

BHARAT INTERN TASK 2- TITANIC CLASSIFICATION

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

D:\Users\Namitha CV\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.24.3)
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")

In [2]: df=pd.read_csv("C:/Users/Namitha CV/Downloads/Titanic-Dataset.csv")
df.head()
```

	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	Cummings, 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 [3]: df.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   PassengerId         891 non-null    int64
 1   Survived            891 non-null    int64
 2   Pclass              891 non-null    int64
 3   Name                891 non-null    object
 4   Sex                 891 non-null    object
 5   Age                 714 non-null    float64
 6   SibSp               891 non-null    int64
 7   Parch              891 non-null    int64
 8   Ticket              891 non-null    object
 9   Fare                891 non-null    float64
10  Cabin               294 non-null    object
11  Embarked            889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
In [5]: df.nunique()
```

```
PassengerId    891
Survived         2
Pclass          3
Name            891
Sex              2
Age             88
SibSp           7
Parch           7
Ticket         681
Fare           248
Cabin          147
Embarked        3
dtype: int64
```

```
In [6]: df.isnull().sum()
```

```
PassengerId    0
Survived        0
Pclass          0
Name            0
Sex             0
Age           177
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin         687
Embarked        2
dtype: int64
```

```
In [7]: df.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
In [8]: ## Clean continuous variables
# Fill missing for 'Age'
df['Age'].fillna(df['Age'].mean(), inplace=True)
```

```
In [9]: df['Family_cnt'] = df['SibSp'] + df['Parch']
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S
...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	S
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	NaN	S

891 rows x 13 columns

```
In [10]: df.drop(["PassengerId", "SibSp", "Parch"], axis=1, inplace=True)
```

```
In [11]: df.groupby(df['Cabin']).isnull()[['Survived']].mean()
df['Cabin_d']=np.where(df['Cabin'].isnull(),0,1)
```

```
In [12]: # Convert 'Sex' to numeric
gender = {'male': 0, 'female': 1}
df['Sex'] = df['Sex'].map(gender)
```

```
In [13]: df.drop(['Cabin', 'Embarked', 'Name', 'Ticket'], axis=1, inplace=True)
```

```
In [14]: df.isnull().sum()
```

```
Survived    0
Pclass      0
Sex          0
Age         0
Fare        0
Family_cnt  0
Cabin_d     0
dtype: int64
```

```
In [15]: #after dropping
df.columns
```

```
Index(['Survived', 'Pclass', 'Sex', 'Age', 'Fare', 'Family_cnt', 'Cabin_d'], dtype='object')
```

```
In [16]: ## Write out cleaned data
df.to_csv('titanic_cleaned.csv', index=False)
```

```
In [17]: from sklearn.model_selection import train_test_split
```

```
In [18]: df = pd.read_csv('titanic_cleaned.csv')
```

```
In [19]: features=df.drop("Survived",axis=1)
labels =df['Survived']
```

```
In [20]: features
```

	Pclass	Sex	Age	Fare	Family_cnt	Cabin_d
0	3	0	22.000000	7.2500	1	0
1	1	1	38.000000	71.2833	1	1
2	3	1	26.000000	7.9250	0	0
3	1	1	35.000000	53.1000	1	1
4	3	0	35.000000	8.0500	0	0
...
886	2	0	27.000000	13.0000	0	0
887	1	1	19.000000	30.0000	0	1
888	3	1	29.699118	23.4500	3	0
889	1	0	26.000000	30.0000	0	1
890	3	0	32.000000	7.7500	0	0

891 rows x 6 columns

```
In [21]: labels
```

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

Name: Survived, Length: 891, dtype: int64

```
In [22]: X_train,X_test,Y_train,Y_test = train_test_split(features,labels,test_size=0.3)
for dataset in [Y_train, Y_test]:
    print(dataset,len(dataset) / len(labels)*100, 2))
```

