In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [11]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [22]:
```
#titanic classification
#build a predictive model to determine the livlihood of survived for passangers on the titanic using data science tecnique in python
```

In [23]:
```
import sandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
import seaborn as sns
import matplotlib.pyplot as plt
```

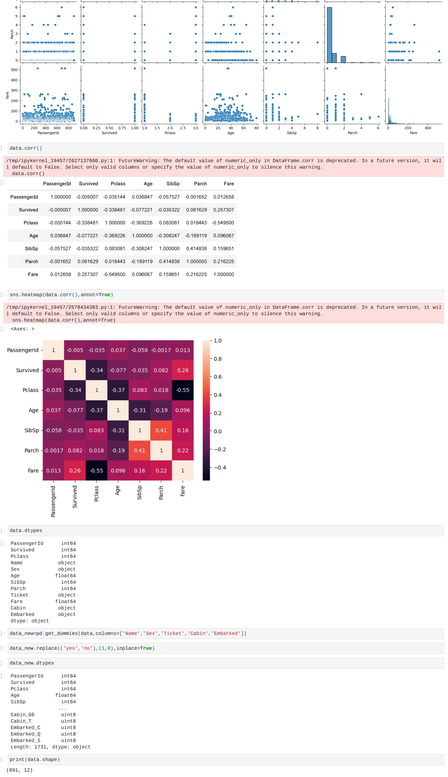
In [29]:
```
data = pd.read_csv('/home/student/titanic-dataset.csv')
data.head()
```

Out[29]:

	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

In [30]:
```
sns.pairplot(data)
```

Out[30]:
```
<seaborn.axisgrid.PairGrid at 0x7f7b60e98900>
```



In [31]:
```
data.corr()
```

```
/tmp/ipykernel_19457/2627137660.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  data.corr()
```

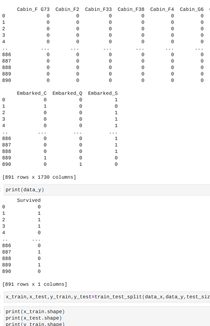
Out[31]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

In [32]:
```
sns.heatmap(data.corr(),annot=True)
```

```
/tmp/ipykernel_19457/2573434383.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.corr(),annot=True)
```

Out[32]:
```
<Axes: >
```



In [33]:
```
data.dtypes
```

Out[33]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [37]:
```
data_new=pd.get_dummies(data,columns=['Name','Sex','Ticket','Cabin','Embarked'])
```

In [39]:
```
data_new.replace({'yes','no'},{1,0},inplace=True)
```

In [40]:
```
data_new.dtypes
```

Out[40]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Age            float64
SibSp            int64
               ...   
Cabin_G6         uint8
Cabin_T          uint8
Embarked_C       uint8
Embarked_Q       uint8
Embarked_S       uint8
Length: 1731, dtype: object
```

In [41]:
```
print(data.shape)
```

```
(891, 12)
```

In [42]:
```
data.Survived.unique()
```

Out[42]:
```
array([0, 1])
```

In [49]:
```
data_y=pd.DataFrame(data_new['Survived'])
```

In [51]:
```
data_x=data_new.drop(['Survived'],axis=1)
```

In [53]:
```
print(data_x)
```

```
     PassengerId  Pclass   Age  SibSp  Parch     Fare  \
0              1       3  22.0      1      0   7.2500   
1              2       1  38.0      1      0  71.2833   
2              3       3  26.0      0      0   7.9250   
3              4       1  35.0      1      0  53.1000   
4              5       3  35.0      0      0   8.0500   
..           ...     ...   ...    ...    ...      ...   
886          887       2  27.0      0      0  13.0000   
887          888       1  19.0      0      0  30.0000   
888          889       3   NaN      1      2  23.4500   
889          890       1  26.0      0      0  30.0000   
890          891       3  32.0      0      0   7.7500   

     Name_Abbing, Mr. Anthony  Name_Abbott, Mr. Rossmore Edward  \
0                           0                                 0   
1                           0                                 0   
2                           0                                 0   
3                           0                                 0   
4                           0                                 0   
..                        ...                               ...   
886                         0                                 0   
887                         0                                 0   
888                         0                                 0   
889                         0                                 0   
890                         0                                 0   

     Name_Abbott, Mrs. Stanton (Rosa Hunt)  Name_Abelson, Mr. Samuel  ...  \
0                                        0                         0  ...   
1                                        0                         0  ...   
2                                        0                         0  ...   
3                                        0                         0  ...   
4                                        0                         0  ...   
..                                     ...                       ...  ...   
886                                      0                         0  ...   
887                                      0                         0  ...   
888                                      0                         0  ...   
889                                      0                         0  ...   
890                                      0                         0  ...   

     Cabin_F G73  Cabin_F2  Cabin_F33  Cabin_F38  Cabin_F4  Cabin_G6  Cabin_T  \
0              0         0          0          0         0         0        0   
1              0         0          0          0         0         0        0   
2              0         0          0          0         0         0        0   
3              0         0          0          0         0         0        0   
4              0         0          0          0         0         0        0   
..           ...       ...        ...        ...       ...       ...      ...   
886            0         0          0          0         0         0        0   
887            0         0          0          0         0         0        0   
888            0         0          0          0         0         0        0   
889            0         0          0          0         0         0        0   
890            0         0          0          0         0         0        0   

     Embarked_C  Embarked_Q  Embarked_S  
0             0           0           1  
1             1           0           0  
2             0           0           1  
3             0           0           1  
4             0           0           1  
..          ...         ...         ...  
886           0           0           1  
887           0           0           1  
888           0           0           1  
889           1           0           0  
890           0           1           0  

[891 rows x 1730 columns]
```

In [54]:
```
print(data_y)
```

```
     Survived
0           0
1           1
2           1
3           1
4           0
..        ...
886         0
887         1
888         0
889         1
890         0

[891 rows x 1 columns]
```

In [55]:
```
x_train,x_test,y_train,y_test=train_test_split(data_x,data_y,test_size=0.3,random_state=2,stratify=data_y)
```

In [58]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(623, 1730)
(268, 1730)
(623, 1)
(268, 1)
```

In [66]:
```
from sklearn.impute import SimpleImputer


imputer = SimpleImputer(strategy='mean')
x_train_imputed = imputer.fit_transform(x_train)
x_test_imputed = imputer.transform(x_test)

model = LogisticRegression()
model.fit(x_train_imputed, y_train)

predictions = model.predict(x_test_imputed)

accuracy = accuracy_score(y_test, predictions)
print('Accuracy:', accuracy)
```

```
/home/student/anaconda3/lib/python3.10/site-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1 d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
Accuracy: 0.7611940298507462
/home/student/anaconda3/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: