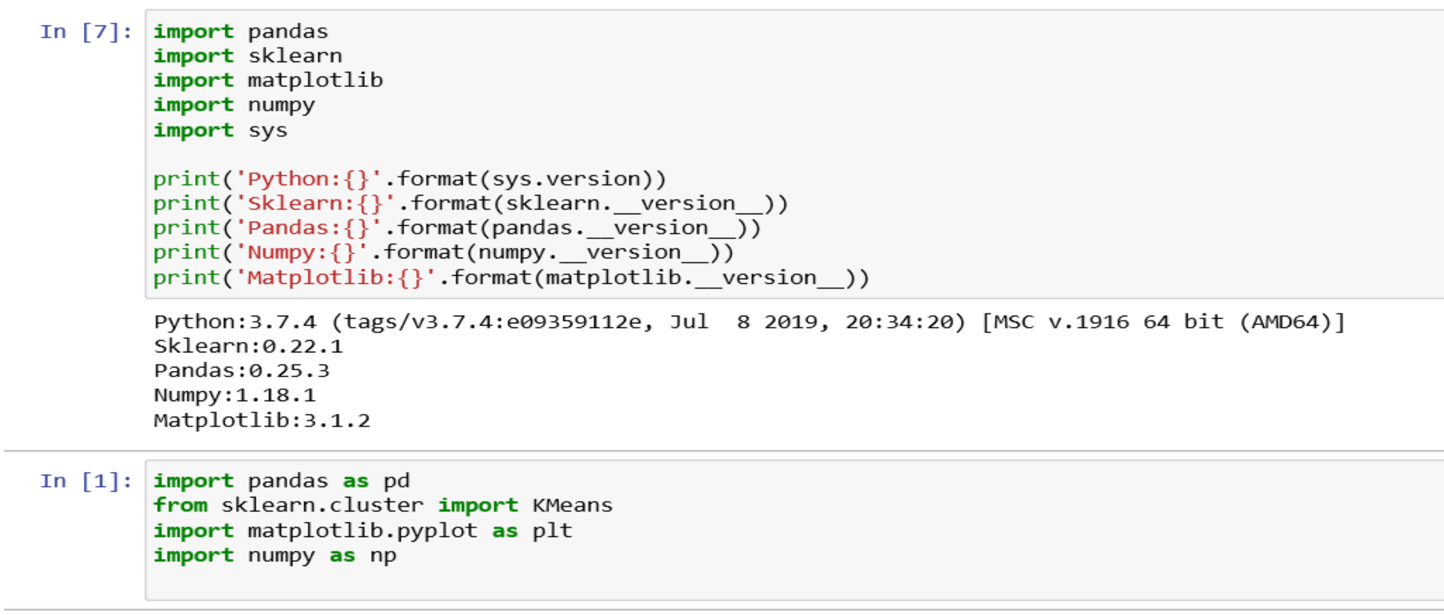
CRICKETER’S PERFORMANCE PREDICTOR:

INTRODUCTION: This predictor plays a vital role in selecting the right player for the right tournament. As we all know selecting the player for a tournament becomes a tedious task as we have to choose the right players for the right format of cricket. By using man power it is very hard task to analize each and every cricketer’s previous statistics and records.So,we have built a predictor which considers the player’s statistics as input and predicts how he plays on the particular opponent in the particular format of cricket.

In this project we build and deploy the Randomforest classifier algorithm

1.Importing Necessary Libraries:

