SmartBridge Applied Data Science

Name: Routhu Sai Praneeth

Mail: routhusai.praneeth2020@vitstudent.ac.in

ADS Assignment 2

Titanic Ship Case Study:

Problem Description:

On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. Translated 32% survival rate.

- One of the reasons that the shipwreck led to such loss of life was that there were notenough lifeboats for the passengers and crew.
- Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

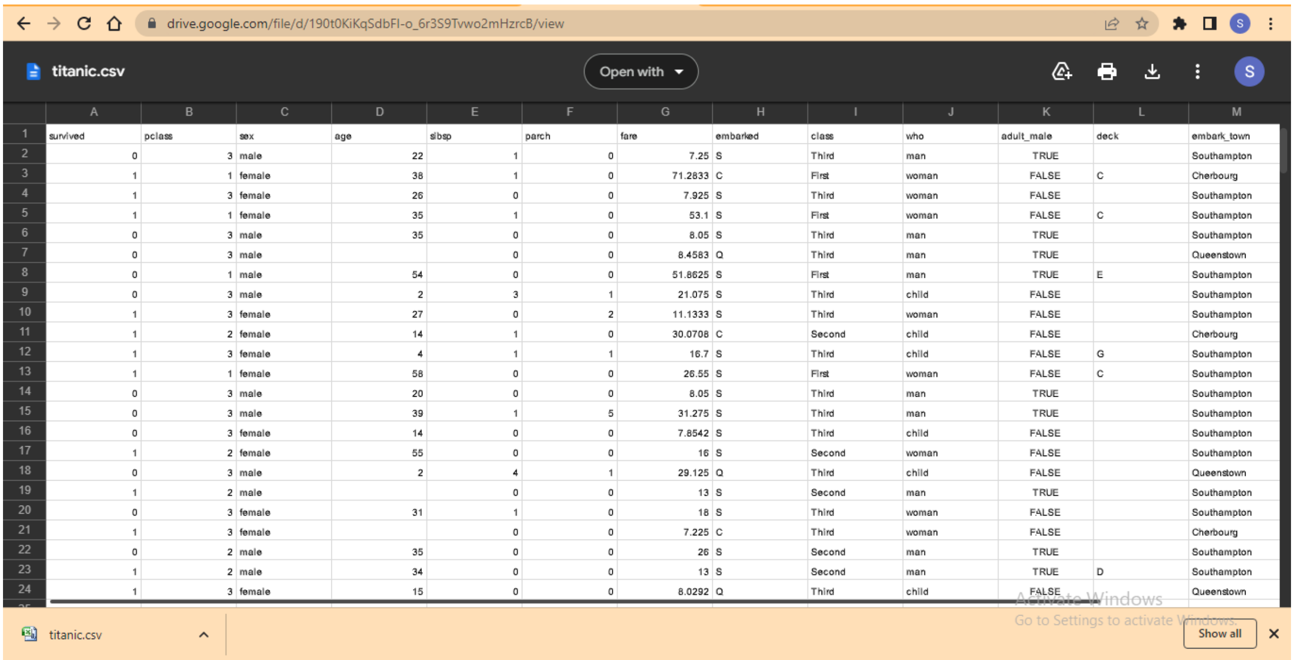
The problem associated with the Titanic dataset is to predict whether a passenger survived the disaster or not. The dataset contains various features such as passenger class, age, gender, cabin, fare, and whether the passenger had any siblings or spouses on board. These features canbe used to build a predictive model to determine the likelihood of a passenger surviving the disaster. The dataset offers opportunities for feature engineering, data visualization, and model selection, making it a valuable resource for developing and testing data analysis and machine learning skills.

Drive Link to Colab File:

Link

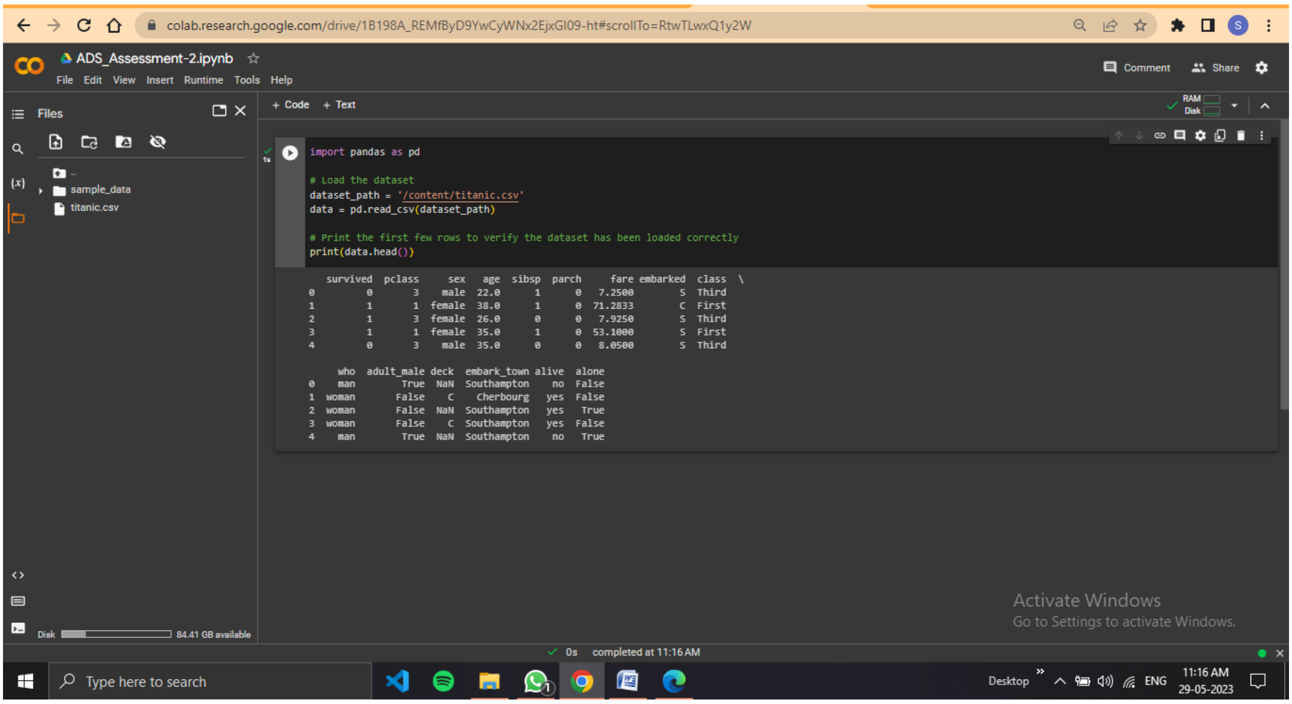
Tasks:

1.Downloading the dataset:



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town
2	0	3	male	22	1	0	7.25	S	Third	man	TRUE		Southampton
3	1	1	female	38	1	0	71.2833	C	First	woman	FALSE	C	Cherbourg
4	1	3	female	26	0	0	7.925	S	Third	woman	FALSE		Southampton
5	1	1	female	35	1	0	53.1	S	First	woman	FALSE	C	Southampton
6	0	3	male	35	0	0	8.05	S	Third	man	TRUE		Southampton
7	0	3	male		0	0	8.4583	Q	Third	man	TRUE		Queenstown
8	0	1	male	54	0	0	51.8625	S	First	man	TRUE	E	Southampton
9	0	3	male	2	3	1	21.075	S	Third	child	FALSE		Southampton
10	1	3	female	27	0	2	11.1333	S	Third	woman	FALSE		Southampton
11	1	2	female	14	1	0	30.0708	C	Second	child	FALSE		Cherbourg
12	1	3	female	4	1	1	16.7	S	Third	child	FALSE	G	Southampton
13	1	1	female	58	0	0	26.55	S	First	woman	FALSE	C	Southampton
14	0	3	male	20	0	0	8.05	S	Third	man	TRUE		Southampton
15	0	3	male	39	1	5	31.275	S	Third	man	TRUE		Southampton
16	0	3	female	14	0	0	7.8542	S	Third	child	FALSE		Southampton
17	1	2	female	55	0	0	16	S	Second	woman	FALSE		Southampton
18	0	3	male	2	4	1	29.125	Q	Third	child	FALSE		Queenstown
19	1	2	male		0	0	13	S	Second	man	TRUE		Southampton
20	0	3	female	31	1	0	18	S	Third	woman	FALSE		Southampton
21	1	3	female		0	0	7.225	C	Third	woman	FALSE		Cherbourg
22	0	2	male	35	0	0	26	S	Second	man	TRUE		Southampton
23	1	2	male	34	0	0	13	S	Second	man	TRUE	D	Southampton
24	1	3	female	15	0	0	8.0292	Q	Third	child	FALSE		Queenstown

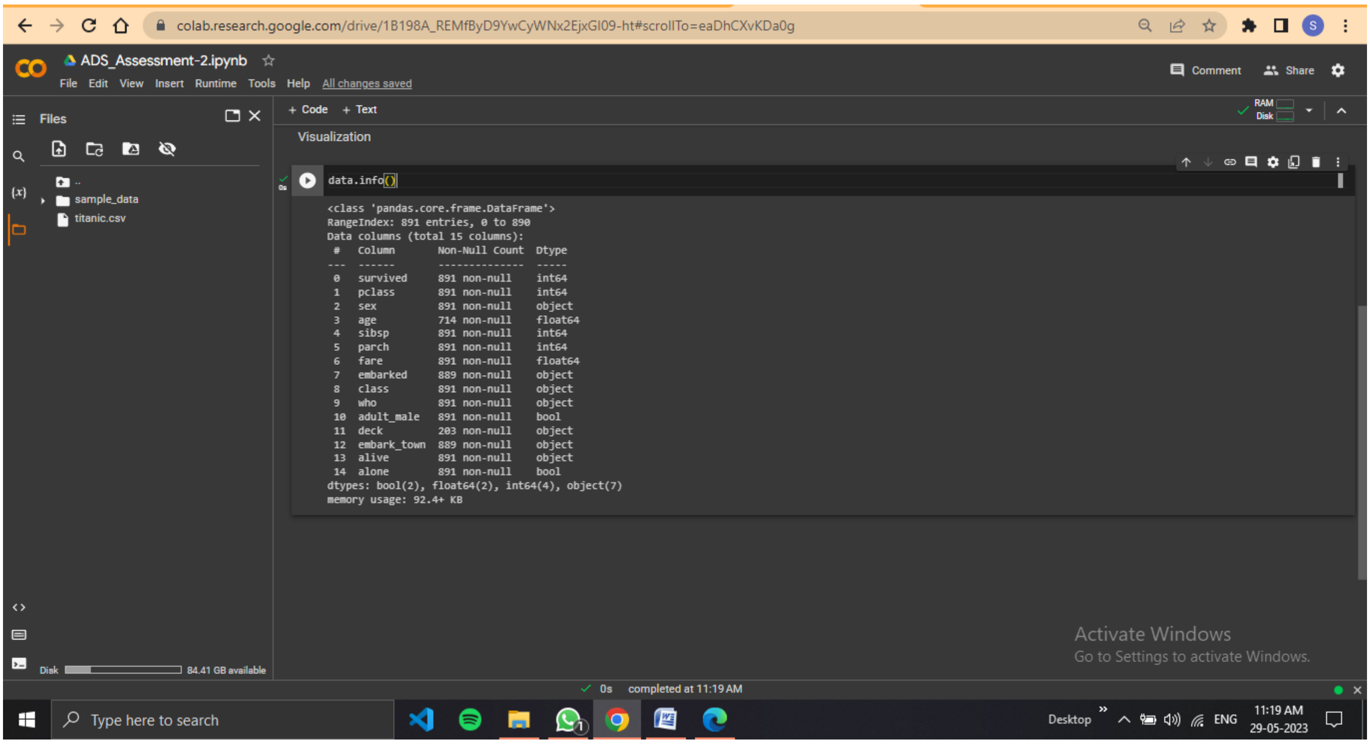
2.Load the dataset.



