DASARI.NAGAVENI

EMAIL: dasari.nagaveni2020@vitstudent.ac.in

Vellore institute of technology

Chennai

Assignment-2

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 not

enough 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 upperclass.

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 can

be 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.

Perform Below Tasks to complete the assignment:-

1. Download the dataset: Dataset

2. Load the dataset.

3. Perform Below Visualizations.

- Univariate Analysis
- Bi - Variate Analysis
- Multi - Variate Analysis

4. Perform descriptive statistics on the dataset.

5. Handle the Missing values.

6. Find the outliers and replace the outliers

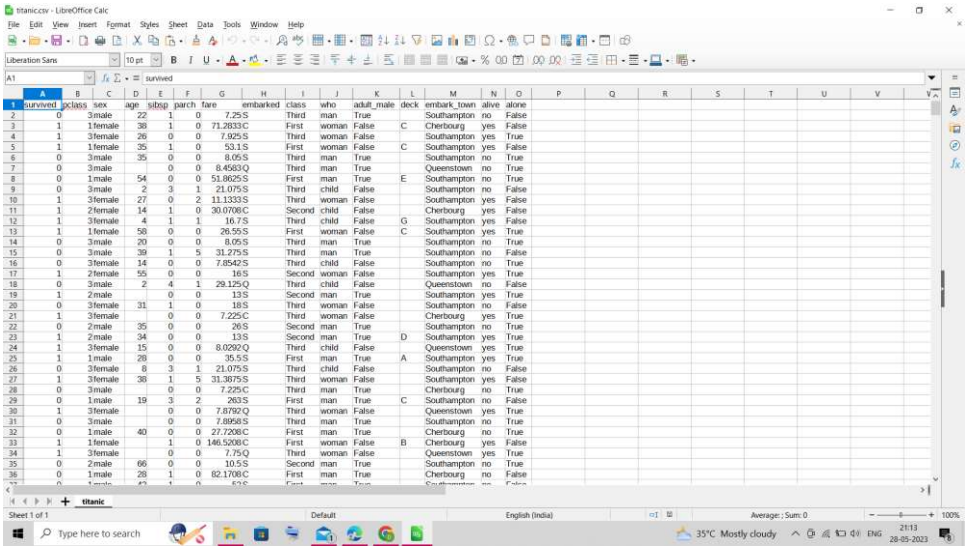
7. Check for Categorical columns and perform encoding.

8. Split the data into dependent and independent variables.

9. Scale the independent variables

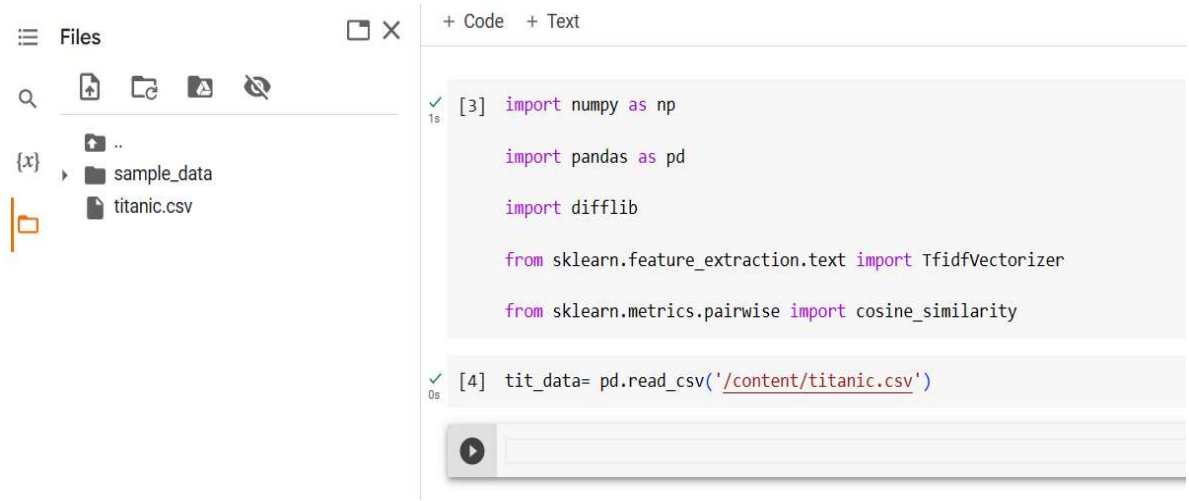
10. **Split the data into training and testing**

[1] Download the dataset: Dataset



survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	3	male	22	1	0	7.25	S	Third	man	True		Southampton	no	False
1	1	female	38	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
1	3	female	26	0	0	7.925	S	Third	woman	False		Southampton	yes	True
1	1	female	35	1	0	53.1	S	First	woman	False	C	Southampton	yes	False
0	3	male	35	0	0	8.05	S	Third	man	True		Southampton	no	True
0	3	male		0	0	8.4583	Q	Third	man	True		Queenstown	no	True
0	1	male	54	0	0	51.8625	S	First	man	True	E	Southampton	no	True
0	3	male	2	3	1	21.075	S	Third	child	False		Southampton	no	False
1	3	female	27	0	2	11.1333	S	Third	woman	False		Southampton	yes	False
1	2	female	14	1	0	30.0708	C	Second	child	False		Cherbourg	yes	False
1	3	female	4	1	1	16.7	S	Third	child	False	G	Southampton	yes	False
1	1	female	58	0	0	26.55	S	First	woman	False	C	Southampton	yes	True
0	3	male	20	0	0	8.05	S	Third	man	True		Southampton	no	True
0	3	male	39	1	5	31.275	S	Third	man	True		Southampton	no	False
0	3	female	14	0	0	7.8542	S	Third	child	False		Southampton	no	True
1	2	female	55	0	0	16	S	Second	woman	False		Southampton	yes	True
0	3	male	2	4	1	29.125	Q	Third	child	False		Queenstown	no	False
1	2	male		0	0	13	S	Second	man	True		Southampton	yes	True
0	3	female	31	1	0	18	S	Third	woman	False		Southampton	no	False
1	3	female		0	0	7.225	C	Third	woman	False		Cherbourg	yes	True
0	2	male	35	0	0	26	S	Second	man	True		Southampton	no	True
1	2	male	34	0	0	13	S	Second	man	True	D	Southampton	yes	True
1	3	female	15	0	0	8.0292	Q	Third	child	False		Queenstown	yes	True
1	1	male	28	0	0	35.5	S	First	man	True	A	Southampton	yes	True
0	3	female	8	3	1	21.075	S	Third	child	False		Southampton	no	False
1	3	female	38	1	5	31.3875	S	Third	woman	False		Southampton	yes	False
0	3	male		0	0	7.225	C	Third	man	True		Cherbourg	no	True
0	1	male	19	3	2	263	S	First	man	True	C	Southampton	no	False
1	3	female		0	0	7.8792	Q	Third	woman	False		Queenstown	yes	True
0	3	male		0	0	7.8958	S	Third	man	True		Southampton	no	True
0	1	male	40	0	0	27.7208	C	First	man	True		Cherbourg	no	True
1	1	female		1	0	146.5208	C	First	woman	False	B	Cherbourg	yes	False
1	3	female		0	0	7.75	Q	Third	woman	False		Queenstown	yes	True
0	2	male	66	0	0	10.5	S	Second	man	True		Southampton	no	True
0	1	male	28	1	0	82.1708	C	First	man	True		Cherbourg	no	False

[2] Load the dataset.



[3] Perform Below Visualizations.

- Univariate Analysis
- Bi - Variate Analysis

- Multi - Variate Analysis

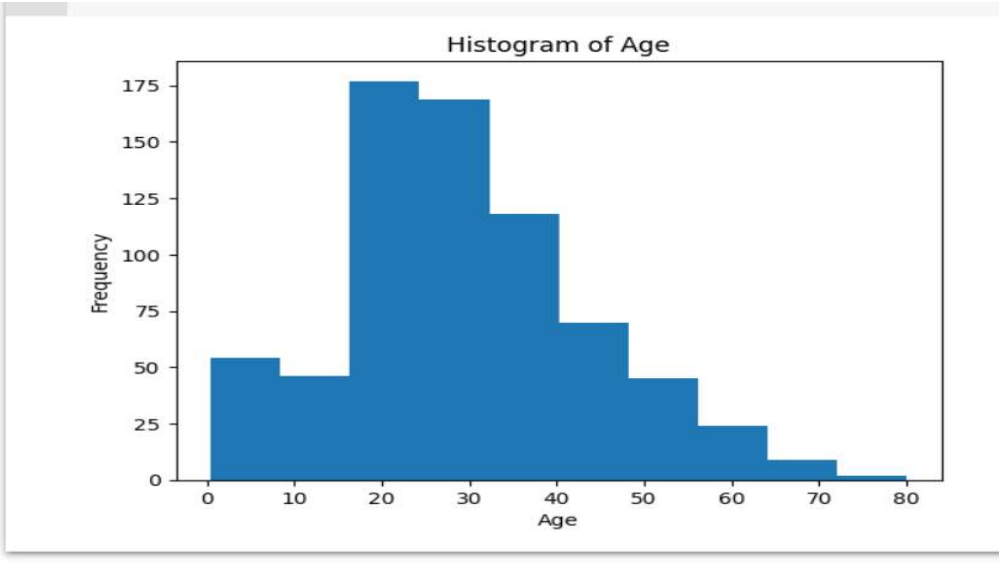
CODE

- Univariate Analysis

HISTOGRAM FOR AGE

```
import matplotlib.pyplot as plt


# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
plt.hist(tit_data['age'].dropna(), bins=10)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age')
plt.show()
```

OUTPUT



BOXPLOT

```
import seaborn as sns


# Assuming your data is in a pandas DataFrame called 'df'
sns.boxplot(x=df['age'])
plt.xlabel('age')
plt.title('Boxplot')
plt.show()
```

OUTPUT

- Bi - Variate Analysis

SCATTER PLOT

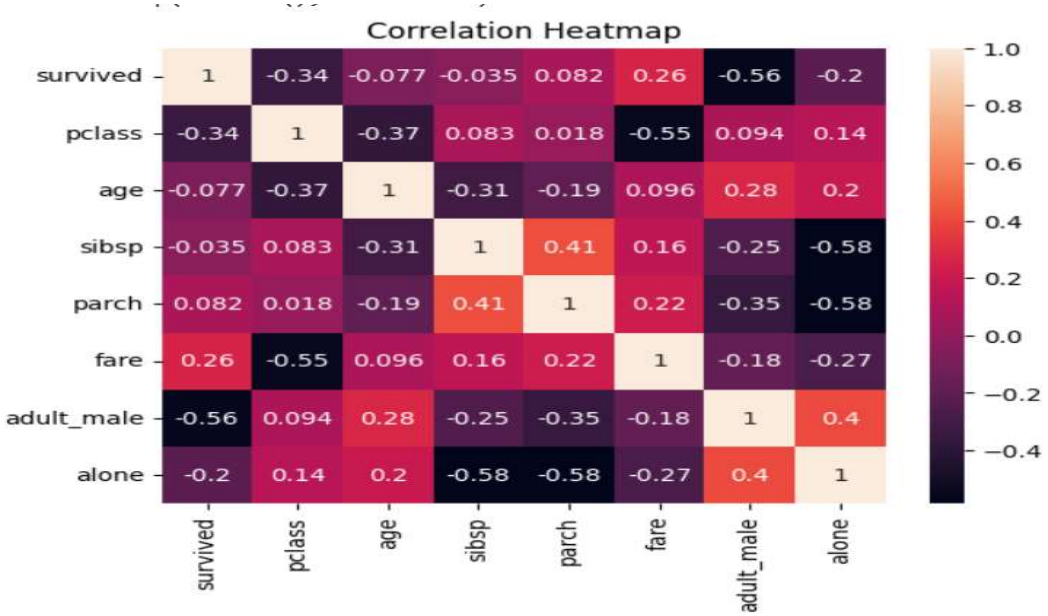
```
import matplotlib.pyplot as plt


# Assuming your data is in a pandas DataFrame called 'df'
plt.scatter(df['age'], df['fare'])
plt.xlabel('age')
plt.ylabel('fare')
plt.title('Scatter Plot')
plt.show()
```

