```
import pandas as pd
import numpy as np
import seaborn as sb
```

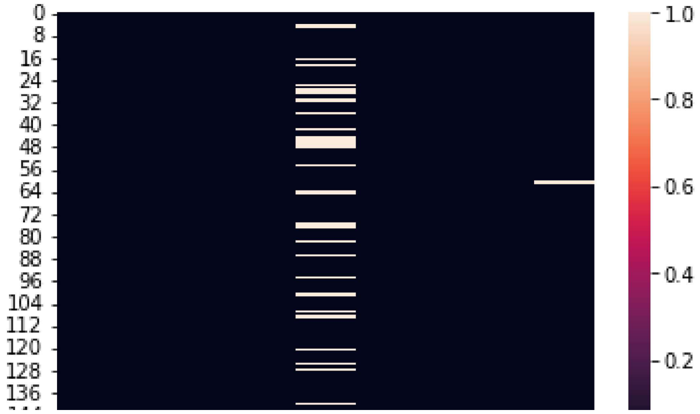
```
# Read the dataset into a dataframe
df = pd.read_csv('/content/titanic (2).csv')
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.

```
# Drop some columns which is not relevant to the analysis (they are not numeric)
cols_to_drop = ['Name', 'Ticket', 'Cabin']
df = df.drop(cols_to_drop, axis=1)
```

```
df.info()
sb.heatmap(df.isnull())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  156 non-null    int64  
 1   Survived     156 non-null    int64  
 2   Pclass       156 non-null    int64  
 3   Sex          156 non-null    object 
 4   Age          126 non-null    float64
 5   SibSp        156 non-null    int64  
 6   Parch        156 non-null    int64  
 7   Fare         156 non-null    float64
 8   Embarked     155 non-null    object 
dtypes: float64(2), int64(5), object(2)
memory usage: 11.1+ KB
<matplotlib.axes._subplots.AxesSubplot at 0x7f09c7f21c50>
```



```
# To replace missing values with interpolated values, for example Age
df['Age'] = df['Age'].interpolate()


# Drop all rows with missin data
df = df.dropna()


# First, create dummy columns from the Embarked and Sex columns
EmbarkedColumnDummy = pd.get_dummies(df['Embarked'])
SexColumnDummy = pd.get_dummies(df['Sex'])


df = pd.concat((df, EmbarkedColumnDummy, SexColumnDummy), axis=1)


# Drop the redundant columns thus converted
df = df.drop(['Sex','Embarked'],axis=1)


# Seperate the dataframe into X and y data
X = df.values
y = df['Survived'].values

# Delete the Survived column from X
X = np.delete(X,1,axis=1)
```

```
X = np.delete(X,1,axis=1)


 #Split the dataset into 70% Training and 30% Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=0)


# Using simple Decision Tree classifier
from sklearn import tree
dt_clf = tree.DecisionTreeClassifier(max_depth=5)
dt_clf.fit(X_train, y_train)
dt_clf.score(X_test, y_test)

     0.7872340425531915


y_pred = dt_clf.predict(X_test)

from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)

     array([[31,  4],
            [ 6,  6]])


#Perform Classification Using Random Forest Classifier
from sklearn import ensemble
rf_clf = ensemble.RandomForestClassifier(n_estimators=100)
rf_clf.fit(X_train, y_train)
rf_clf.score(X_test, y_test)

Output: 0.7368421052631579


#This classifier is available in the ensemble module which we already imported. So we don’t r
gb_clf = ensemble.GradientBoostingClassifier()
gb_clf.fit(X_train, y_train)
gb_clf.score(X_test, y_test)

     0.7021276595744681


# Let's  tune this Gradient booster.
gb_clf = ensemble.GradientBoostingClassifier(n_estimators=50)
gb_clf.fit(X_train,y_train)
gb_clf.score(X_test, y_test)

     0.7446808510638298
```

Colab paid products - Cancel contracts here

✓ 0s completed at 9:40 PM ● ×