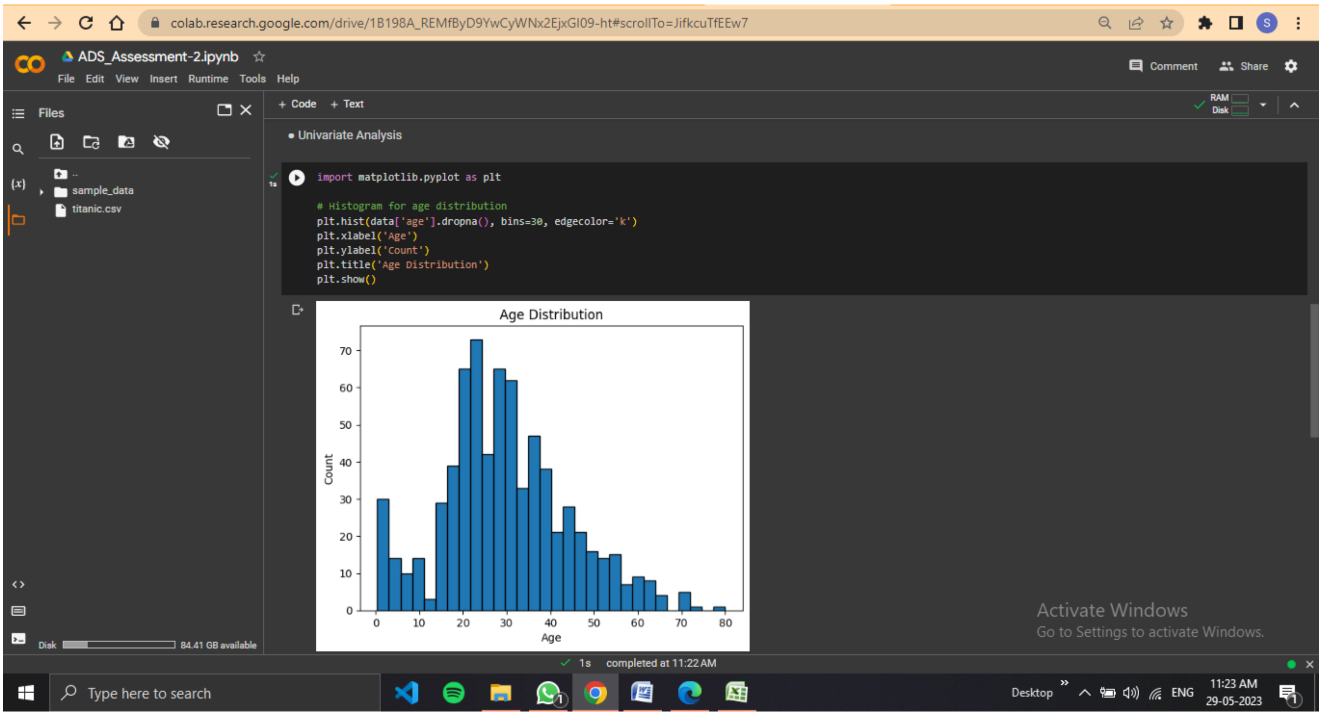

```python
import pandas as pd

# Load the dataset
dataset_path = '/content/titanic.csv'
data = pd.read_csv(dataset_path)

# Print the first few rows to verify the dataset has been loaded correctly
print(data.head())
```

