January 19, 2022

[107]:

```
import pandas as pd
import io
import os
```

[108]:

```
#Name :

#Sex :

#Age :

#SibSp :    Sibling(   )  Spouse(  )

#Parch :    Parent( )  Child( )

#Ticket :

#Fare :

#Cabin :

#Embarked :    (CQS)
```

[109]:

```
os.chdir(r'C:\Users\gram\   ')
df = pd.read_csv('train.csv', encoding = 'utf-8')

train = df
print(train)
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …       …
886          887         0       2
887          888         1       1
888          889         0       3
```

```
889          890         1       1
890          891         0       3

                                                   Name     Sex   Age  SibSp  \
0                               Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                                Heikkinen, Miss. Laina  female  26.0      0
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                              Allen, Mr. William Henry    male  35.0      0
..                                                  …     …   …    …
886                               Montvila, Rev. Juozas    male  27.0      0
887                        Graham, Miss. Margaret Edith  female  19.0      0
888            Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                               Behr, Mr. Karl Howell    male  26.0      0
890                                 Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
[110]: os.chdir(r'C:\Users\gram\  ')
df0 = pd.read_csv('test.csv', encoding = 'utf-8')

test = df0
print(test)
```

```
     PassengerId  Pclass                                          Name  \
0            892       3                              Kelly, Mr. James
1            893       3              Wilkes, Mrs. James (Ellen Needs)
2            894       2                     Myles, Mr. Thomas Francis
3            895       3                              Wirz, Mr. Albert
4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
..             …     …                                           …
413         1305       3                            Spector, Mr. Woolf
414         1306       1                  Oliva y Ocana, Dona. Fermina
415         1307       3                  Saether, Mr. Simon Sivertsen
416         1308       3                           Ware, Mr. Frederick
```

```
417         1309       3                      Peter, Master. Michael J

        Sex   Age  SibSp  Parch              Ticket      Fare Cabin Embarked
0      male  34.5      0      0              330911    7.8292   NaN        Q
1    female  47.0      1      0              363272    7.0000   NaN        S
2      male  62.0      0      0              240276    9.6875   NaN        Q
3      male  27.0      0      0              315154    8.6625   NaN        S
4    female  22.0      1      1             3101298   12.2875   NaN        S
..      …    …     …     …                 …        …     …        …
413    male   NaN      0      0          A.5. 3236    8.0500   NaN        S
414  female  39.0      0      0            PC 17758  108.9000  C105        C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN        S
416    male   NaN      0      0              359309    8.0500   NaN        S
417    male   NaN      1      1                2668   22.3583   NaN        C

[418 rows x 11 columns]
```

[111]:
```
train.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        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[112]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

[113]:
```
train.head()
```

[113]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

[114]:
```
test.head()
```

[114]:
```
   PassengerId  Pclass                                          Name     Sex  \
0          892       3                              Kelly, Mr. James    male
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female
2          894       2                     Myles, Mr. Thomas Francis    male
3          895       3                              Wirz, Mr. Albert    male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female

    Age  SibSp  Parch   Ticket     Fare Cabin Embarked
0  34.5      0      0   330911   7.8292   NaN        Q
1  47.0      1      0   363272   7.0000   NaN        S
2  62.0      0      0   240276   9.6875   NaN        Q
3  27.0      0      0   315154   8.6625   NaN        S
```

```
4  22.0      1      1  3101298  12.2875   NaN        S
```

[115]:

```
print(train.columns)
#train   features
```

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

[116]:

```
print(pd.isnull(train).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
```

[117]:

```
print(pd.isnull(test).sum())
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

[118]:

