

NAME : **K.DEVIKA**

CLASS : CSE-2A

ID : **S180632**

SUBJECT : Data Science with python

(A real time data science project on sports persons)

## **JOB PREDICTION MODEL FOR SPORTS**

### **PERSONS(BALL BADMINTON)**

#### **INTRODUCTION:**

Sports are a crucial part of a student's growth and development. They help in the development of mental health and physical fitness of the body. Through participation in the sports and games, a student gains various skills, experience and confidence that are helpful for developing their personality.

Sports are useful for people who build leadership skills, reduces chances of stroke and diabetes, improves muscular strength, reduces the stress, improves focus, get job opportunity, improves discipline, etc..

From many of these, one of the advantage is that this model is helpful to the people who want to do job on the basis of the game ball badminton.

## WHY THE MODEL IS?

Every sports person may have their own certifications, those people want to do the job for well-settled in their career. So, this model predicts whether they are getting the job based on the age, Qualification, International level winning, national level winning certificates and experience.

## HOW HAS THE DATA BEEN COLLECTED?

I played ball badminton (Sport) and I participate in many nationals and one in international. So I took the whole data from my co-players. From every individual player, I collected their details.

## ABOUT DATASET:-

This data set contains name, age, game, place, qualification of the person, number of national level participation, number of national level winnings, number of international level participation and number of international level winnings and experience.

**Sports dataset.xlsx**

	A	B	C	D	E	F	G	H	I	J	K
1	Name	Age	Game	District	Qualifica	National level partici	National level winnir	International participa	International winning	JOB	Experience
2	K.Devika	19	Ball Badmir	Visakhapa	1	15	15	1	1	0	5
3	K.Rupendra	22	Ball Badmir	Chittoor	2	7	7	0	0	0	4
4	R.Kavya Sri	19	Ball Badmir	Visakhapa	1	4	4	0	0	0	4
5	Ch.DhanaLakshmi	19	Ball Badmir	Visakhapa	1	4	4	1	1	0	5
6	G.Veeraju	20	Ball Badmir	Visakhapa	1	2	2	0	0	0	3
7	G.Hemanth	22	Ball Badmir	Visakhapa	2	6	6	1	1	1	5
8	R.Vamsi	22	Ball Badmir	Visakhapa	2	3	3	0	0	0	3
9	K.Samba Siva	19	Ball Badmir	Visakhapa	1	1	1	0	0	0	2
10	G.Sivaji	20	Ball Badmir	Visakhapa	1	1	1	0	0	0	2
11	R.Kalyani	17	Ball Badmir	Visakhapa	0	1	1	0	0	0	2
12	B.V.D Kavya	17	Ball Badmir	Vijayanaga	0	1	1	0	0	0	2
13	P.V.V.S Jaya Lakshmi	23	Ball Badmir	Vijayanaga	2	13	10	0	0	1	5
14	K.Thanooja	19	Ball Badmir	Nellore	1	3	3	0	0	0	3
15	M.Swetha	19	Ball Badmir	Nellore	1	6	5	0	0	0	3
16	K.Lavanya	19	Ball Badmir	Visakhapa	1	1	1	0	0	0	3
17	M.Sravanthi	17	Ball Badmir	Visakhapa	1	5	5	1	1	0	3
18	P.Bhavani	19	Ball Badmir	Krishna	1	3	3	0	0	0	3
19	Ch.Madhumitha	26	Ball Badmir	Nellore	0	1	1	0	0	0	2
20	E.Harish	26	Ball Badmir	Tamilnadu	3	5	5	1	1	1	3
21	K.Shyam kumar	25	Ball Badmir	Visakhapa	3	5	5	1	1	1	5
22	Ch.Mutyalanaidu	25	Ball Badmir	Visakhapa	0	3	3	0	0	0	6
23	V.Suresh Kumar	20	Ball Badmir	Kurnool	1	1	1	0	0	0	3
24	T.Prema Latha	19	Ball Badmir	Guntur	1	2	2	0	0	0	2
25	K.Laxman Rao	21	Ball Badmir	Vijayanaga	1	1	1	0	0	0	2
26	P.Srinu	23	Ball Badmir	Chittoor	2	2	2	0	0	0	2
27	B.Yedukondalu	20	Ball Badmir	Prakasam	1	2	2	0	0	0	2

## GETTING STARTED WITH CODE:

### 1. Import python libraries:

In this model I am using Pandas, Numpy and Matplotlib libraries. Pandas is used for working with datasets. Numpy is used for working with array and matplotlib is used for plotting or that serves a visualization utility. By using these libraries I build my model.

```
import matplotlib.pyplot as plt

import pandas as pd
```

```
import numpy as np
```

## 2. Creating DATAFRAME

```
a=pd.read_excel(r"C:\Users\Devika\Downloads\Sports  
dataset.xlsx",sheet_name="Sheet1")
```

```
#printing data
```

```
df.head(5)
```

### OUTPUT:

	Name	Age	Game	District	Qualification	National level participation	National level winnings	International participation	International winning	JOB	Experience
0	K.Devika	19	Ball Badminton	Visakhapatnam	1	16	15	1	1	0	5
1	K.Rupendra	22	Ball Badminton	Chittoor	2	7	7	0	0	0	4
2	R.Kavya Sri	19	Ball Badminton	Visakhapatnam	1	4	3	0	0	0	4
3	Ch.DhanaLakshmi	19	Ball Badminton	Visakhapatnam	1	4	4	1	1	0	5
4	G.Veeraraju	20	Ball Badminton	Visakhapatnam	1	2	2	0	0	0	3

## 3. Getting columns

```
df.columns
```

### OUTPUT:

```
Index(['Name', 'Age', 'Game', 'District', 'Qualification',  
      'National level participation', 'National level winnings',  
      'International participation', 'International winning', 'JOB',  
      'Experience'], dtype='object')
```

## 4. Cleaning and preparing the data

In the process of cleaning data, I could not find any noise for the data. Because I took the real data from not from any other sources. But I have done some operations like checking for any null values, information about the data set, description of the data set and count of the data set.

```
df.isnull()
```

**OUTPUT:**

	Name	Age	Game	District	Qualification	National level participation	National level winnings	International participation	International winning	JOB	Experience
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False

```
df.info()
```

**OUTPUT:**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Name                                  131 non-null    object
1   Age                                   131 non-null    int64
2   Game                                  131 non-null    object
3   District                              131 non-null    object
4   Qualification                         131 non-null    int64
5   National level participation          131 non-null    int64
6   National level winnings              131 non-null    int64
7   International participation           131 non-null    int64
8   International winning                 131 non-null    int64
9   JOB                                   131 non-null    int64
10  Experience                            131 non-null    int64
dtypes: int64(8), object(3)
memory usage: 11.4+ KB
```

```
df.describe()
```

### OUTPUT:

	Age	Qualification	National level participation	National level winnings	International participation	International winning	JOB	Experience
count	99.000000	99.000000	99.000000	99.000000	99.000000	99.000000	99.000000	99.000000
mean	22.505051	1.545455	4.121212	3.848485	0.444444	0.404040	0.282828	3.333333
std	4.815735	0.786009	2.760038	2.580791	0.673435	0.653393	0.452666	2.258770
min	15.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000
25%	19.000000	1.000000	2.000000	1.500000	0.000000	0.000000	0.000000	2.000000
50%	22.000000	1.000000	4.000000	3.000000	0.000000	0.000000	0.000000	3.000000
75%	25.000000	2.000000	6.000000	5.000000	1.000000	1.000000	1.000000	4.000000
max	45.000000	3.000000	16.000000	15.000000	3.000000	3.000000	1.000000	15.000000

```
df.count()
```

### OUTPUT:

```
Name      131
Age        131
Game       131
District   131
Qualification  131
National level participation  131
National level winnings      131
International participation    131
International winning         131
JOB      131
Experience  131
dtype: int64
```

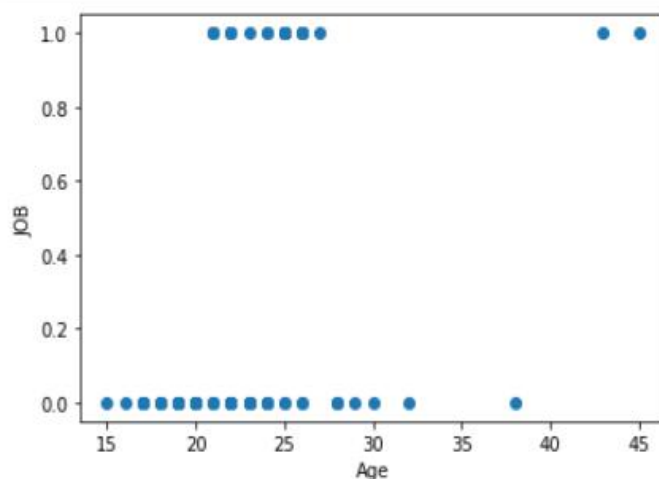
## 5. Visualization

For visualization, I am importing matplotlib library at the beginning of the code.

### 1) Relation between the age and job

```
plt.scatter(df['Age'],df['JOB'])  
  
plt.xlabel('Age')  
  
plt.ylabel('JOB')  
  
plt.show()
```

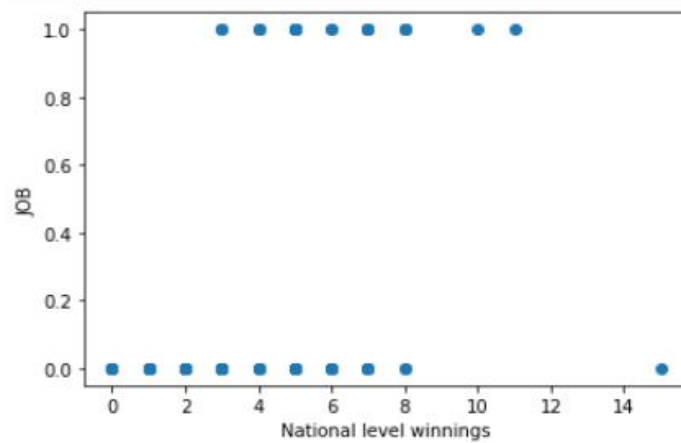
**OUTPUT:**



### 2) Relation between national level winnings and the job

```
plt.scatter(df['National level winnings'],df['JOB'])  
  
plt.xlabel('National level winnings')  
  
plt.ylabel('JOB')  
  
plt.show()
```

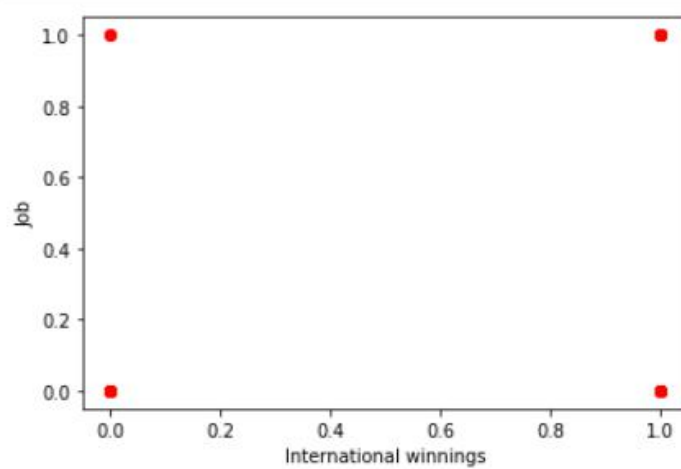
## OUTPUT:



## 3) Relation between international winnings and the job

```
plt.scatter(df['International winning'],df['JOB'])  
plt.xlabel('International winnings')  
plt.ylabel('Job')  
plt.show()
```

## OUTPUT:





#### 4) Relation between qualification and job

```
#RELATION BETWEEN QUALIFICATION AND JOB

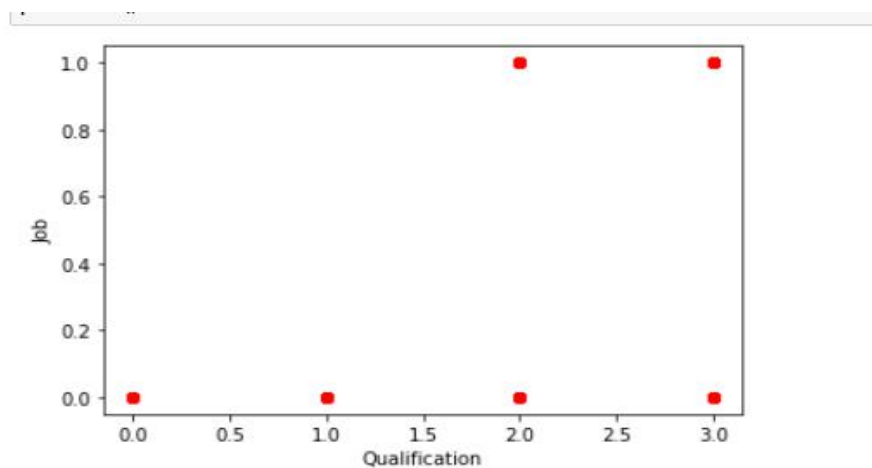
plt.scatter(df['Qualification'],df['JOB'],c="red")

plt.xlabel('Qualification')

plt.ylabel('Job')

plt.show()
```

#### OUTPUT:-



### 6.IMPORTING MODULES

```
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import train_test_split
```

### 7.Preparing dependent and independent variables

```
x=df.drop(['Name','Game','District','National level participation','International participation','JOB'],axis=1)

y=df['JOB']

print(x)
```

## OUTPUT:-

	Age	Qualification	National level winnings	International winning	\
0	19		1	15	1
1	22		2	7	0
2	19		1	4	0
3	19		1	4	1
4	20		1	2	0
..	...				
126	25		0	1	1
127	26		0	1	1
128	26		0	1	1
129	26		0	1	1
130	18		1	10	0

Experience

0	5
1	4
2	4
3	5
4	3
..	...
126	3
127	3
128	3
129	3
130	5

[131 rows x 5 columns]

```
print(y)
```

## OUTPUT:-

0	0
1	0
2	0
3	0
4	0
..	
126	0
127	0
128	0
129	0
130	0

Name: JOB, Length: 131, dtype: int64

## 8. Dividing the dataset

Here, the data will be divided into two parts, some data for training and remaining data for testing.

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1)
```

## 9. Reshaping the dataset

```
y_train=np.array(y_train).reshape(-1,1)
y_test=np.array(y_test).reshape(-1,1)
```

## 10. Printing the x training data

```
x_train.shape
```

**OUTPUT:**

(104, 5)

## 11. Classification model

In this model prediction, I am using an algorithm which is DecisionTreeClassifier.

```
model=DecisionTreeClassifier()
```

## 12. Fitting the model

```
model.fit(x_train,y_train)
```

**OUTPUT:**

```
DecisionTreeClassifier()
```

### 13. Predicting the testing data values

```
a=model.predict(x_test)

for i in a:

    if(i==0):

        print(" YOU WILL NOT GET THE JOB")

    else:

        print("YOU WILL GET THE JOB")
```

**OUTPUT:**

```
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL NOT GET JOB
YOU WILL GET JOB
YOU WILL GET JOB

array([0, 0, 0, 0, 0, 0, 0, 0, 1, 1], dtype=int64)
```

### 14) Actual testing values

```
print(y_test)
```

**OUTPUT:**

```
[[1]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [1]
 [1]]
```

## 15. Predicting some values:-

**Explanation:-**(Some conditions are given in the program for user convenience)

Here I consider the sequence of the input array is age, qualification, national level winning certificates, international winning certificates and their experience.

The first parameter is age, every applicant who are applied for job those person's age should be in between 18 and 27. The person who are under 18 and 27 is not eligible to apply.

The second parameter is qualification, I consider 0,1,2,3 for 10<sup>th</sup>, intermediate, degree or B Tech and higher education respectively.

If the person age is below and is equal to 18 then the person eligibilities are 10<sup>th</sup> and intermediate(0 and 1). If the person age is below and is equal to 22 then the person eligibility qualifications are 0,1,2. If the person age is below and is equal to 27 then the person eligibilities are 0,1,2,3.

The third parameter is national winning certificates and fourth one is international winning certificates.

The person should have equal number of international and national certificates or else international certificates is less than national certificates.

The fifth parameter is experience which is used to declare the person having fake certificates or duplicate certificates by using the count of national certificates.

### CODE:-

```
t=np.array([[18,1,1,1,2]])  
if(t[0,0]<18):
```

```

    print("You are not eligible because you are under 18 ")
elif(t[0,0]>27 ):
    print("You are not eligible because your age is above 27")
elif(t[0,0]<=21 and t[0,1]>1):
    print("According to your age, your eligibilities are 10th and
    equivalent.so please enter correct qualification")
elif(t[0,0]==22 and t[0,1]>2):
    print("According to your age, your eligibilities are 10th,equivalent
    and degree.so please enter correct qualification")
elif(t[0,1]>3):
    print("Out of the range,age between 23 to 27 the eligibilities
    0,1,2,3")
elif(t[0,3]>t[0,2]):
    print("Not possible to get more number of international certificates
    than national level certificates")
elif(t[0,4]*3<t[0,2]):
    print("It is not possible to get more no of nationals as per your
    experience.Because,a person can play maximum three no of nationals
    per year.")
else:
    a=model.predict(t)
    print("The Result is : {}".format(a))
    if(a==0):
        print("You will not get the job")
    else:

```

```
print("You will get the job")
```

**OUTPUT:**

```
The Result is : [0]  
You will not get the job
```

```
t=np.array([[22,2,10,2,4]])
```

**OUTPUT:-**

```
The Result is : [1]  
You will get the job
```

**CONCLUSION:**

This model would be helpful to the ball badminton players to know they are getting the job or not so that they can prepare accordingly.