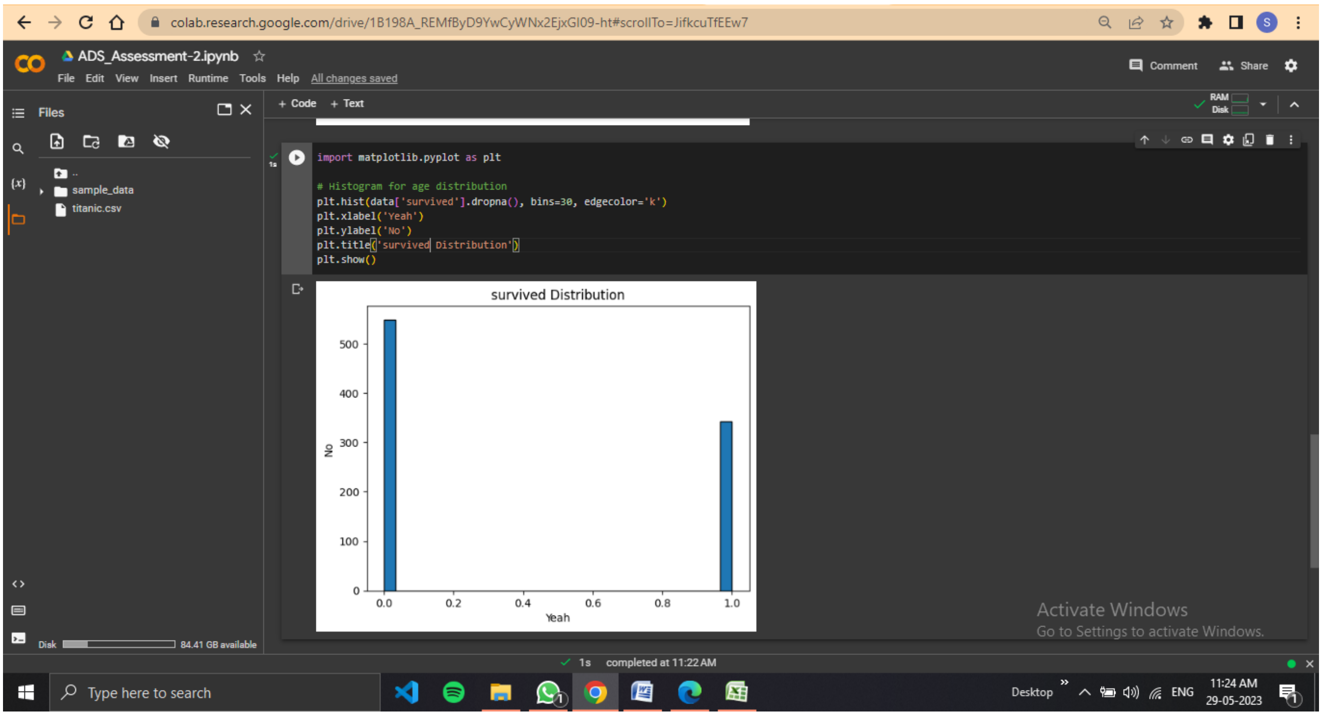
```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

3.Perform Below Visualizations.

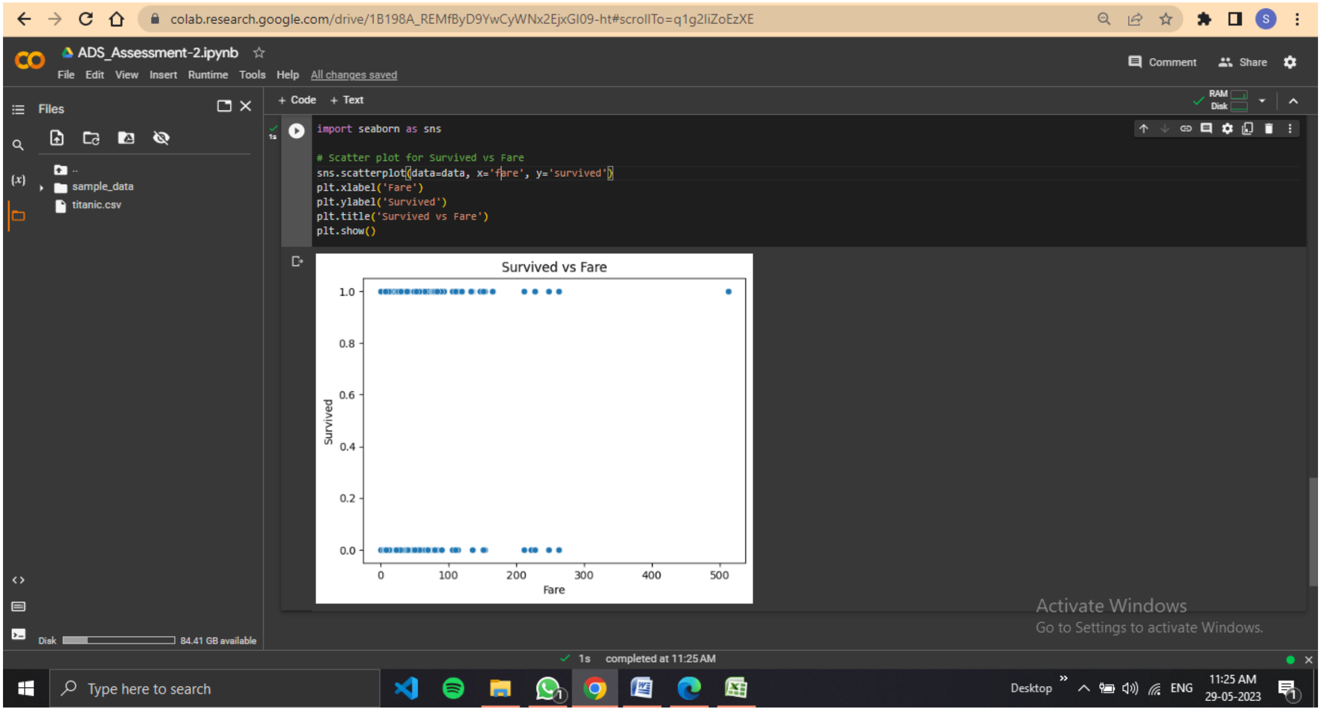


• Univariate Analysis

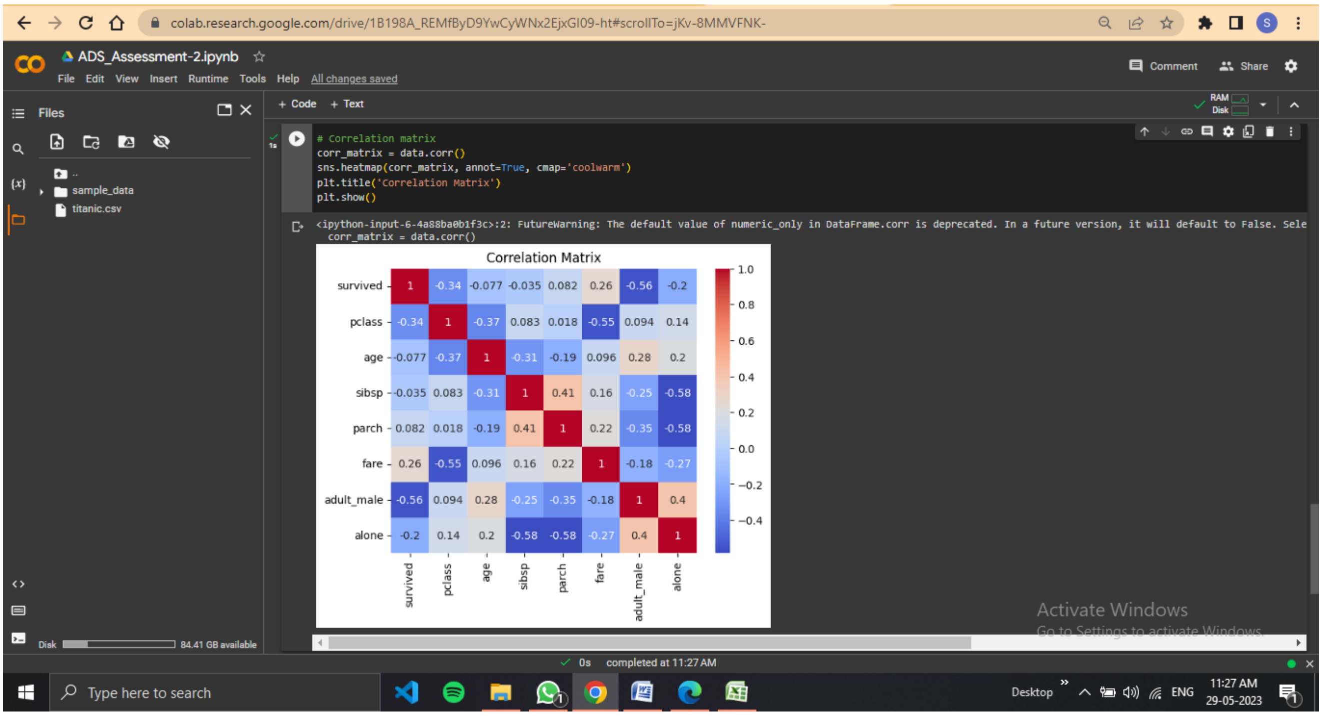