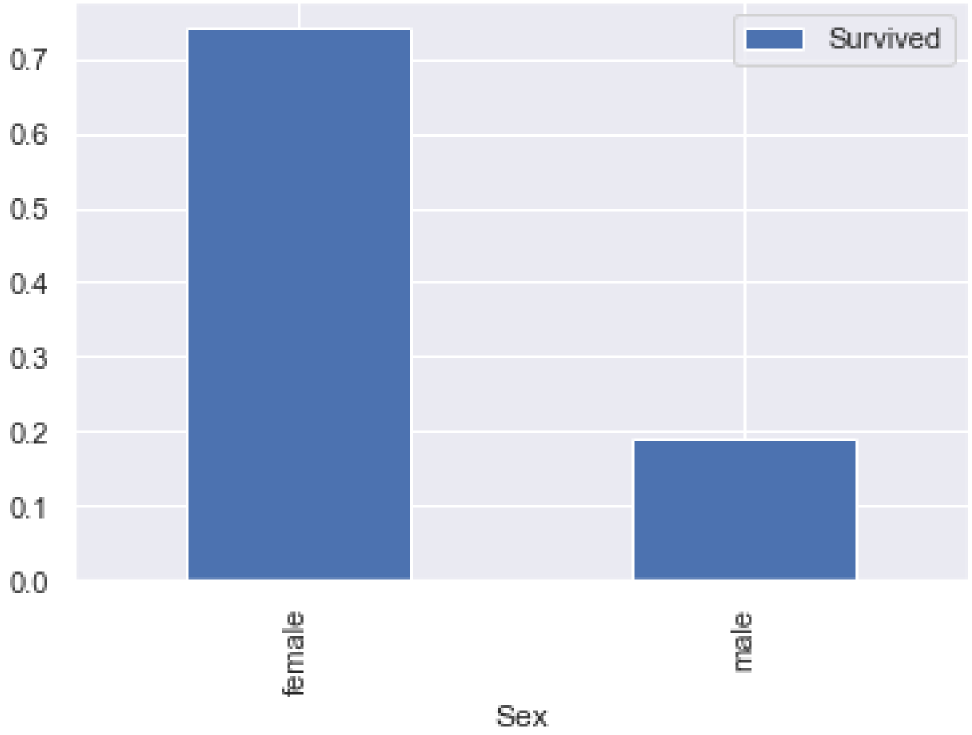
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

[119]:

```
def pivot(feature) :
  pclass_pivot = train.pivot_table(index=[feature],values = "Survived")
  pclass_pivot.plot.bar()
  plt.show()
#pivot
```