OUTPUT

HEAT MAP

```
import seaborn as sns

# Assuming your data is in a pandas DataFrame called 'df'
sns.heatmap(df.corr(), annot=True)
plt.title('Correlation Heatmap')
plt.show()
```

OUTPUT



- Multi - Variate Analysis

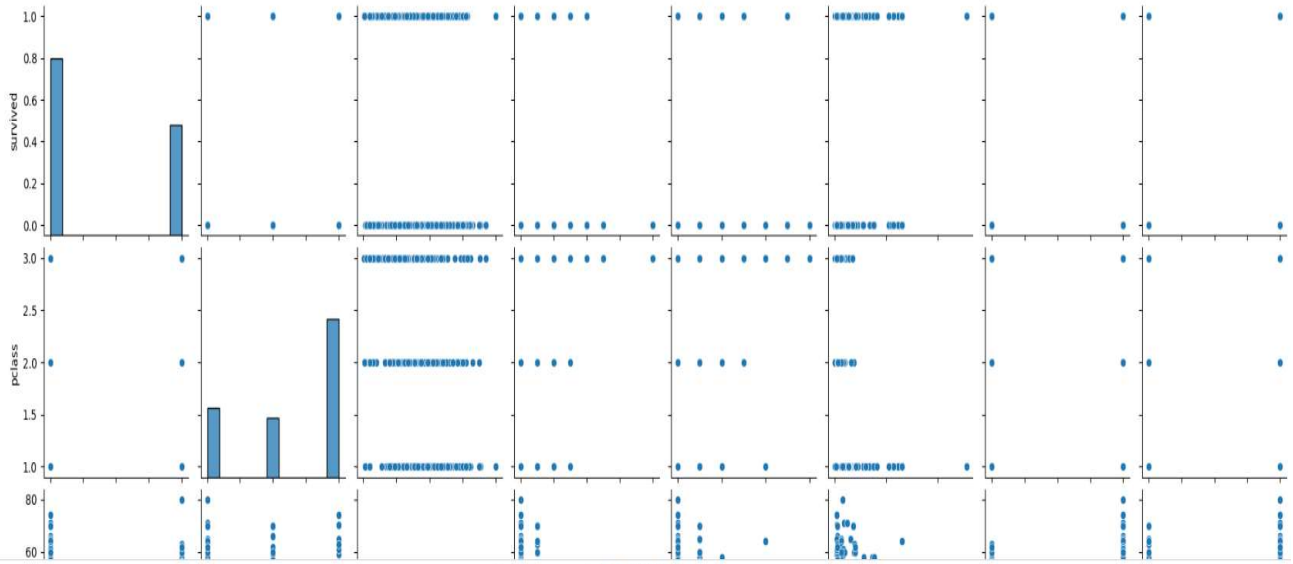
PAIR PLOT

```
import seaborn as sns

# Assuming your data is in a pandas DataFrame called 'df'
sns.pairplot(df)
plt.title('Pairplot')
plt.show()
```

OUTPUT

```
import seaborn as sns

# Assuming your data is in a pandas DataFrame called 'df'
sns.pairplot(df)
plt.title('Pairplot')
plt.show()
```

```
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'> for compatibility.
<__array_function__ internals>:180: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'> for compatibility.
```