• Bi - Variate Analysis



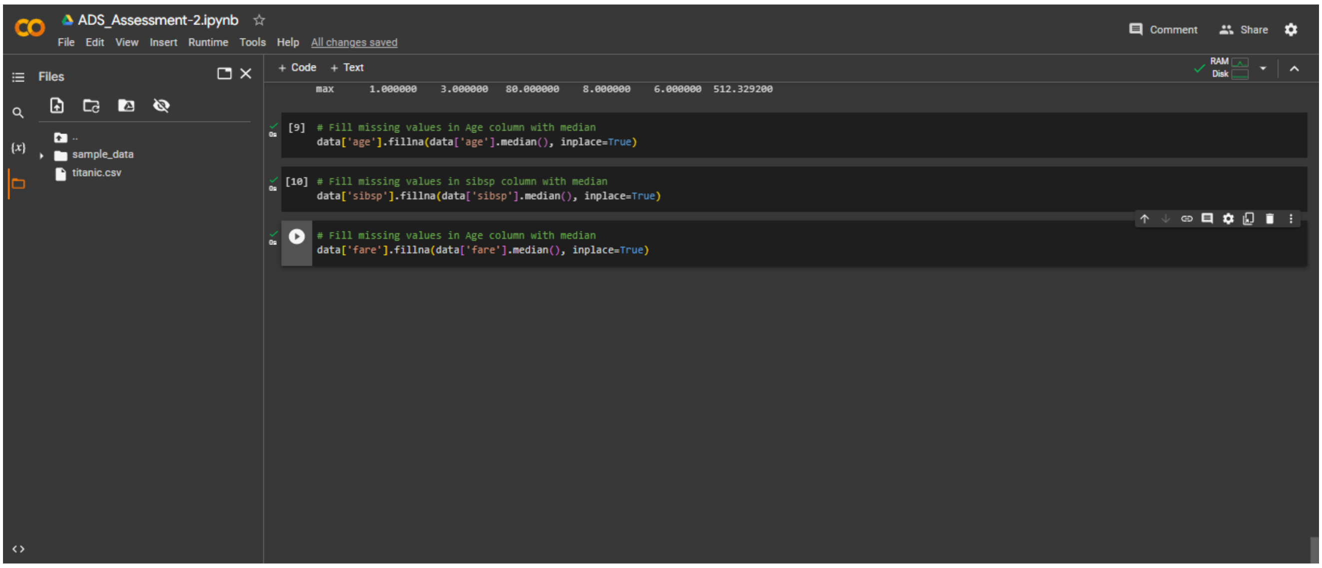
- **Multi - Variate Analysis**



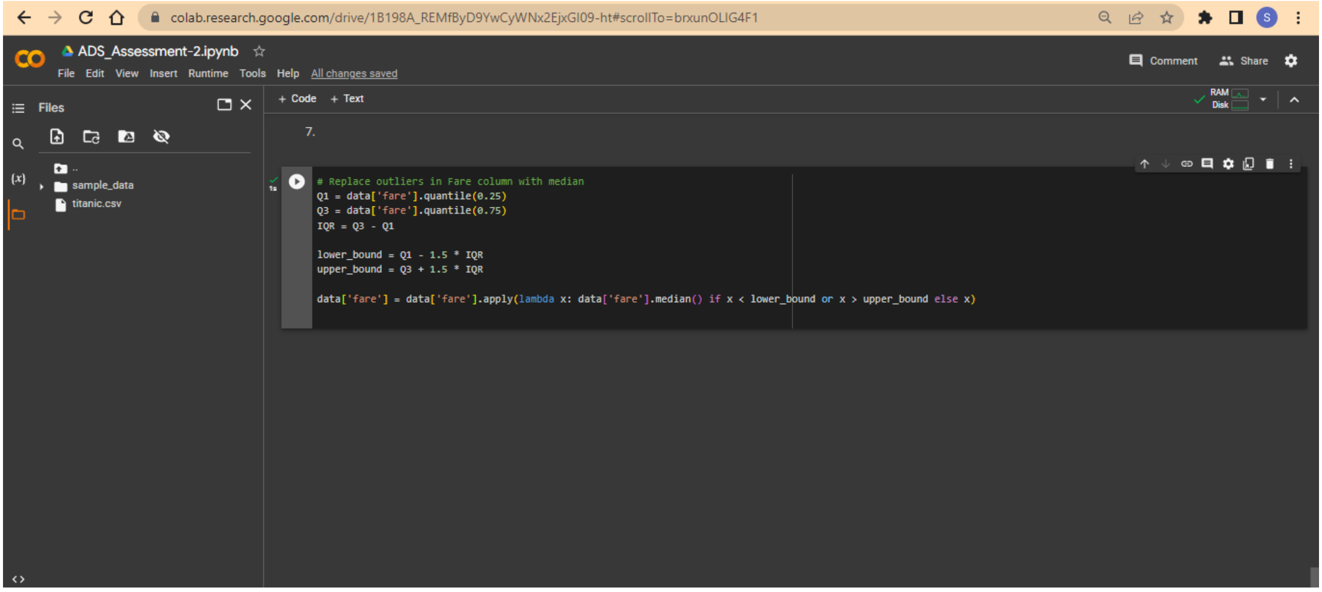
4.Perform descriptive statistics on the dataset.

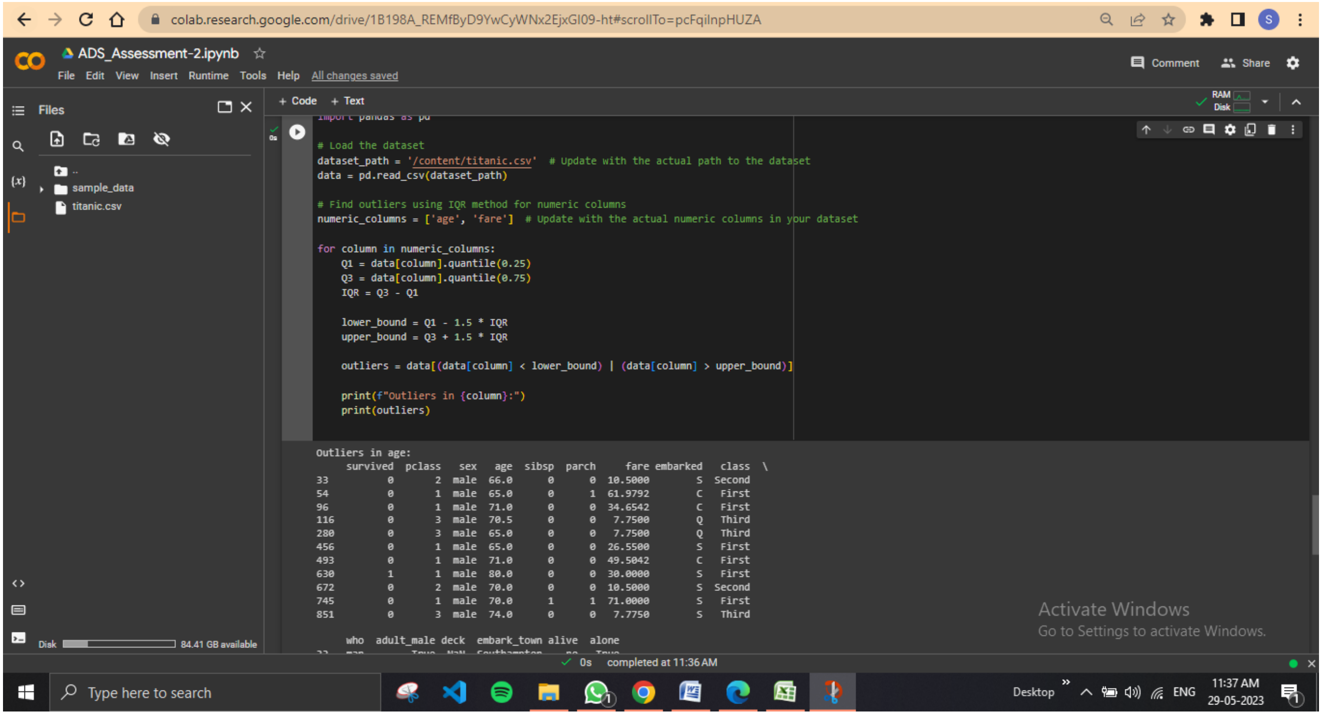
```
print(data.describe())

         survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

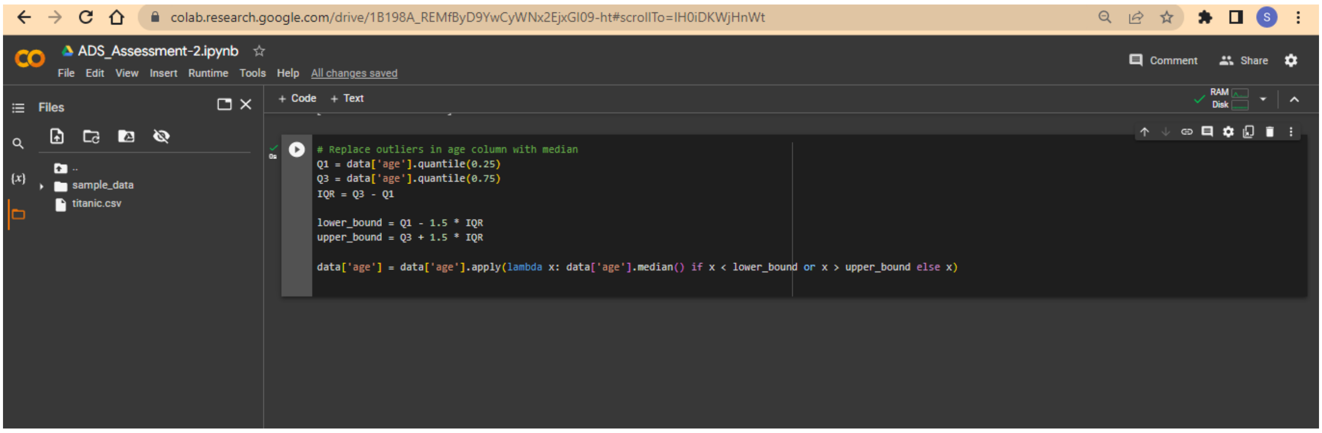
5.Handle the Missing values.



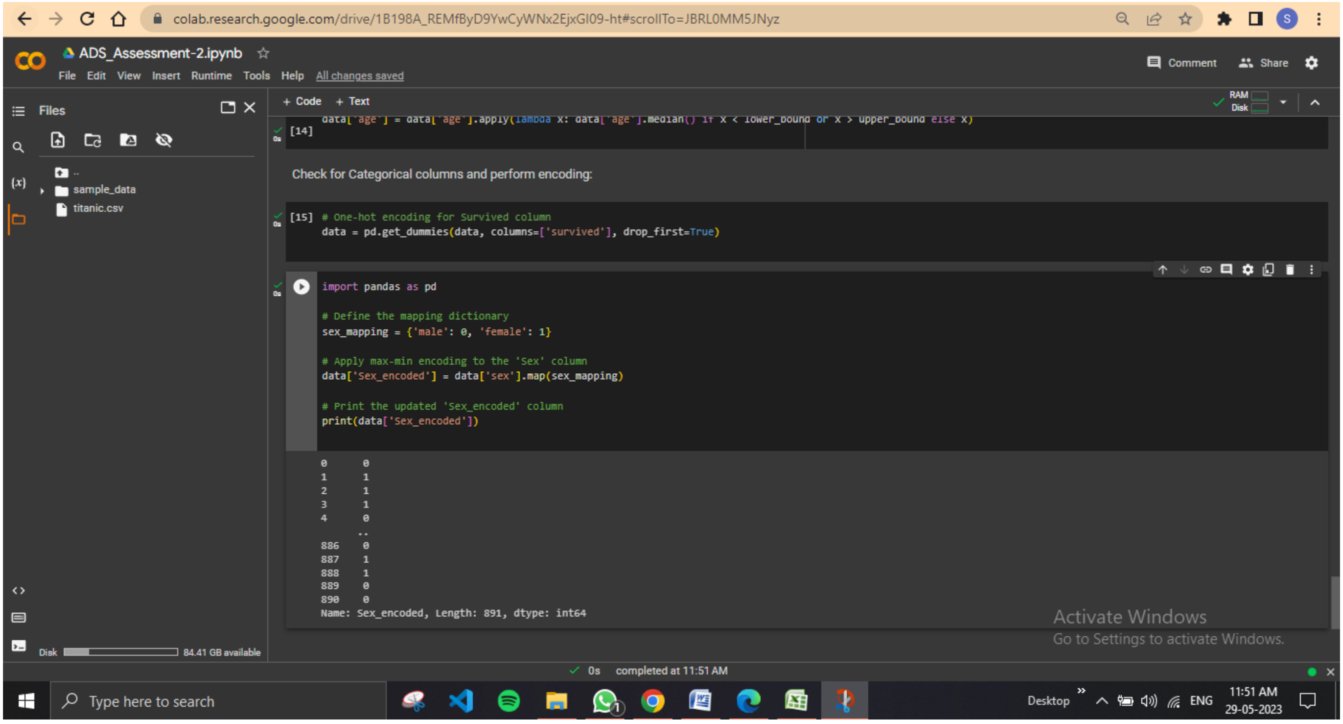
6.Find the outliers and replace the outliers

