Project introduction / problem statement :-

- > implement a predictive model to determine whether an employee is going to quit or not from the orginazation

Data source :- Kaagle

Describe the dataset :-

Dataset Structure: 1470 observations (rows), 35 features (variables)

Target_Varible :- Attrition

Missing Data: there is no missing data! this will make it easier to work with the dataset.

In Attrition Column , YES - means person is about to leave or person has already left ,
NO- means person has not left or still working

Data Type: We only have two datatypes in this dataset: factors and integers

Label" Attrition is the label in our dataset and we would like to find out why employees are leaving the organization!

Imbalanced dataset: 1237 (84% of cases) employees did not leave the organization while 237 (16% of cases) did leave the organization making our dataset to be considered imbalanced since more people stay in the organization than they actually leave.

Describe the treatment on the data :-

1) import pandas and numpy to data manupulation and analysis

2) read the csv data set using pandas

3) find the heads and tails of the data

4) then go to the eda part ; data info , data describe , find correlation b/w the independent variable

5) check the relationship with independent variable and target variable to find the which variable is more significant with the hellp of graphical representation .
.
6) drop the variable , those variable is not influences to my target variable .

7) then go to label encoder , and to convert the non numeric value to numeric value .

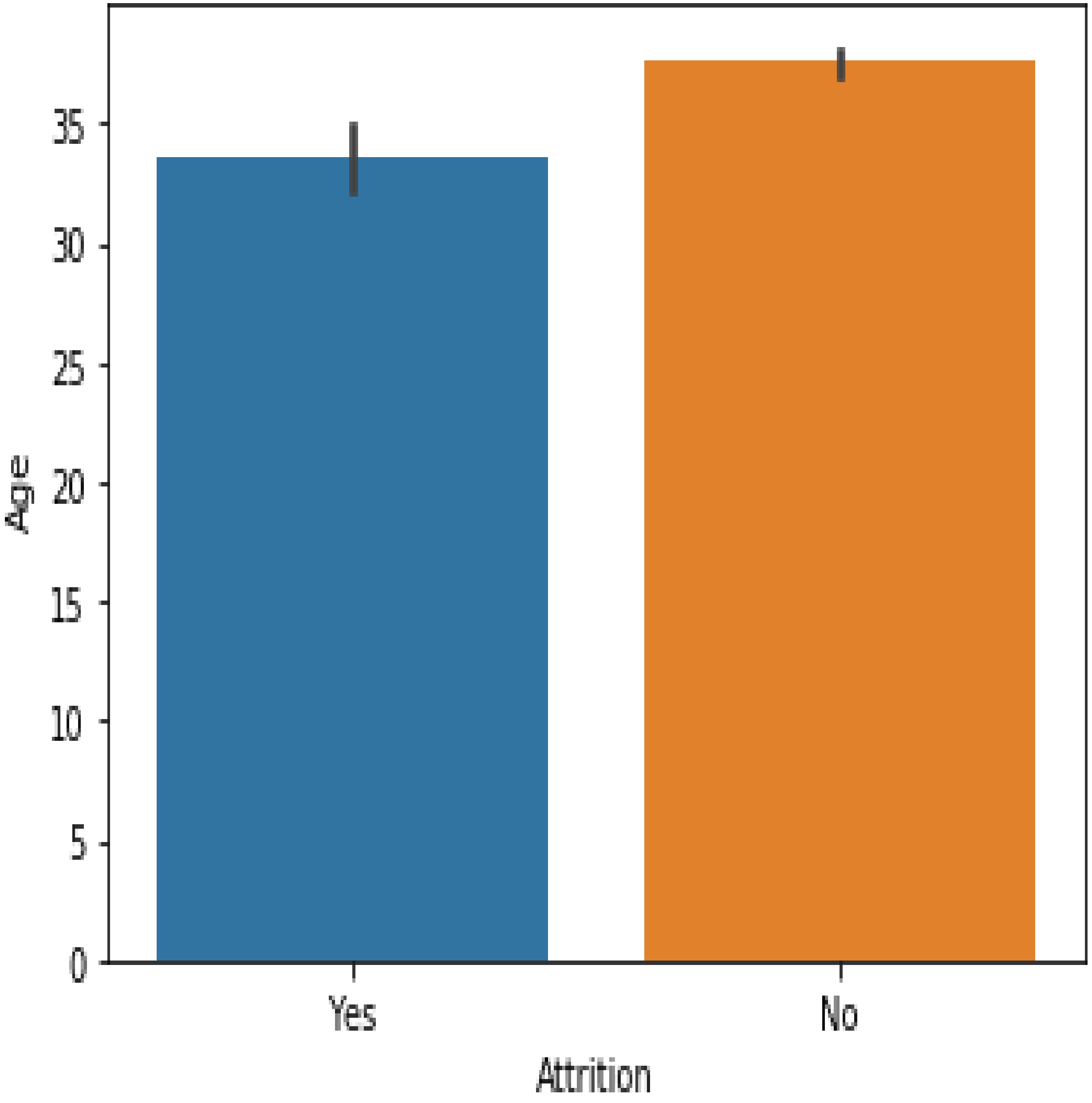
8) we have check vif factor , check the multicolinerity is exist in the data or not .

