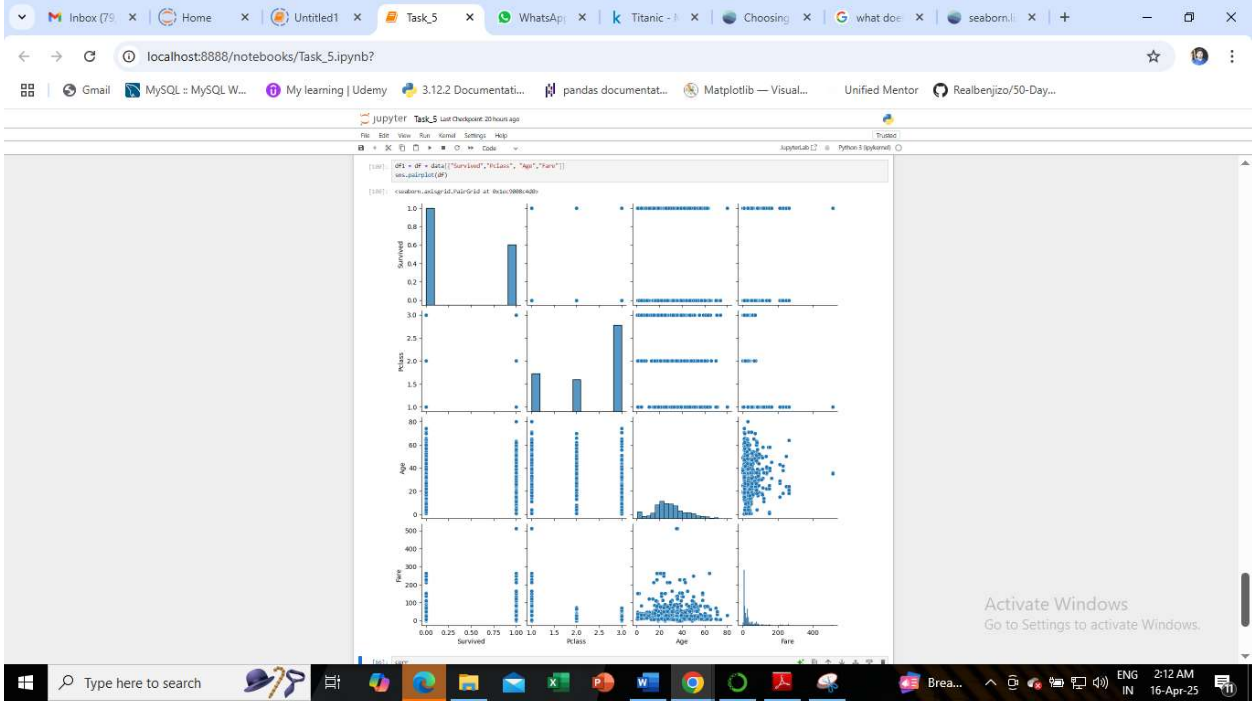


```
[92]: df = data[["Survived","Pclass", "Age","Fare","SibSp", "Parch"]]
      corr = df.corr()
      corr
```

```
[92]:
          Survived    Pclass       Age      Fare     SibSp     Parch
Survived  1.000000 -0.338481 -0.077221  0.257307 -0.035322  0.081629
  Pclass -0.338481  1.000000 -0.369226 -0.549500  0.083081  0.018443
     Age -0.077221 -0.369226  1.000000  0.096067 -0.308247 -0.189119
    Fare  0.257307 -0.549500  0.096067  1.000000  0.159651  0.216225
   SibSp -0.035322  0.083081 -0.308247  0.159651  1.000000  0.414838
   Parch  0.081629  0.018443 -0.189119  0.216225  0.414838  1.000000
```

```
[94]: sns.heatmap(corr, cmap="RdBu", annot=True)
      plt.show()
```

The pairplot() gave us plots of one column values against all other column values. Like for example, we can see here that with the increase of fare the number diminished for people irrespective of the fact whether they survived or not.

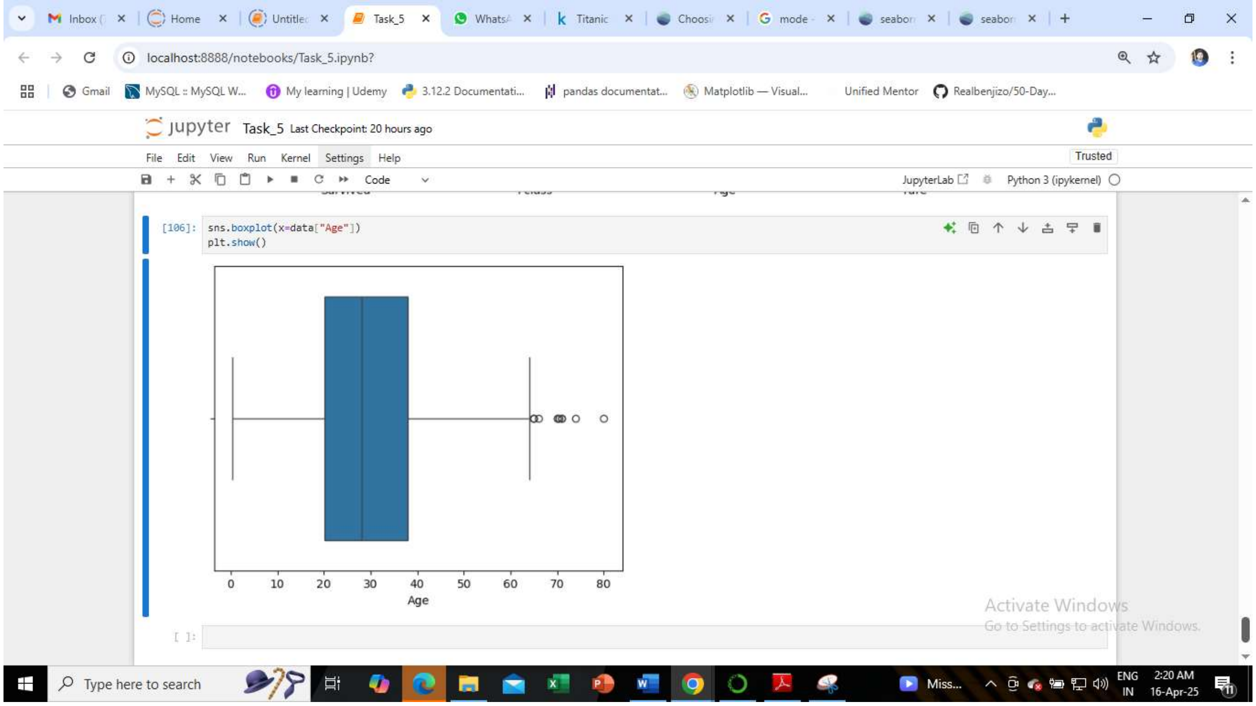


```
[100]: df1 = df = data[["Survived","Pclass", "Age","Fare"]]
       sns.pairplot(df)
```

```
[100]: <seaborn.axisgrid.PairGrid at 0x1ac9088c4d0>
```

Task_5 Last Checkpoint: 20 hours ago
localhost:8888/notebooks/Task_5.ipynb?
Survived
Pclass
Age
Fare
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG IN
2:12 AM
16-Apr-25

The boxplot again showing us that most of the people were in the age range of 20 and 40. The median lies in between that age range and there are some outliers in the range of 65-80.



localhost:8888/notebooks/Task_5.ipynb?
jupyter Task_5 Last Checkpoint: 20 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
0
10
20
30
40
50
60
70
80
Age
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:20 AM
16-Apr-25

```
[106]: sns.boxplot(x=data["Age"])
       plt.show()
```

Summary of findings-

- Most of the people are in the age range of 24-40.
- There were quite enough data columns where numerical analysis could be done.
- Passenger Id and Cabin columns were removed as they were not helping any inferences and plots.
- There were more people who could not survive.
- Most of the people made through journey in class 3.
- Comparatively, higher number of passengers died from class 3.
- Most of the passengers paid for the fare below 100 Dollars.