```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

train = pd.read_csv('titanic_train.csv')
train.isnull()

sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')

sns.set_style('whitegrid')
sns.countplot(x='Survived',data=train)

sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Gender',data=train,palette='RdBu_r')

sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=train,palette='rainbow')

sns.distplot(train['Age'].dropna(),kde=False,color='darkred',bins=40)

train['Age'].hist(bins=30,color='darkred',alpha=0.3)
sns.countplot(x='SibSp',data=train)

train['Fare'].hist(color='green',bins=40,figsize=(8,4))

plt.figure(figsize=(12, 7))
sns.boxplot(x='Pclass',y='Age',data=train,palette='winter')
```

```
def impute_age(cols):
    Age = cols[0]
    Pclass = cols[1]

    if pd.isnull(Age):

        if Pclass == 1:
            return 37

        elif Pclass == 2:
            return 29

        else:
            return 24

    else:
        return Age



train['Age'] = train[['Age','Pclass']].apply(impute_age,axis=1)

sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')

train.drop('Cabin',axis=1,inplace=True)
train.dropna(inplace=True)
train.head()
train.info()
pd.get_dummies(train['Embarked'],drop_first=True).head()
sex = pd.get_dummies(train['Gender'],drop_first=True)
embark = pd.get_dummies(train['Embarked'],drop_first=True)
train.drop(['Gender','Embarked','Name','Ticket'],axis=1,inplace=True)
train.head()


train = pd.concat([train,sex,embark],axis=1)
train.head()


train.drop('Survived',axis=1).head()

train['Survived'].head()

from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(train.drop('Survived',axis=1),
                             train['Survived'], test_size=0.30,
                             random_state=101)


from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

```
predictions = logmodel.predict(X_test)

from sklearn.metrics import confusion_matrix

accuracy=confusion_matrix(y_test,predictions)


accuracy



from sklearn.metrics import accuracy_score
accuracy


predictions


from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```