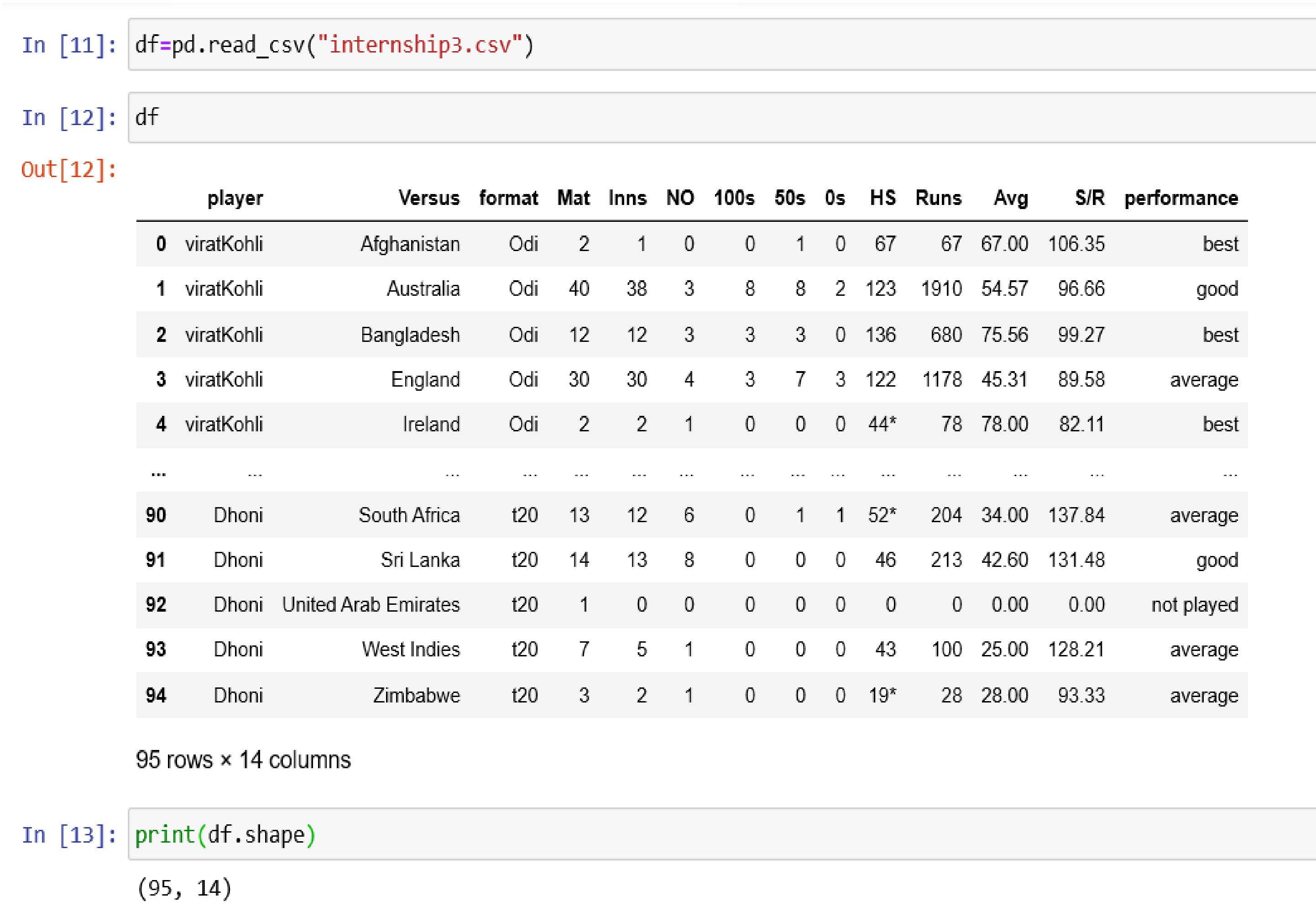


```
In [7]: import pandas
        import sklearn
        import matplotlib
        import numpy
        import sys

        print('Python:{}'.format(sys.version))
        print('Sklearn:{}'.format(sklearn.__version__))
        print('Pandas:{}'.format(pandas.__version__))
        print('Numpy:{}'.format(numpy.__version__))
        print('Matplotlib:{}'.format(matplotlib.__version__))

        Python:3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)]
        Sklearn:0.22.1
        Pandas:0.25.3
        Numpy:1.18.1
        Matplotlib:3.1.2
```

```
In [1]: import pandas as pd
        from sklearn.cluster import KMeans
        import matplotlib.pyplot as plt
        import numpy as np
```

2.Dataset:

In the following cells we will import our dataset from a .csv file as a pandas data frame.Further more,will begin exploring a dataset to gain an understanding of the type,quantity and distribution of data in our dataset.



```
In [11]: df=pd.read_csv("internship3.csv")
```

```
In [12]: df
```

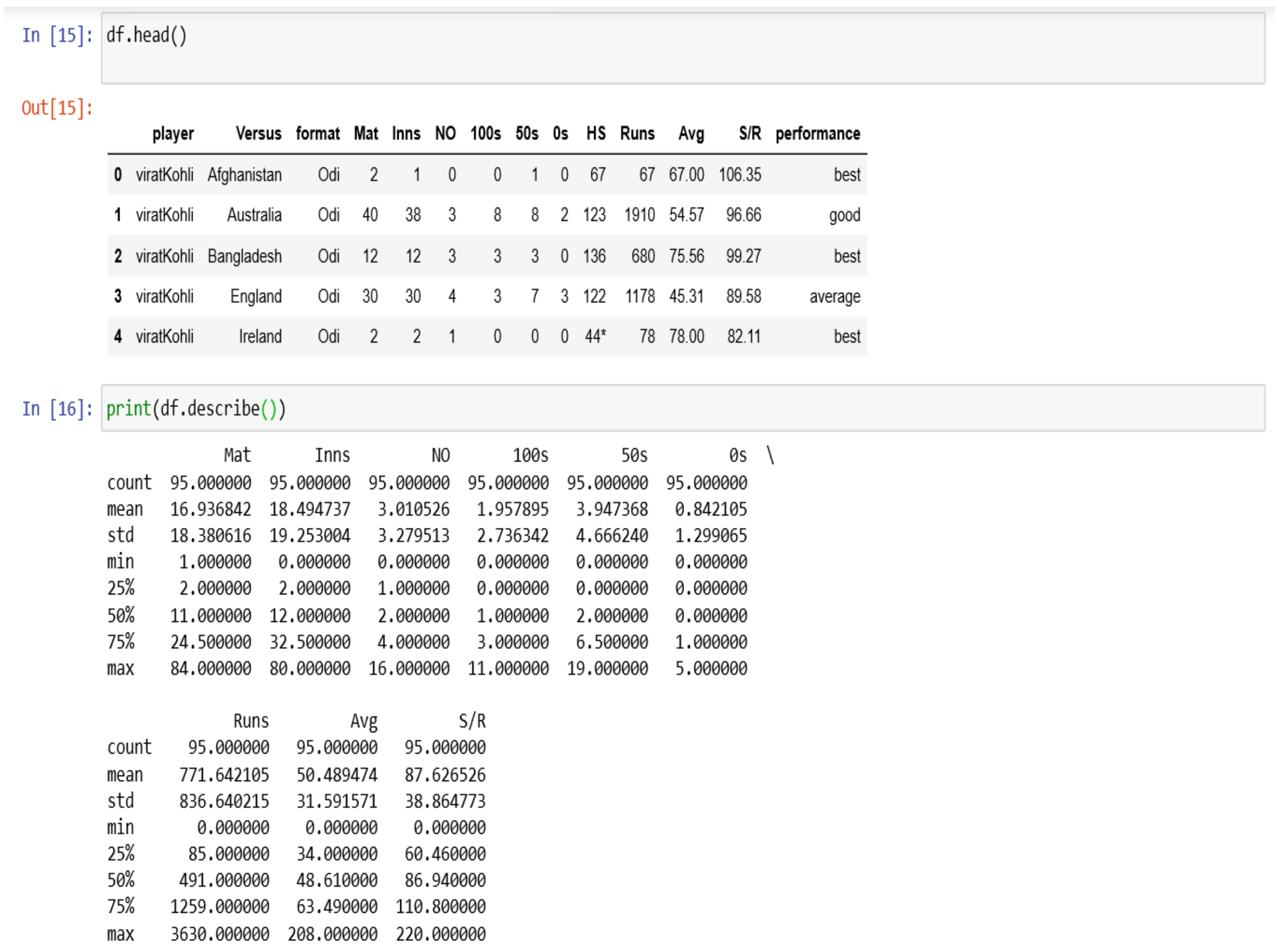
Out[12]:

	player	Versus	format	Mat	Inns	NO	100s	50s	0s	HS	Runs	Avg	S/R	performance
0	viratKohli	Afghanistan	Odi	2	1	0	0	1	0	67	67	67.00	106.35	best
1	viratKohli	Australia	Odi	40	38	3	8	8	2	123	1910	54.57	96.66	good
2	viratKohli	Bangladesh	Odi	12	12	3	3	3	0	136	680	75.56	99.27	best
3	viratKohli	England	Odi	30	30	4	3	7	3	122	1178	45.31	89.58	average
4	viratKohli	Ireland	Odi	2	2	1	0	0	0	44*	78	78.00	82.11	best
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
90	Dhoni	South Africa	t20	13	12	6	0	1	1	52*	204	34.00	137.84	average
91	Dhoni	Sri Lanka	t20	14	13	8	0	0	0	46	213	42.60	131.48	good
92	Dhoni	United Arab Emirates	t20	1	0	0	0	0	0	0	0	0.00	0.00	not played
93	Dhoni	West Indies	t20	7	5	1	0	0	0	43	100	25.00	128.21	average
94	Dhoni	Zimbabwe	t20	3	2	1	0	0	0	19*	28	28.00	93.33	average

95 rows × 14 columns

```
In [13]: print(df.shape)
```

```
(95, 14)
```



In [15]:
```
df.head()
```

Out[15]:

	player	Versus	format	Mat	Inns	NO	100s	50s	0s	HS	Runs	Avg	S/R	performance
0	viratKohli	Afghanistan	Odi	2	1	0	0	1	0	67	67	67.00	106.35	best
1	viratKohli	Australia	Odi	40	38	3	8	8	2	123	1910	54.57	96.66	good
2	viratKohli	Bangladesh	Odi	12	12	3	3	3	0	136	680	75.56	99.27	best
3	viratKohli	England	Odi	30	30	4	3	7	3	122	1178	45.31	89.58	average
4	viratKohli	Ireland	Odi	2	2	1	0	0	0	44*	78	78.00	82.11	best

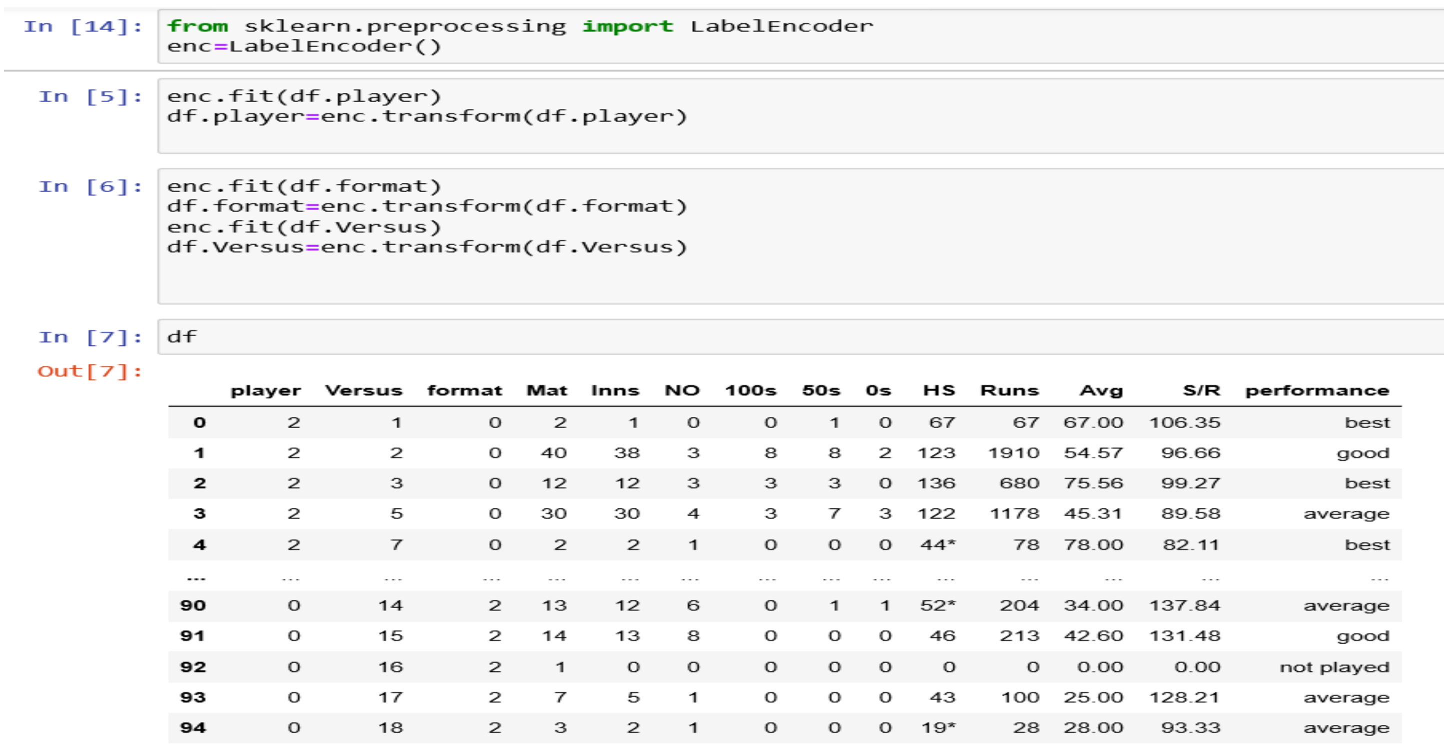
In [16]:
```
print(df.describe())
```

```
             Mat       Inns         NO       100s        50s         0s  \
count  95.000000  95.000000  95.000000  95.000000  95.000000  95.000000
mean   16.936842  18.494737   3.010526   1.957895   3.947368   0.842105
std    18.380616  19.253004   3.279513   2.736342   4.666240   1.299065
min     1.000000   0.000000   0.000000   0.000000   0.000000   0.000000
25%     2.000000   2.000000   1.000000   0.000000   0.000000   0.000000
50%    11.000000  12.000000   2.000000   1.000000   2.000000   0.000000
75%    24.500000  32.500000   4.000000   3.000000   6.500000   1.000000
max    84.000000  80.000000  16.000000  11.000000  19.000000   5.000000

              Runs         Avg         S/R
count    95.000000   95.000000   95.000000
mean    771.642105   50.489474   87.626526
std     836.640215   31.591571   38.864773
min       0.000000    0.000000    0.000000
25%      85.000000   34.000000   60.460000
50%     491.000000   48.610000   86.940000
75%    1259.000000   63.490000  110.800000
max    3630.000000  208.000000  220.000000
```

3.Preprocessing of data:

In this Project, we have preprocessed the player,versus,format columns.

Preprocessing is nothing but converting non numeric data to numeric data



In [14]:
```
from sklearn.preprocessing import LabelEncoder
enc=LabelEncoder()
```

In [5]:
```
enc.fit(df.player)
df.player=enc.transform(df.player)
```

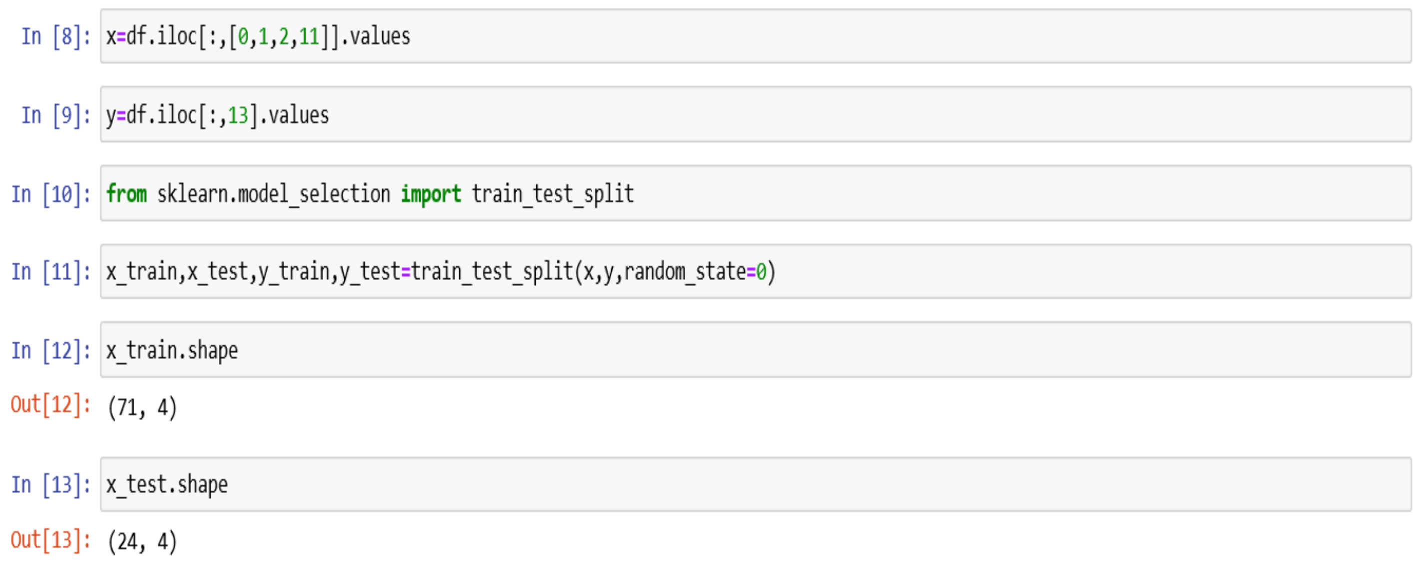
In [6]:
```
enc.fit(df.format)
df.format=enc.transform(df.format)
enc.fit(df.Versus)
df.Versus=enc.transform(df.Versus)
```

In [7]:
```
df
```

Out[7]:

	player	Versus	format	Mat	Inns	NO	100s	50s	0s	HS	Runs	Avg	S/R	performance
0	2	1	0	2	1	0	0	1	0	67	67	67.00	106.35	best
1	2	2	0	40	38	3	8	8	2	123	1910	54.57	96.66	good
2	2	3	0	12	12	3	3	3	0	136	680	75.56	99.27	best
3	2	5	0	30	30	4	3	7	3	122	1178	45.31	89.58	average
4	2	7	0	2	2	1	0	0	0	44*	78	78.00	82.11	best
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
90	0	14	2	13	12	6	0	1	1	52*	204	34.00	137.84	average
91	0	15	2	14	13	8	0	0	0	46	213	42.60	131.48	good
92	0	16	2	1	0	0	0	0	0	0	0	0.00	0.00	not played
93	0	17	2	7	5	1	0	0	0	43	100	25.00	128.21	average
94	0	18	2	3	2	1	0	0	0	19*	28	28.00	93.33	average

4.Training the Models:



In [8]:
```
x=df.iloc[:,[0,1,2,11]].values
```

In [9]:
```
y=df.iloc[:,13].values
```

In [10]:
```
from sklearn.model_selection import train_test_split
```

In [11]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)
```

In [12]:
```
x_train.shape
```

Out[12]:
```
(71, 4)
```

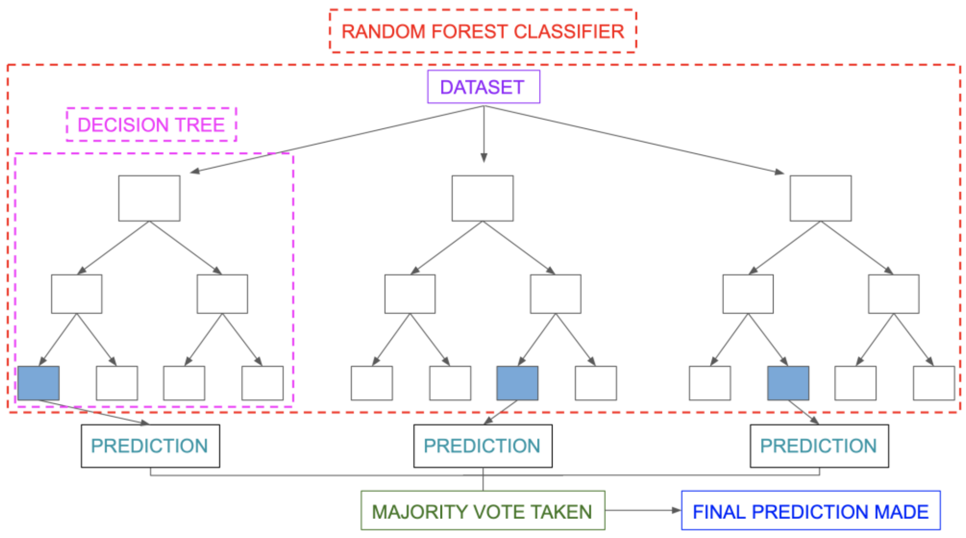
In [13]:
```
x_test.shape
```

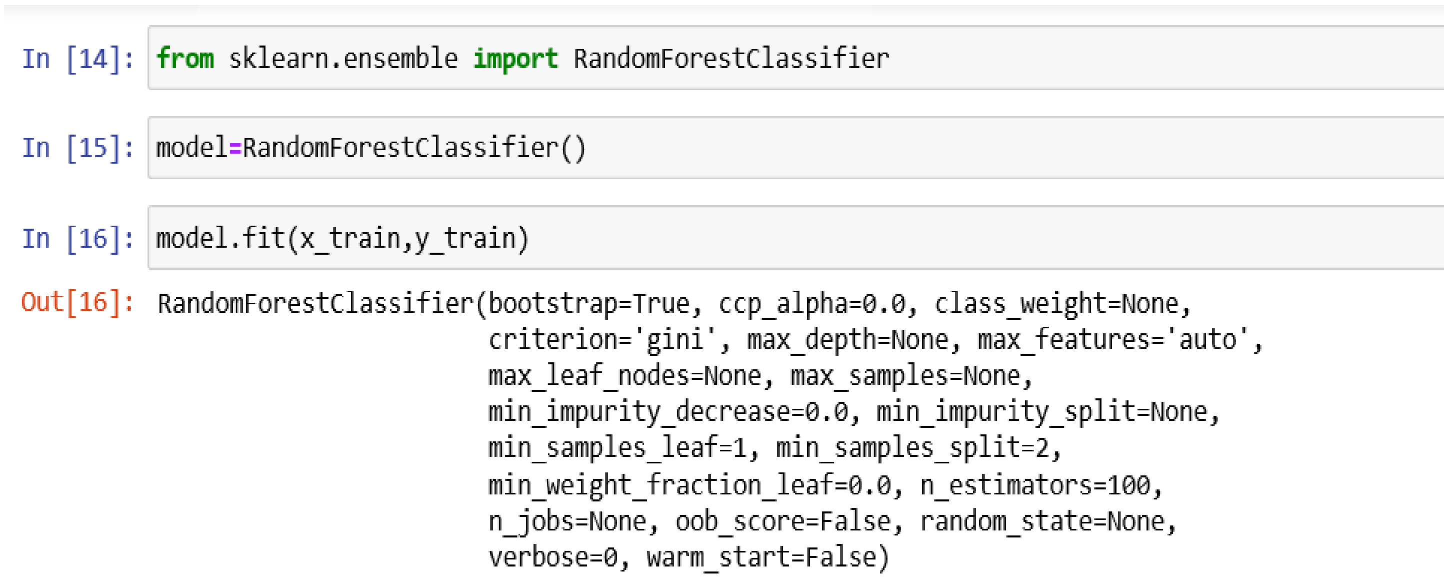
Out[13]:
```
(24, 4)
```

5.RandomForestClassifier Algorithm:

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method.

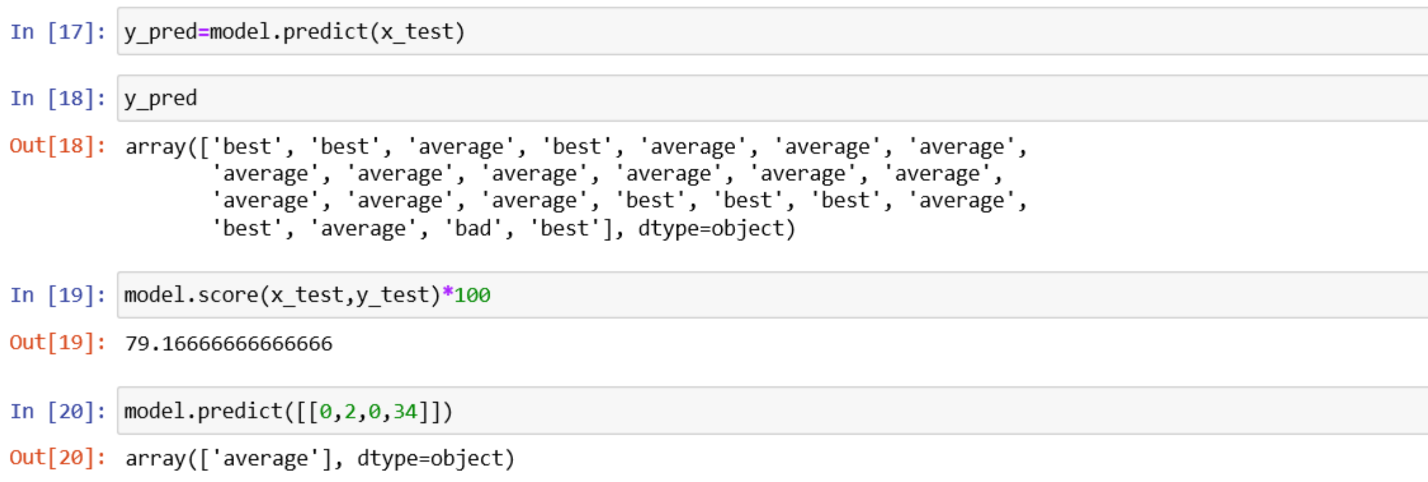




```
In [14]: from sklearn.ensemble import RandomForestClassifier

In [15]: model=RandomForestClassifier()

In [16]: model.fit(x_train,y_train)

Out[16]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                                criterion='gini', max_depth=None, max_features='auto',
                                max_leaf_nodes=None, max_samples=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=100,
                                n_jobs=None, oob_score=False, random_state=None,
                                verbose=0, warm_start=False)
```

6.Predicting the Results:

The results are predicted by considering the performance of the player.



```
In [17]: y_pred=model.predict(x_test)

In [18]: y_pred

Out[18]: array(['best', 'best', 'average', 'best', 'average', 'average', 'average',
               'average', 'average', 'average', 'average', 'average', 'average',
               'average', 'average', 'average', 'best', 'best', 'best', 'average',
               'best', 'average', 'bad', 'best'], dtype=object)

In [19]: model.score(x_test,y_test)*100

Out[19]: 79.16666666666666

In [20]: model.predict([[0,2,0,34]])

Out[20]: array(['average'], dtype=object)
```