```
69.92
30.08
```

```
In [23]: X_train
```

	Pclass	Sex	Age	Fare	Family_cnt	Cabin_d
554	3	1	22.000000	7.7750	0	0
882	3	1	22.000000	10.5167	0	0
37	3	0	21.000000	8.0500	0	0
364	3	0	29.699118	15.5000	1	0
7	3	0	2.000000	21.0750	4	0
...
148	2	0	36.500000	26.0000	2	1
319	1	1	40.000000	134.5000	2	1
844	3	0	17.000000	8.6625	0	0
758	3	0	34.000000	8.0500	0	0
743	3	0	24.000000	16.1000	1	0

623 rows x 6 columns

```
In [24]: Y_train
```

554	1
882	0
37	0
364	0
7	0
...	...
148	0
319	1
844	0
758	0
743	0

Name: Survived, Length: 623, dtype: int64

```
In [25]: #Write out all data
X_train.to_csv('train.csv', index=False)
X_test.to_csv('test.csv', index=False)

Y_train.to_csv('train_labels.csv', index=False, header=False)
Y_test.to_csv('test_labels.csv', index=False, header=False)
```

```
In [26]: train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
```

```
In [27]: train.head()
```

	Pclass	Sex	Age	Fare	Family_cnt	Cabin_d
0	3	1	22.000000	7.7750	0	0
1	3	1	22.000000	10.5167	0	0
2	3	0	21.000000	8.0500	0	0
3	3	0	29.699118	15.5000	1	0
4	3	0	2.000000	21.0750	4	0

```
In [28]: #Logistic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train,Y_train)
train_pred = lr.predict(X_train)
test_pred = lr.predict(X_test)
```

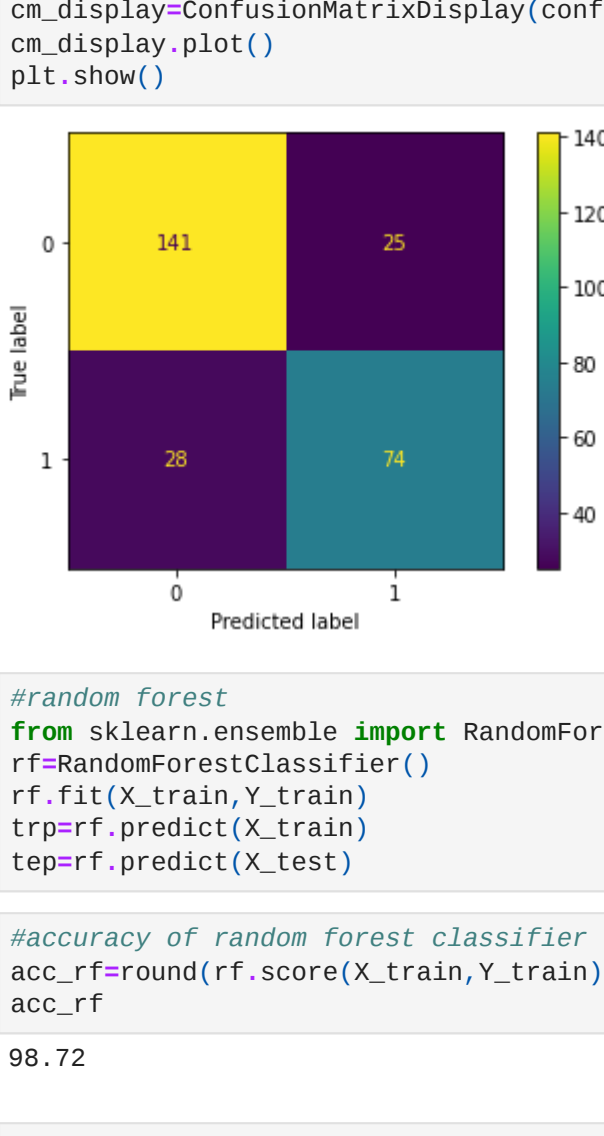
```
In [29]: #accuracy of Logistic Regression
from sklearn.metrics import accuracy_score
acc_log=round(lr.score(X_train,Y_train)*100,2)
acc_log
```