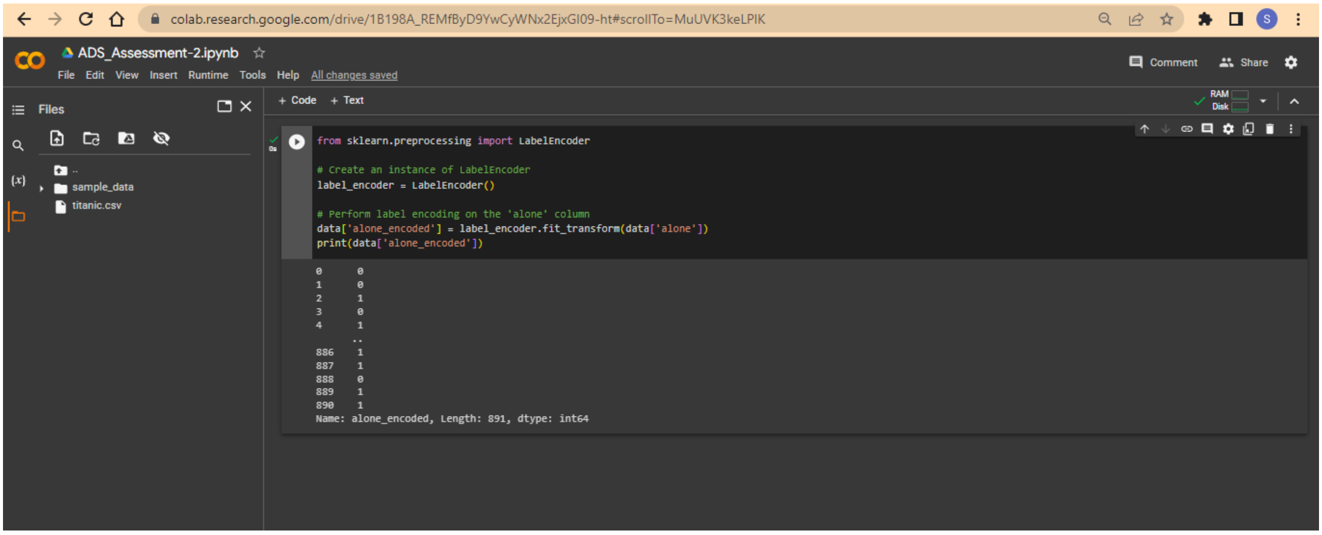




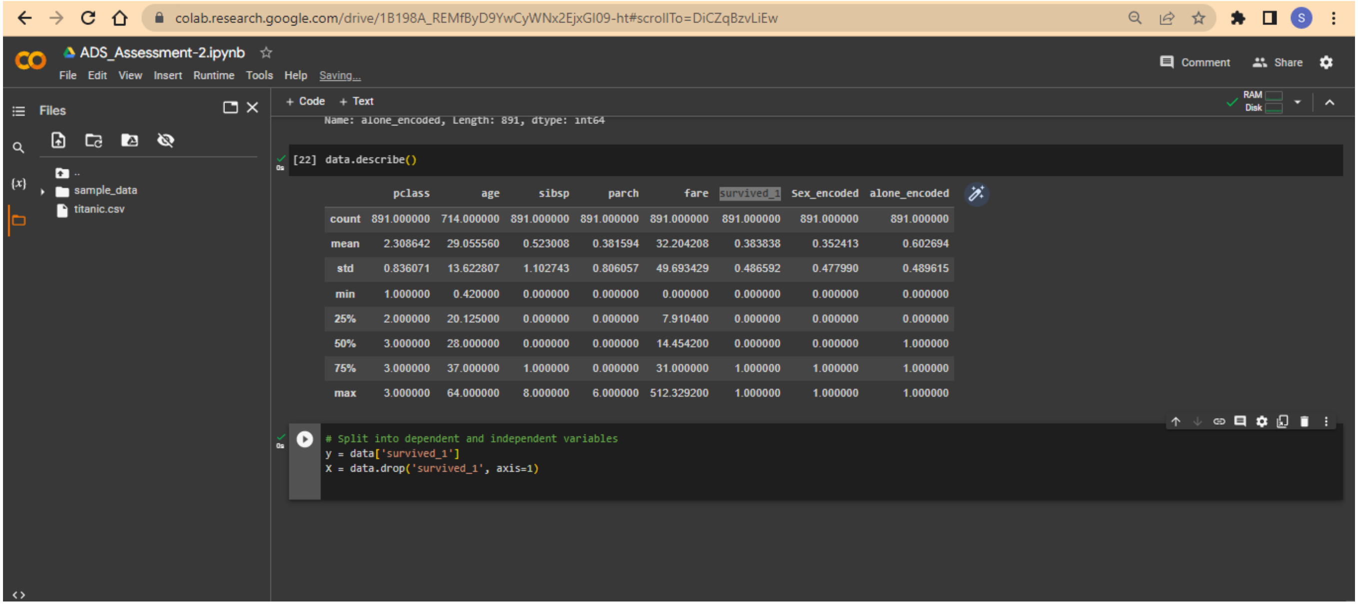


7.Check for Categorical columns and perform encoding.





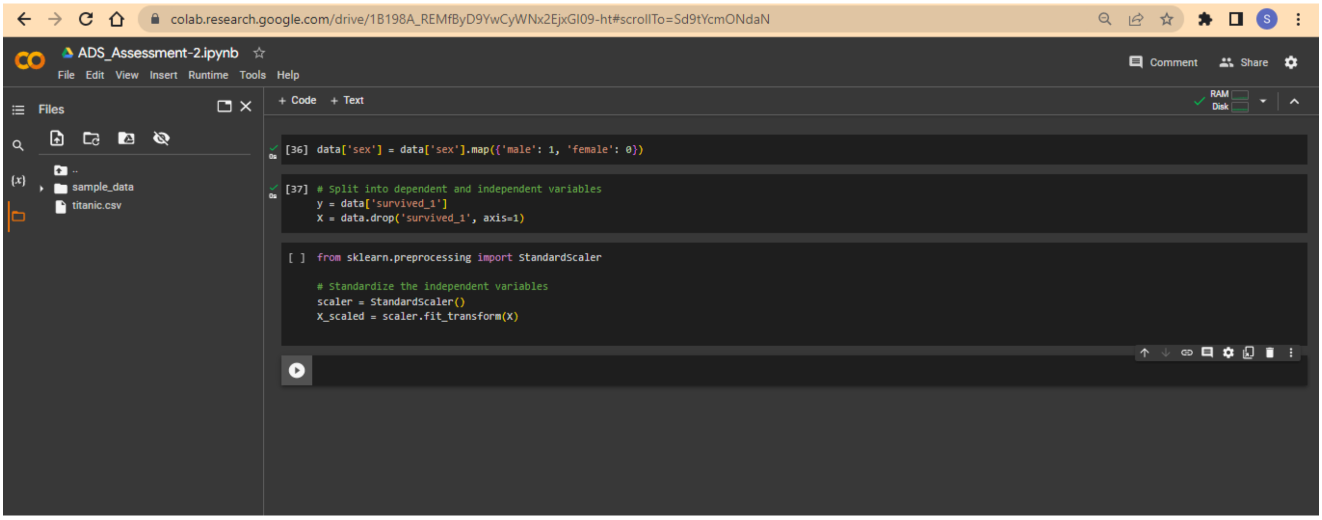
8.Split the data into dependent and independent variables.



	pclass	age	sibsp	parch	fare	survived_1	Sex_encoded	alone_encoded
count	891.000000	714.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	2.308642	29.055560	0.523008	0.381594	32.204208	0.383838	0.352413	0.602694
std	0.836071	13.622807	1.102743	0.806057	49.693429	0.486592	0.477990	0.489615
min	1.000000	0.420000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	20.125000	0.000000	0.000000	7.910400	0.000000	0.000000	0.000000
50%	3.000000	28.000000	0.000000	0.000000	14.454200	0.000000	0.000000	1.000000
75%	3.000000	37.000000	1.000000	0.000000	31.000000	1.000000	1.000000	1.000000
max	3.000000	64.000000	8.000000	6.000000	512.329200	1.000000	1.000000	1.000000

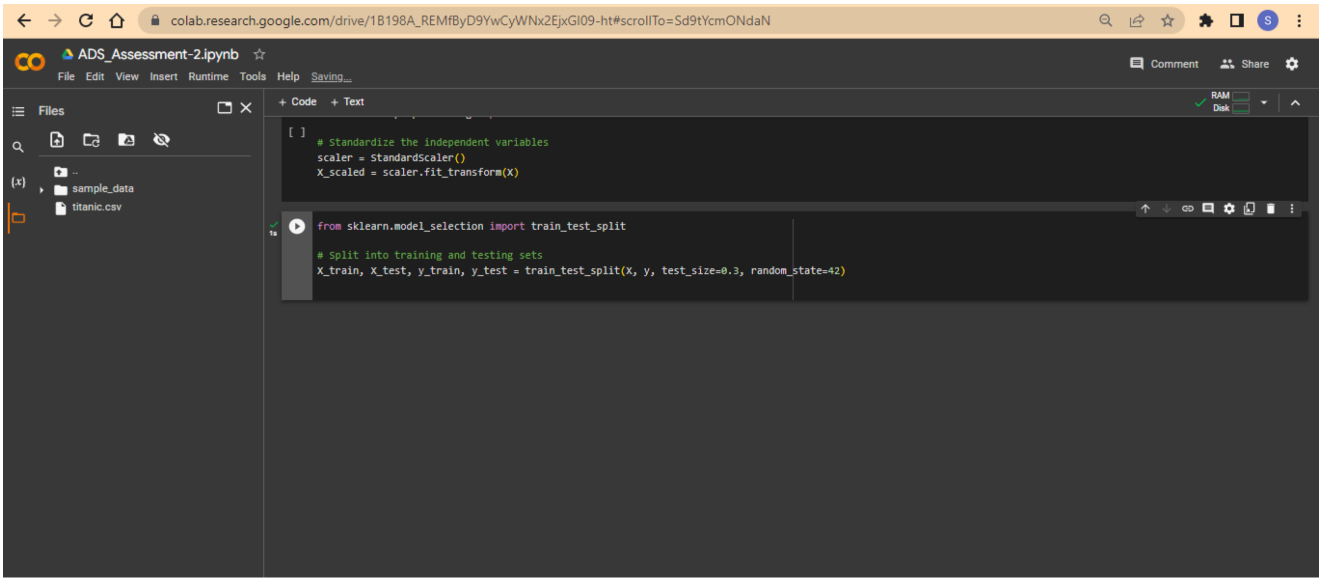
```
# Split into dependent and independent variables
y = data['survived_1']
X = data.drop('survived_1', axis=1)
```

9.Scale the independent variables.




```
[36] data['sex'] = data['sex'].map({'male': 1, 'female': 0})

[37] # Split into dependent and independent variables
     y = data['survived_1']
     X = data.drop('survived_1', axis=1)

[ ]  from sklearn.preprocessing import StandardScaler

     # Standardize the independent variables
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
```

10. Split the data into training and testing




```
[ ]  # Standardize the independent variables
     scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)

from sklearn.model_selection import train_test_split

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```