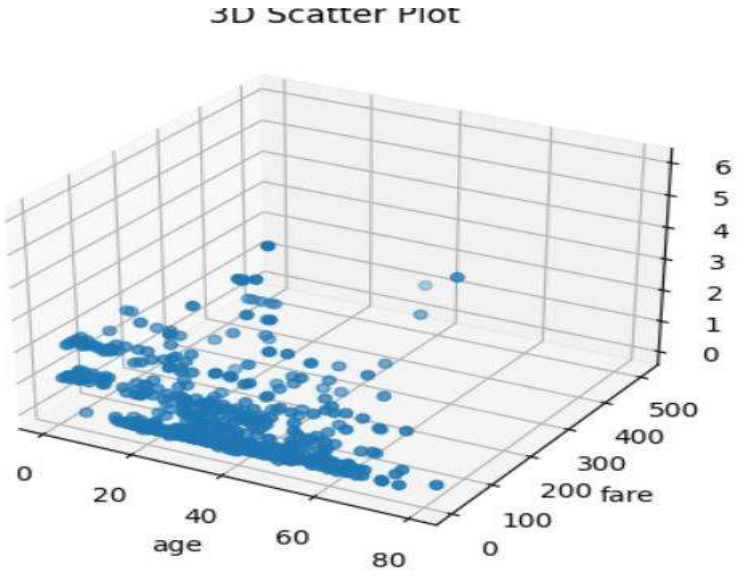


✓ 20s completed at 9:36 PM

3D SCATTER PLOT

```
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D



# Assuming your data is in a pandas DataFrame called 'df'

fig = plt.figure()

ax = fig.add_subplot(111, projection='3d')

ax.scatter(df['age'], df['fare'], df['parch'])

ax.set_xlabel('age')

ax.set_ylabel('fare')

ax.set_zlabel('parch')

plt.title('3D Scatter Plot')
```

```
plt.show()
```



---

[4]

Perform descriptive statistics on the dataset.

```
import pandas as pd


# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Select numeric columns for descriptive statistics
numeric_columns = ['age', 'fare']


# Compute descriptive statistics
statistics = df[numeric_columns].describe()


# Print the descriptive statistics
print(statistics)
```

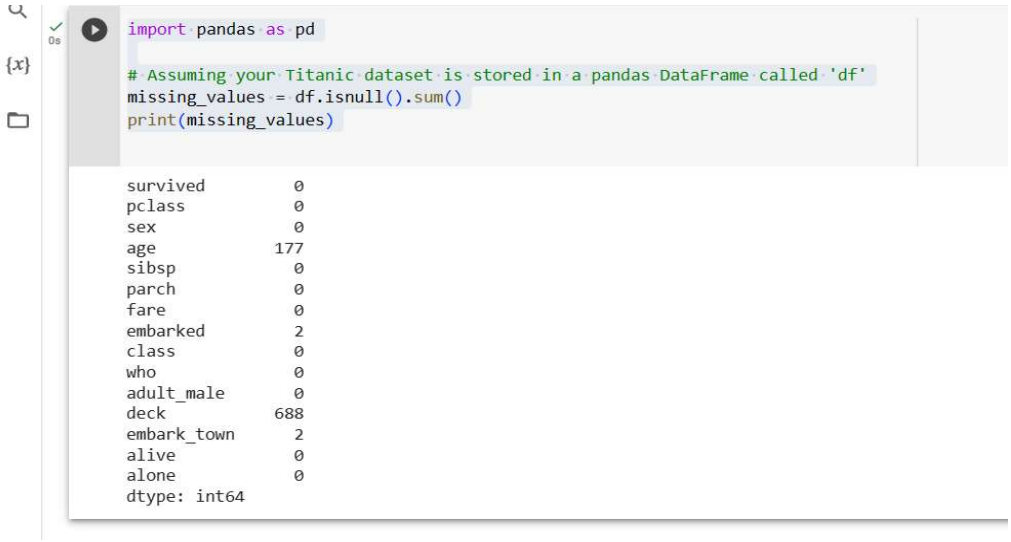
```python
import pandas as pd

# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Select numeric columns for descriptive statistics
numeric_columns = ['age', 'fare']

# Compute descriptive statistics
statistics = df[numeric_columns].describe()

# Print the descriptive statistics
print(statistics)
```

```
              age        fare
count  714.000000  891.000000
mean    29.699118   32.204208
std     14.526497   49.693429
min      0.420000    0.000000
25%     20.125000    7.910400
50%     28.000000   14.454200
75%     38.000000   31.000000
max     80.000000  512.329200
```

[5] Handle the Missing values.

IDENTIFY MISSING VALUES

```
import pandas as pd


# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'

missing_values = df.isnull().sum()

print(missing_values)
```



```
import pandas as pd

# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
missing_values = df.isnull().sum()
print(missing_values)
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

DROP MISSING VALUES

```
# Drop rows with any missing values

df.dropna(inplace=True)
```

IMPUTe MISSING VALUE

```
# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'

mean_age = df['age'].mean()

df['age'].fillna(mean_age, inplace=True)
```

CREATE INDICATOR VARIABLE

[6]

Find the outliers and replace the outliers

IDENTIFY OUTLIERS

```
import numpy as np
```

```
# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
column = 'fare'
threshold = 3  # Set the threshold for outlier detection

z_scores = (df[column] - df[column].mean()) / df[column].std()
outliers = df[np.abs(z_scores) > threshold]
print(outliers)
```

REPLACE OUTLIERS

```
# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
column = 'fare'
median_value = df[column].median()

df.loc[np.abs(z_scores) > threshold, column] = median_value
```

[7] Check for Categorical columns and perform encoding.

IDENTIFYING CATEGORICAL COLUMNS

import pandas as pd

Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'

categorical_columns = df.select_dtypes(include=['object', 'category']).columns

print(categorical_columns)

PERFORM ENCODING

[8]

Split the data into dependent and independent variables.

```
import pandas as pd


# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Identify the dependent variable (target variable)
target_variable = 'alive'


# Identify the independent variables (features)
independent_variables = ['pclass', 'age', 'sex', 'fare']


# Split the data into dependent and independent variables
X = df[independent_variables]
y = df[target_variable]


# Print the shape of the data
print("Independent variables (X):", X.shape)
print("Dependent variable (y):", y.shape)
```

```
[28] one_hot_encoded = pd.get_dummies(df['sex'], prefix='column')
     df = pd.concat([df, one_hot_encoded], axis=1)
```

```
import pandas as pd

# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Identify the dependent variable (target variable)
target_variable = 'alive'