```
80.42
```

```
In [30]: #confusion matrix
from sklearn.metrics import confusion_matrix
cm1=confusion_matrix(Y_test,test_pred)
```

```
Out[30]: array([[145, 21],
        [ 31, 71]], dtype=int64)
```

```
In [31]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm_display=ConfusionMatrixDisplay(confusion_matrix=cm1)
cm_display.plot()
plt.show()
```



```
In [32]: #decision trees
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train,Y_train)
trainpr=dt.predict(X_train)
testpr=dt.predict(X_test)
```

```
In [33]: #accuracy of decision tree
acc_dt=round(dt.score(X_train,Y_train)*100,2)
acc_dt
```

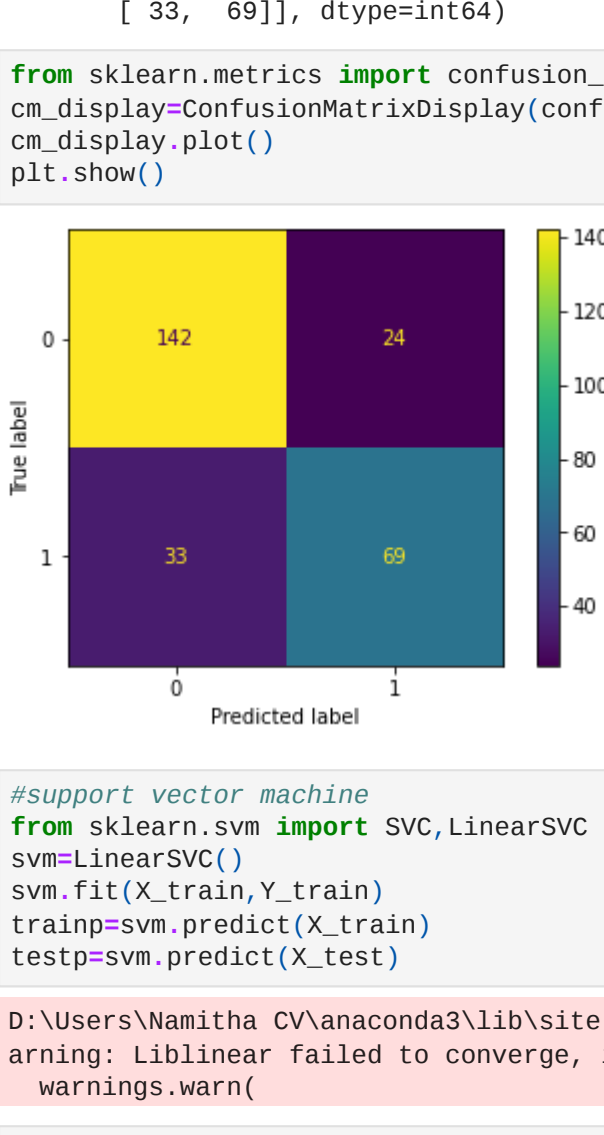
```
98.72
```

```
Out[33]: 98.72
```

```
In [34]: #confusion matrix
from sklearn.metrics import confusion_matrix
cm2=confusion_matrix(Y_test,testpr)
```

```
Out[34]: array([[141, 25],
        [ 28, 74]], dtype=int64)
```

```
In [35]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm_display=ConfusionMatrixDisplay(confusion_matrix=cm2)
cm_display.plot()
plt.show()
```



```
In [36]: #random forest
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()
rf.fit(X_train,Y_train)
trp=rf.predict(X_train)
tep=rf.predict(X_test)
```