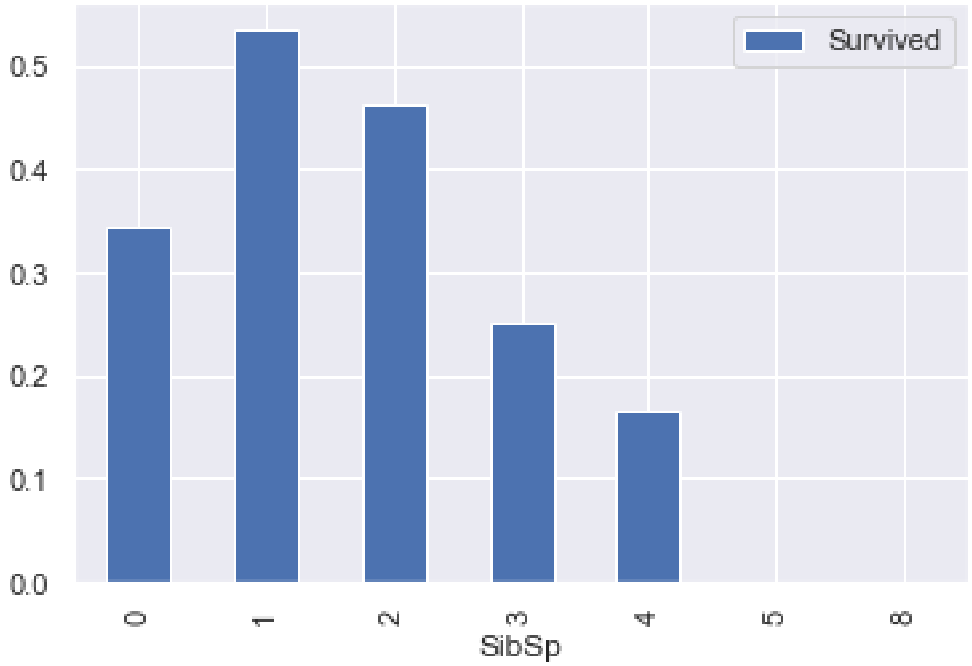
[120]:

```
pivot('Sex')
#
```



[121]:

```
pivot('SibSp')
```



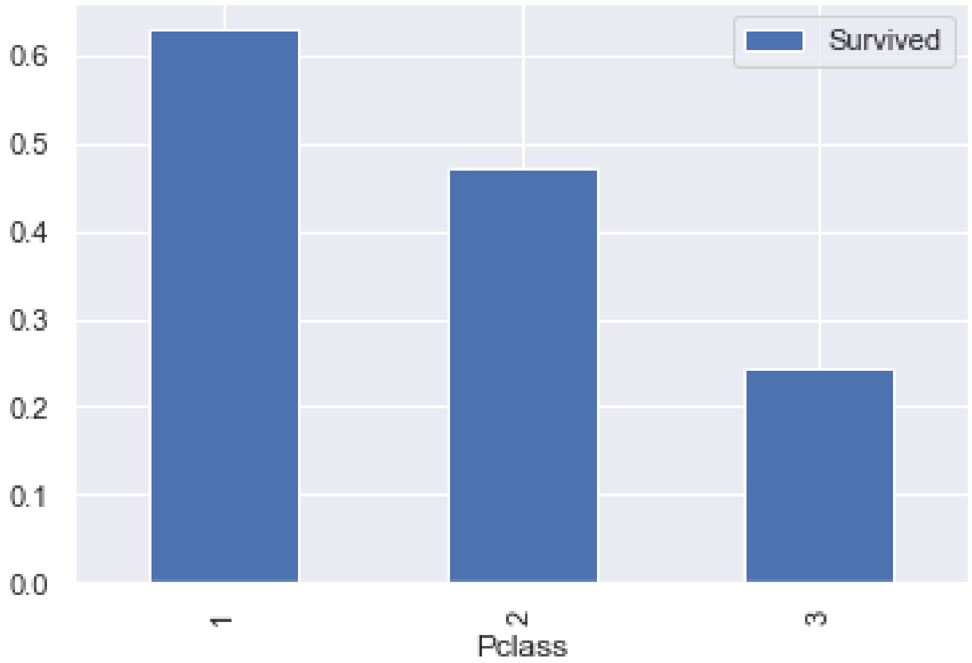
[123]:

```
train_test_data = [train, test]
for dataset in train_test_data :
  dataset["Title"] = dataset['Name'].str.extract('([A-za-z]+)\. ', expand=False)

#     ,
```

[122]:

```
pivot('Pclass')
#1
```



[124]:
```
test['Title'].value_counts #
```

[124]:
```
<bound method IndexOpsMixin.value_counts of 0           Mr
1          Mrs
2           Mr
3           Mr
4          Mrs
        …
413         Mr
414       Dona
415         Mr
416         Mr
417     Master
Name: Title, Length: 418, dtype: object>
```

[125]:
```
train['Title'].value_counts
```

[125]:
```
<bound method IndexOpsMixin.value_counts of 0         Mr
1        Mrs
2       Miss
3        Mrs
4         Mr
       …
886      Rev
887     Miss
```

```
888    Miss
889      Mr
890      Mr
Name: Title, Length: 891, dtype: object>
```

[126]:
```
title_mapping = {"Mr" : 0, "Miss": 1, "Mrs": 2,
                 "Master": 3, "Dr": 3, "Rev": 3, "Col": 3, "Ms": 2, "Mile": 3,␣
 ↪"Major": 3, "Lady": 3, "Capt": 3,
                 "Sir": 3, "Don": 3, "Mme": 3, "Jonkheer": 3, "Countess": 3}
for dataset in train_test_data:
  dataset['Title'] = dataset['Title'].map(title_mapping)

#Mr, Miss, Mrs,         ,
```

[127]:
```
train.head()
```

[127]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked  Title
0      0         A/5 21171   7.2500   NaN        S    0.0
1      0          PC 17599  71.2833   C85        C    2.0
2      0  STON/O2. 3101282   7.9250   NaN        S    1.0
3      0            113803  53.1000  C123        S    2.0
4      0            373450   8.0500   NaN        S    0.0
```

[128]:
```
test["Title"].fillna(df["Title"].mean(), inplace=True)#
```

[129]:
```
test["Title"].fillna(df["Title"].mean(), inplace=True)#
```

[130]:
```
print(pd.isnull(test).sum())
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
```

```
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
Title            0
dtype: int64
```

```
[131]: print(pd.isnull(train).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
Title            2
dtype: int64
```

```
[132]: df1 = train
       df1['Age'].fillna(df1['Age'].median(),inplace=True)
```

```
[133]: df2 = test
       df2['Age'].fillna(df2['Age'].median(),inplace=True)
```

```
[134]: train['Age'].isnull().sum()
```

```
[134]: 0
```

```
[135]: test['Age'].isnull().sum()
```

```
[135]: 0
```

```
[136]: train['Embarked'].value_counts()
```

```
[136]: S    644
       C    168
       Q     77
       Name: Embarked, dtype: int64
```

```
[137]: train["Embarked"] = train["Embarked"].fillna("S")
```

[138]:

```
train["Embarked"][train["Embarked"] == "S"] = 0
train["Embarked"][train["Embarked"] == "C"] = 1
train["Embarked"][train["Embarked"] == "Q"] = 2
```

```
C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[139]:

```
test["Embarked"][test["Embarked"] == "S"] = 0
test["Embarked"][test["Embarked"] == "C"] = 1
test["Embarked"][test["Embarked"] == "Q"] = 2
```

```
C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\ProgramData\Anaconda3\envs\tens_2\lib\site-packages\ipykernel_launcher.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[140]:

```python
train.head()
```

[140]:

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked  Title
0      0         A/5 21171   7.2500   NaN        0    0.0
1      0          PC 17599  71.2833   C85        1    2.0
2      0  STON/O2. 3101282   7.9250   NaN        0    1.0
3      0            113803  53.1000  C123        0    2.0
4      0            373450   8.0500   NaN        0    0.0
```

[141]:

```python
train['Embarked'].isnull().sum()
```

[141]: 0

[142]:

```python
test['Embarked'].isnull().sum()
```

[142]: 0

[143]:

```python
test["Fare"].fillna(df["Fare"].mean(), inplace=True)
```

[144]:

```python
#   drop, axis
train = train.drop(["Name"], axis = 1)
test = test.drop(["Name"], axis = 1)
```

[145]:

```python
train = train.drop(['PassengerId'], axis = 1)
test = test.drop(['PassengerId'], axis = 1)
```

[146]:

```python
train = train.drop(['Title'], axis = 1)
test = test.drop(['Title'], axis = 1)
```

[147]:
```
train = train.drop(['Cabin'], axis = 1)
test = test.drop(['Cabin'], axis = 1)
```

[148]:
```
train = train.drop(['Ticket'], axis = 1)
test = test.drop(['Ticket'], axis = 1)
```

[151]:
```
#
sex_mapping = {"male": 0, "female":1}
train['Sex'] = train['Sex'].map(sex_mapping)
test['Sex'] = test['Sex'].map(sex_mapping)
```

[150]:
```
train.head()
```

[150]:
```
   Survived  Pclass  Sex   Age  SibSp  Parch     Fare Embarked
0         0       3    0  22.0      1      0   7.2500        0
1         1       1    1  38.0      1      0  71.2833        1
2         1       3    1  26.0      0      0   7.9250        0
3         1       1    1  35.0      1      0  53.1000        0
4         0       3    0  35.0      0      0   8.0500        0
```

[76]:
```
test.head()
```

[76]:
```
   Pclass  Sex   Age  SibSp  Parch     Fare Embarked
0       3    0  34.5      0      0   7.8292        2
1       3    1  47.0      1      0   7.0000        0
2       2    0  62.0      0      0   9.6875        2
3       3    0  27.0      0      0   8.6625        0
4       3    1  22.0      1      1  12.2875        0
```

[77]:
```
train.isnull().sum()
```

[77]:
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

[78]:
```
test.isnull().sum()
```

[78]:
```
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
```

```
Fare        0
Embarked    0
dtype: int64
```

[79]:

```
from sklearn.model_selection import train_test_split
```

[80]:

```
pred = train.drop(['Survived'], axis = 1)
target = train["Survived"]
```

[81]:

```
x_train, x_test, y_train, y_test = train_test_split(pred, target, test_size = 0.
 ↪3)
```

[82]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

[83]:

```
logr = LogisticRegression()
logr.fit(x_train, y_train)
y_pred = logr.predict(x_test)
acc_logr = round(accuracy_score(y_pred, y_test) * 100, 2)
print(acc_logr)
```

```
75.37
```

[84]:

```
from sklearn.tree import DecisionTreeClassifier
```

[85]:

```
decisiontree = DecisionTreeClassifier()
decisiontree.fit(x_train, y_train)
y_pred = decisiontree.predict(x_test)
acc_decisiontree = round(accuracy_score(y_pred, y_test) * 100, 2)
print(acc_decisiontree)
```

```
75.37
```

[87]:

```
from sklearn.ensemble import RandomForestClassifier

randomforest = RandomForestClassifier()
randomforest.fit(x_train,y_train)
y_pred = randomforest.predict(x_test)
acc_randomforest = round(accuracy_score(y_pred, y_test) * 100, 2)
print(acc_randomforest)
```

```
80.22
```

[88]:

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)
acc_knn = round(accuracy_score(y_pred, y_test) * 100, 2)
```

```
print(acc_knn)
```

```
66.04
```

[89]:

```
models = pd.DataFrame({
    'Model' : ['KNN', 'Logistic Regression', 'Random Forest', 'Deicision Tree'],
    'Score' : [acc_knn, acc_logr, acc_randomforest, acc_decisiontree]
})
```

[90]:

```
models.sort_values(by='Score', ascending=False)
```

[90]:

```
                 Model  Score
2        Random Forest  80.22
1  Logistic Regression  75.37
3       Deicision Tree  75.37
0                  KNN  66.04
```