9) find the ilocation of the data set

10) go to the sampling , to spilt the in two part train or test

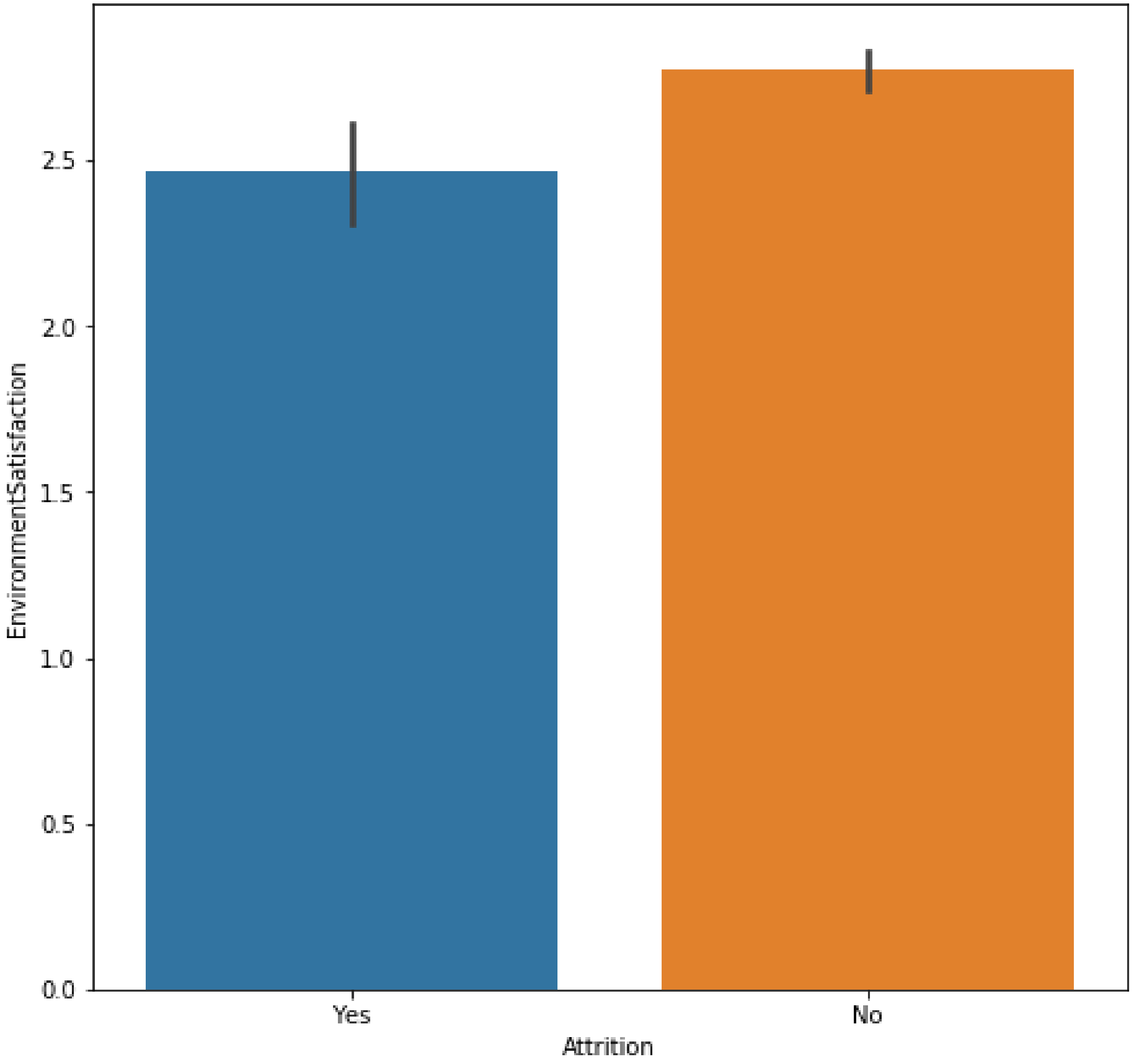
```
sns.barplot(y ="Age" , x= "Attrition", data=atr)
```

SUMMARY :- age of employee more than 40 is staying at the organization but less than age of 40 employee will quit the organization



```
plt.figure(figsize=(8,8)) sns.barplot(y
="EnvironmentSatisfaction" , x= "Attrition", data=atr)
```

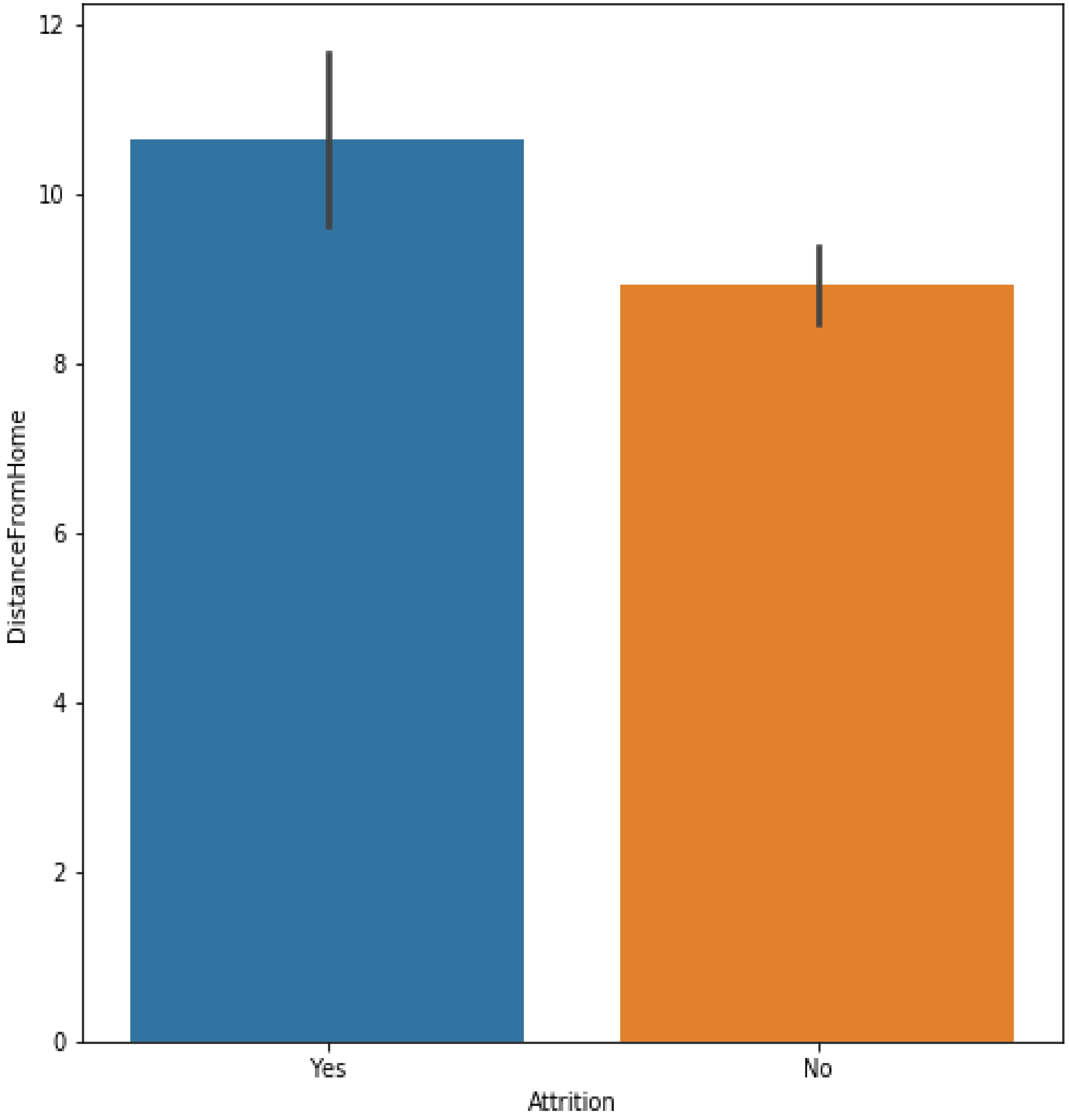
SUMMARY :- EnviromentSatisfaction is directly impacted to attrition data .
. EmployeeSatisfaction Below the 2.5 , the employee are qit the organisation



```
plt.figure(figsize=(8,8)) sns.barplot(y
="DistanceFromHome" , x= "Attrition", data=atr)
```

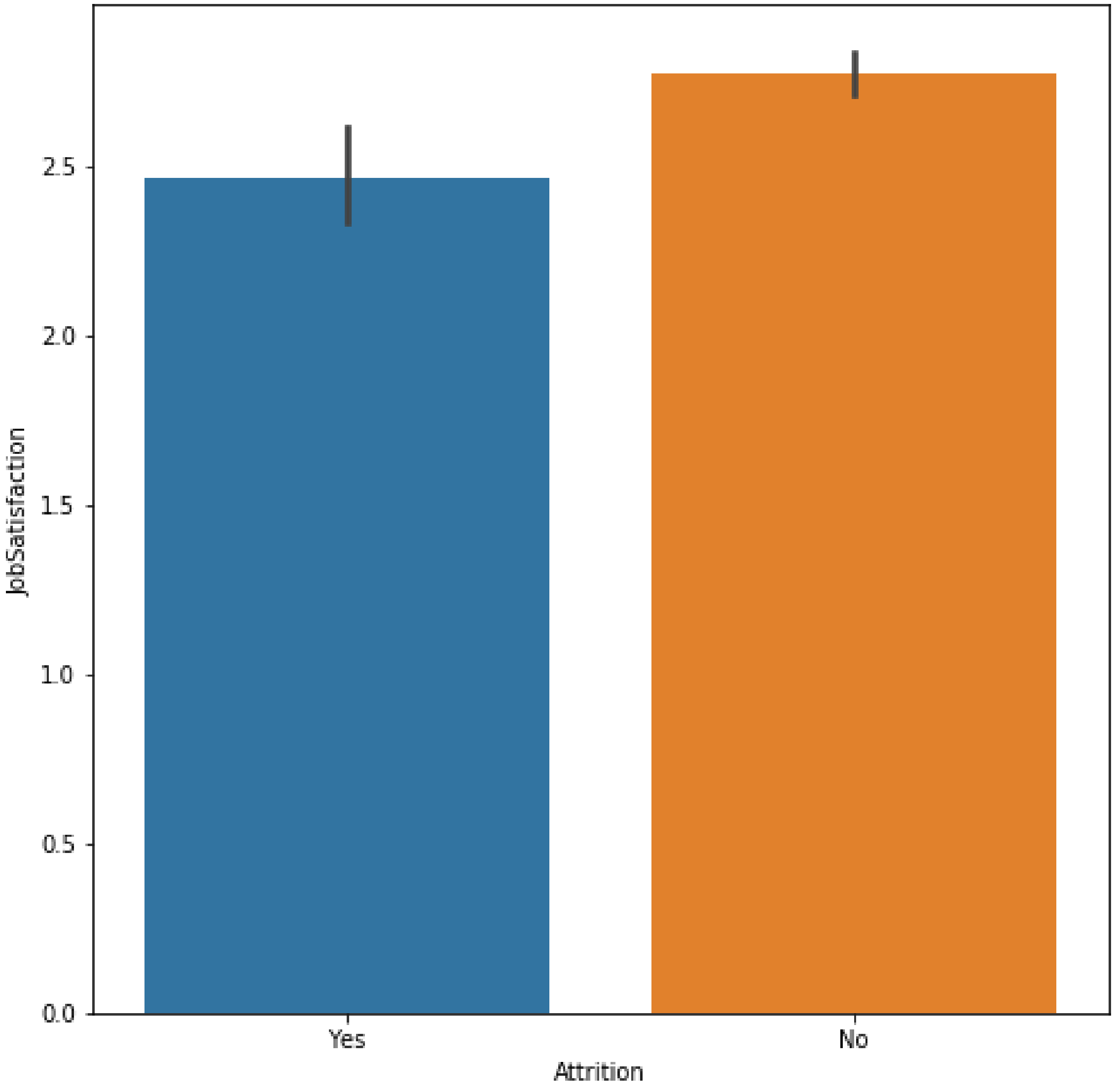
SUMMARY :- distance from home is main parameter to impact the my target variable .

. Above the 9 km distance , the employee is quit in this organization .



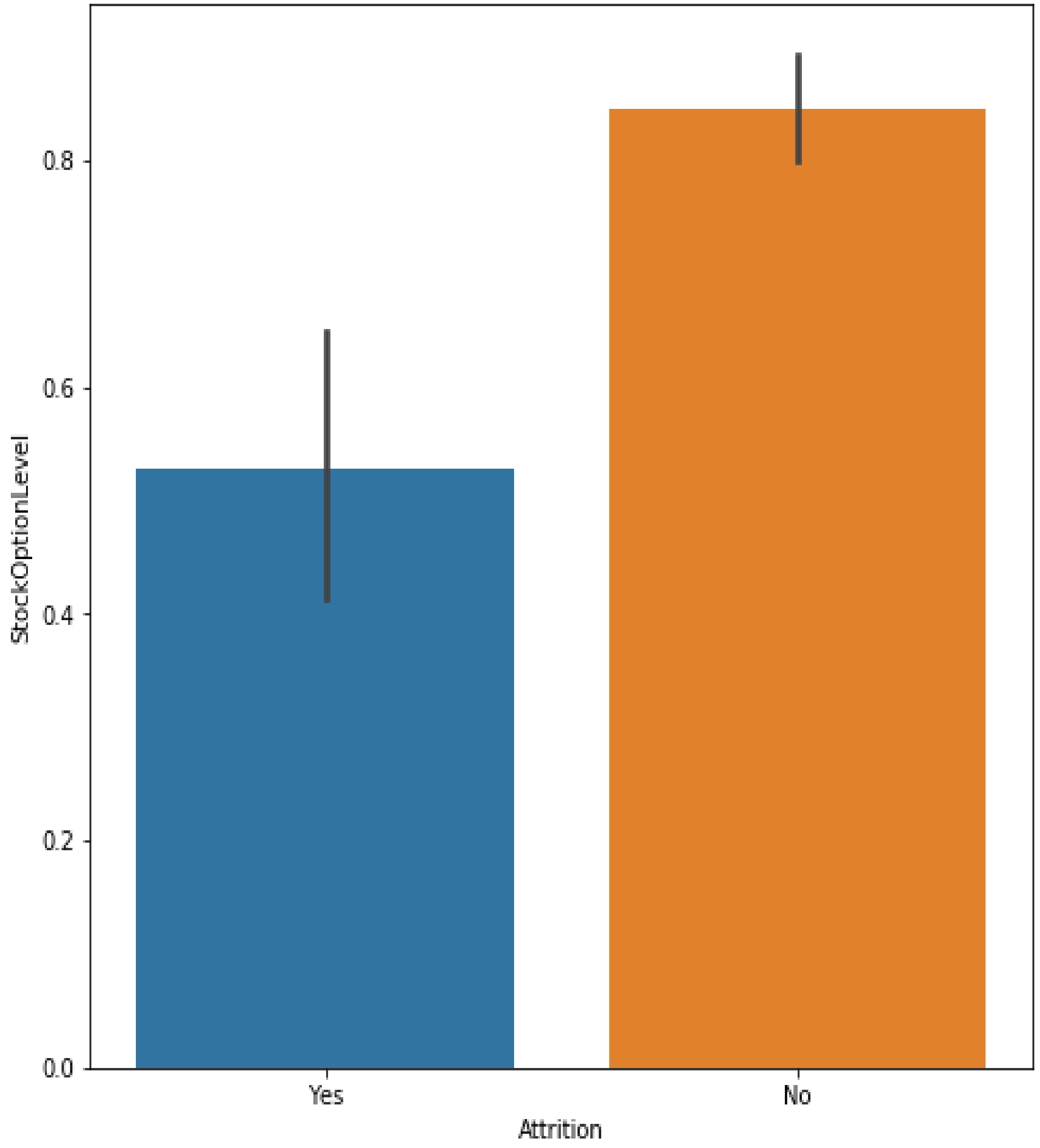
```
plt.figure(figsize=(8,8)) sns.barplot(y ="JobSatisfaction" ,
x= "Attrition", data=atr)
```

SUMMARY :- JobSatisfication is directly Impacted to the Attrition variable .



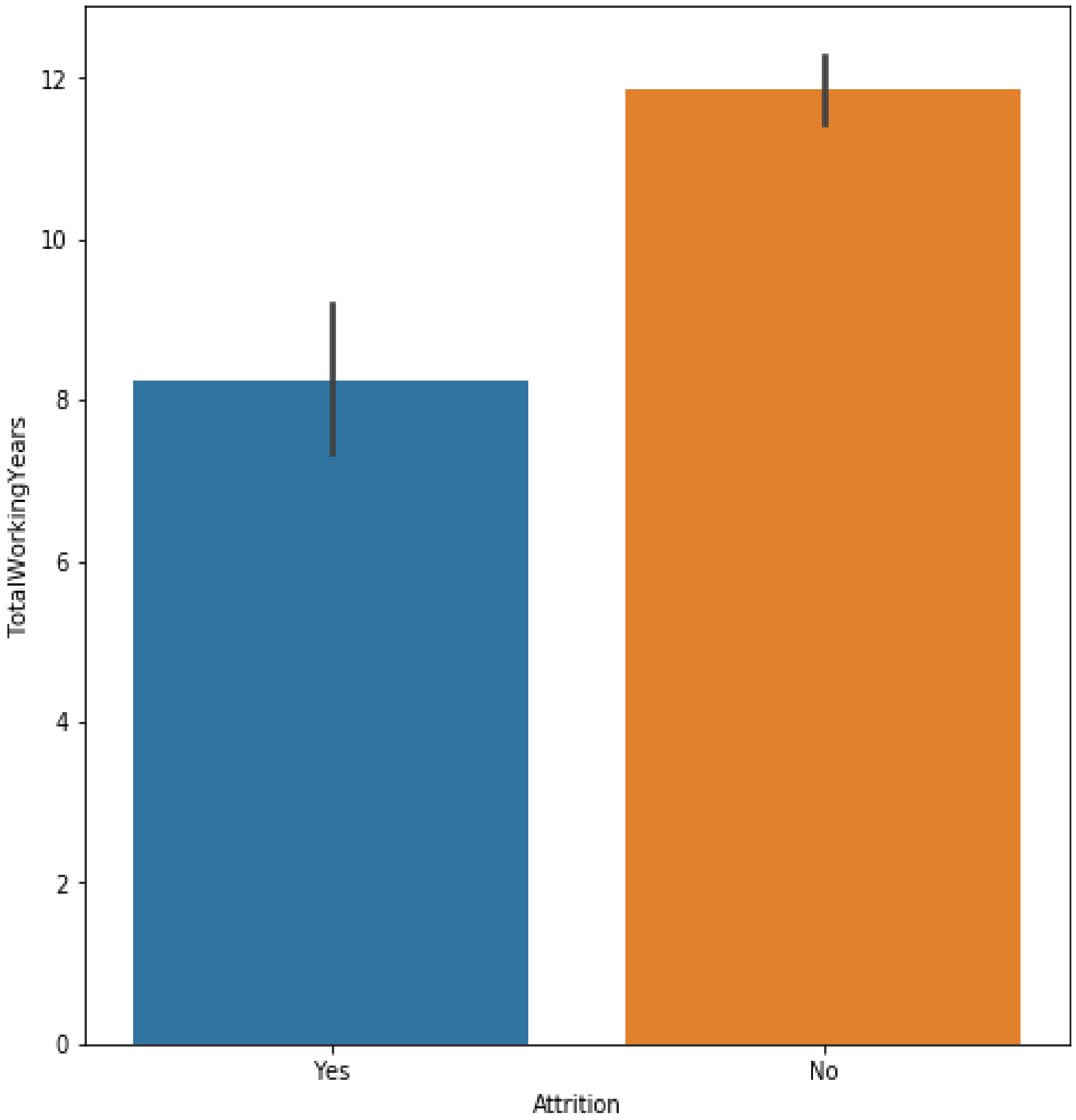
```
plt.figure(figsize=(8,8))
 sns.barplot(y ="StockOptionLevel" , x ="Attrition", data=atr)
```

SUMMARY :- StockOptionLevel is highly impacted my target variable



```
plt.figure(figsize=(8,8)) sns.barplot(y ="TotalWorkingYears" ,
x= "Attrition", data=atr)
```

SUMMARY :- TotalWorking year is my most significant and highly inpacted my target variable .
Above the 8 year of working experience that employee is still working in this organization .



Model Building :-

. logistic regression :- *logistic regression is a method of classification when a target variable is categorical.*

. Its working principle is regression .

. Conf_Mat :- array([[145, 16],
[98, 35]], dtype=int64)

. tp = 145 fp =16 fn = 98 tn = 35

. ACCURACY in the Conf_Mat :- 61.224489795918366

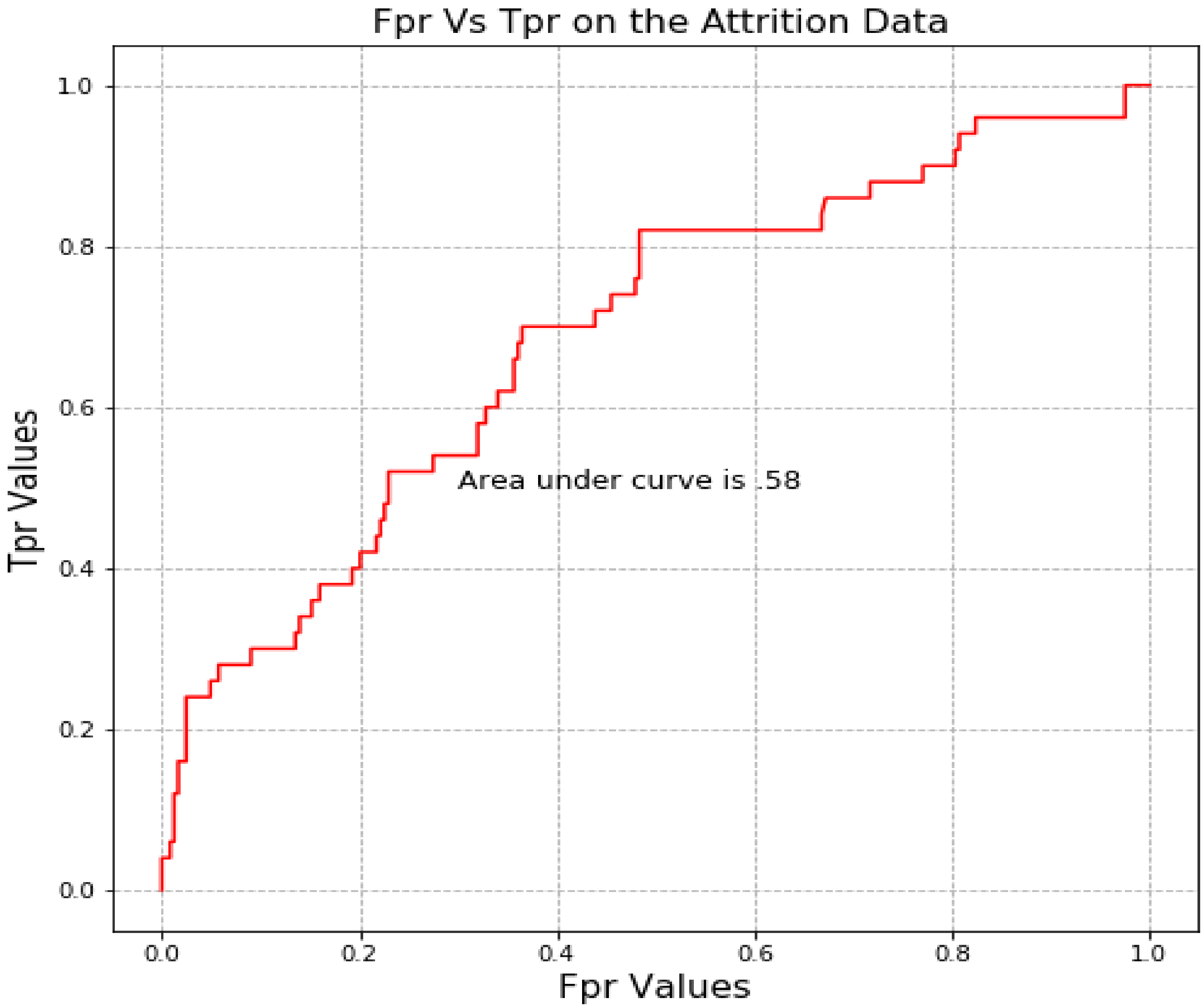
. FPR = *FP / (FP + TN)* ; fpr = 16 / (35+16) ; FPR = 0.3137254901960784

. TPR = *TP /(TP + FN)* ; tpr = 145 / (145 + 98) ; TPR = 0.5967078189300411

. *Area under the curve value :-*

. *r^2 = ssr / sst , ssr = variation explained by the variables*
sst = total variation(mean to real value

. ROC CURVE SCORE :- 0.5818895063746322



Fpr Vs Tpr on the Attrition Data
Tpr Values
Fpr Values
Area under curve is .58
0.0
0.2
0.4
0.6
0.8
1.0

. F1 SCORE :-

f1_score = 2 * (Precision * recall)/ (precision + recall)

. precision = TP /(TP+FP) ; precision = 145 / (145 + 16) ; precision = 0.9006211180124224

. recall = TP / (TP+FN) ; recall = 145 /(145 +98) ; recall = 0.5967078189300411

. f1_score = 2 * (90 * 59)/(90 + 59) ; f1_score :- 71.2751677852349

summary

1) conf_mat accuracy = 61.224489795918366
2) fpr = 0.3137254901960784
3) tpr = 0.5967078189300411
4) f1_score = 71.2751677852349

\# higher the accuracy and higher the tpr (recall) better the model , lower the fpr better the model .

\# higher the f1 score , better the model .