```
In [37]: #accuracy of random forest classifier
acc_rf=round(rf.score(X_train,Y_train)*100,2)
acc_rf
```

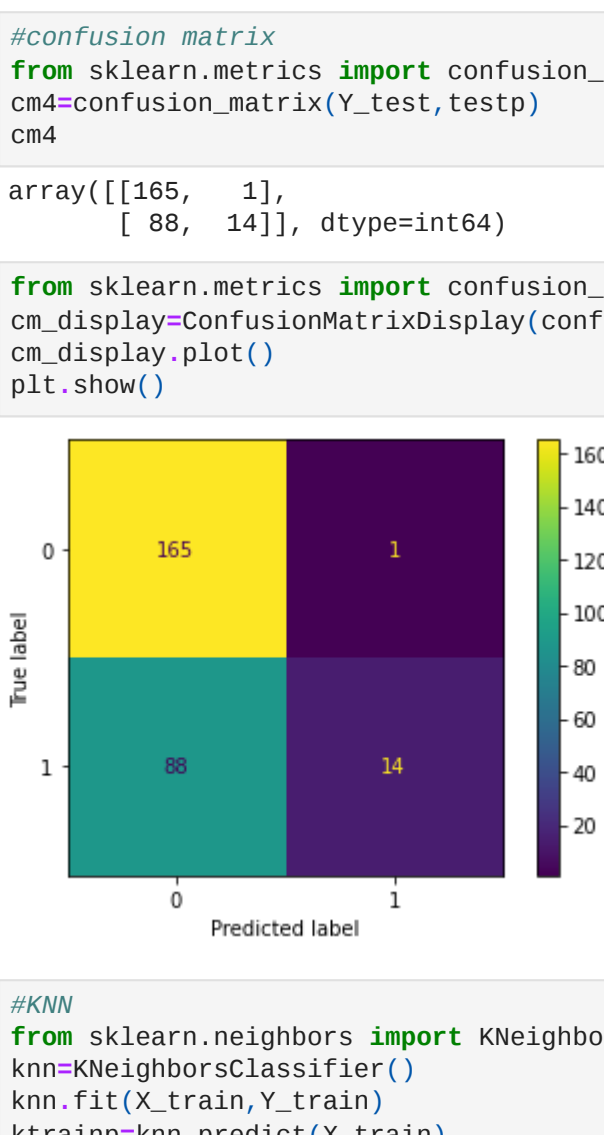
```
98.72
```

```
Out[37]: 98.72
```

```
In [38]: #confusion matrix
from sklearn.metrics import confusion_matrix
cm3=confusion_matrix(Y_test,tep)
```

```
Out[38]: array([[142, 24],
        [ 33, 69]], dtype=int64)
```

```
In [39]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm_display=ConfusionMatrixDisplay(confusion_matrix=cm3)
cm_display.plot()
plt.show()
```



```
In [40]: #support vector machine
from sklearn.svm import SVC,LinearSVC
svm=LinearSVC()
svm.fit(X_train,Y_train)
trainp=svm.predict(X_train)
testp=svm.predict(X_test)
```

D:\Users\Namitha CV\anaconda3\lib\site-packages\sklearn\svm_base.py:1206: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.