# Identify the independent variables (features)
independent_variables = ['pclass', 'age', 'sex', 'fare']

# Split the data into dependent and independent variables
X = df[independent_variables]
y = df[target_variable]

# Print the shape of the data
print("Independent variables (X):", X.shape)
print("Dependent variable (y):", y.shape)
```

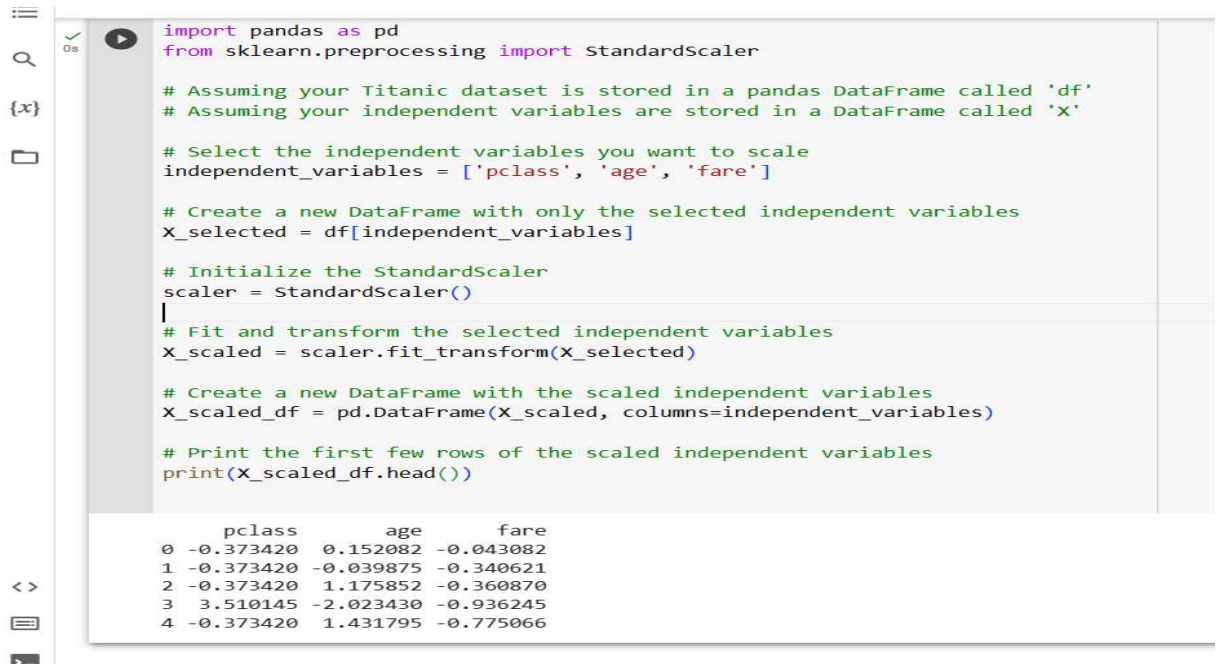
```
Independent variables (X): (182, 4)
Dependent variable (y): (182,)
```

[9]

Scale the independent variables

```
import pandas as pd

from sklearn.preprocessing import StandardScaler


# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'

# Assuming your independent variables are stored in a DataFrame called 'X'


# Select the independent variables you want to scale

independent_variables = ['pclass', 'age', 'fare']


# Create a new DataFrame with only the selected independent variables

X_selected = df[independent_variables]
```

```
# Initialize the StandardScaler

scaler = StandardScaler()


# Fit and transform the selected independent variables

X_scaled = scaler.fit_transform(X_selected)


# Create a new DataFrame with the scaled independent variables

X_scaled_df = pd.DataFrame(X_scaled, columns=independent_variables)


# Print the first few rows of the scaled independent variables

print(X_scaled_df.head())
```



```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Assuming your independent variables are stored in a DataFrame called 'X'

# Select the independent variables you want to scale
independent_variables = ['pclass', 'age', 'fare']

# Create a new DataFrame with only the selected independent variables
X_selected = df[independent_variables]

# Initialize the StandardScaler
scaler = StandardScaler()

# Fit and transform the selected independent variables
X_scaled = scaler.fit_transform(X_selected)

# Create a new DataFrame with the scaled independent variables
X_scaled_df = pd.DataFrame(X_scaled, columns=independent_variables)

# Print the first few rows of the scaled independent variables
print(X_scaled_df.head())
```

```
     pclass       age      fare
0 -0.373420  0.152082 -0.043082
1 -0.373420 -0.039875 -0.340621
2 -0.373420  1.175852 -0.360870
3  3.510145 -2.023430 -0.936245
4 -0.373420  1.431795 -0.775066
```

[10] Split the data into training and testing

```
import pandas as pd

from sklearn.model_selection import train_test_split



# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'

# Assuming your dependent variable is stored in a Series called 'y'

# Assuming your independent variables are stored in a DataFrame called 'X'



# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)



# Print the shape of the training and testing sets

print("X_train shape:", X_train.shape)

print("X_test shape:", X_test.shape)

print("y_train shape:", y_train.shape)

print("y_test shape:", y_test.shape)
```

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Assuming your Titanic dataset is stored in a pandas DataFrame called 'df'
# Assuming your dependent variable is stored in a Series called 'y'
# Assuming your independent variables are stored in a DataFrame called 'X'

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Print the shape of the training and testing sets
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (145, 4)
X_test shape: (37, 4)
y_train shape: (145,)
y_test shape: (37,)
```