```
warnings.warn(
```

```
In [41]: #accuracy of svm
acc_svm=round(svm.score(X_train,Y_train)*100,2)
acc_svm
```

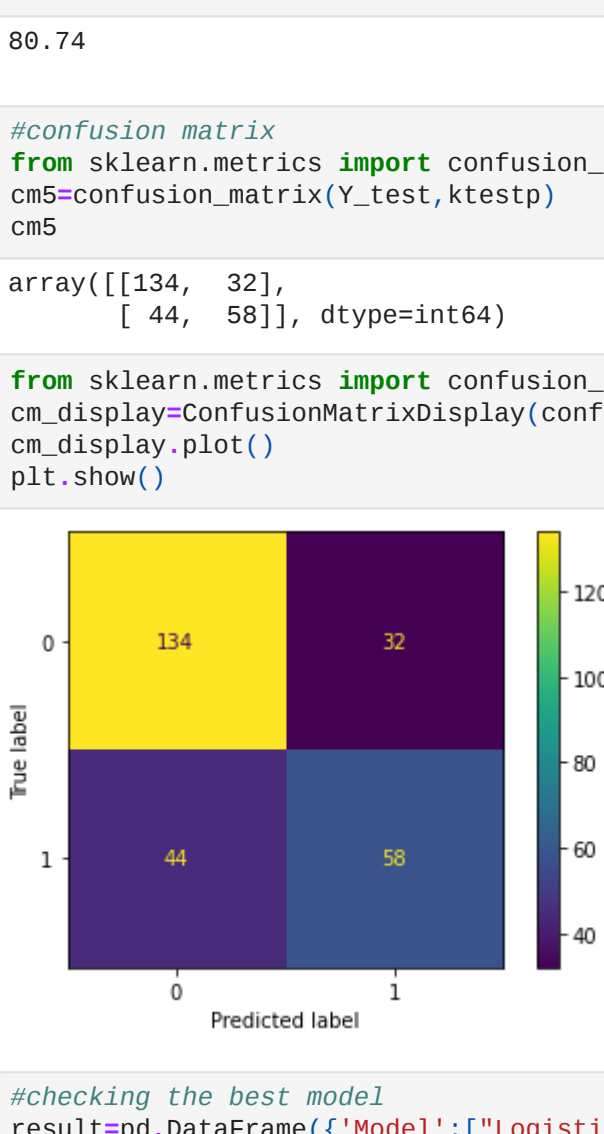
```
66.29
```

```
Out[41]: 66.29
```

```
In [42]: #confusion matrix
from sklearn.metrics import confusion_matrix
cm4=confusion_matrix(Y_test,testp)
```

```
Out[42]: array([[165, 11],
        [ 88, 14]], dtype=int64)
```

```
In [43]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm_display=ConfusionMatrixDisplay(confusion_matrix=cm4)
cm_display.plot()
plt.show()
```



```
In [44]: #KNN
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(X_train,Y_train)
ktrainp=knn.predict(X_train)
ktestp=knn.predict(X_test)
```

```
In [45]: #accuracy of KNN
acc_knn= round(knn.score(X_train,Y_train)*100,2)
acc_knn
```

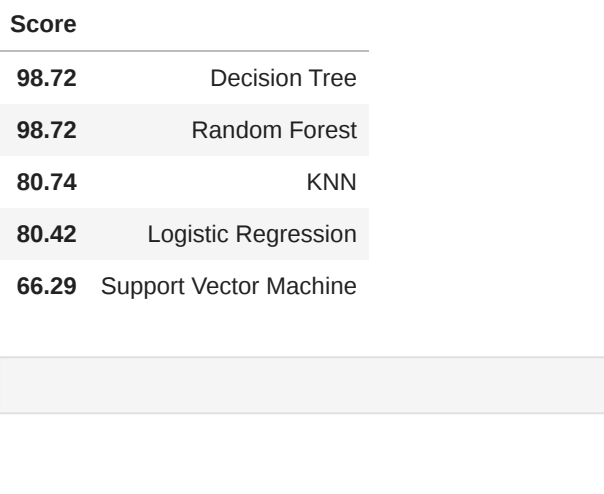
```
80.74
```

```
Out[45]: 80.74
```

```
In [46]: #confusion matrix
from sklearn.metrics import confusion_matrix
cm5=confusion_matrix(Y_test,ktestp)
```

```
Out[46]: array([[134, 32],
        [ 44, 58]], dtype=int64)
```

```
In [47]: from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm_display=ConfusionMatrixDisplay(confusion_matrix=cm5)
cm_display.plot()
plt.show()
```



```
In [48]: #checking the best model
result=pd.DataFrame({'Model':['Logistic Regression',"Decision Tree", "Random Forest", "Support Vector Machine", "KNN", "Logistic Regression"]})
result_df=result_df.sort_values(by="Score",ascending=False)
result_df=result_df.set_index('Score')
```

	Model
98.72	Decision Tree
98.72	Random Forest
80.74	KNN
80.42	Logistic Regression
66.29	Support Vector Machine

```
In [ ]:
```