**A PROJECT REPORT**

**ON**

**EMPLOYEE ATTRITION ANALYSIS**

**BY**

**ANIBABA AYODEJI TEMILOLUWA**

# ABSTRACT

Staff attrition refers to the loss of employees through a natural process, such as retirement, resignation, elimination of a position, personal health, or other similar reasons. With attrition, an employer will not fill the vacancy left by the former employee.

The main aim of this project is to identify reasons employees are prone to leave a certain company X; identify employees prone to leave and predict future employees who would leave.

# TABLE OF CONTENTS

[ABSTRACT 2](#_Toc43471212)

[TABLE OF CONTENTS 3](#_Toc43471213)

[CHAPTER ONE 4](#_Toc43471214)

[OBTAINING THE DATA 4](#_Toc43471215)

[CHAPTER 3 5](#_Toc43471216)

[VISUALIZING THE DATASET USING TABLEAU 5](#_Toc43471217)

[CHAPTER 4 11](#_Toc43471218)

[DATA ANALYSIS WITH PYTHON 11](#_Toc43471219)

[4.1 Preparing the Datasets 11](#_Toc43471220)

[4.2 Analyzing the Correlation Matrix generated 14](#_Toc43471221)

[4.3 Analyzing the Kernel Density Estimate Plot. 15](#_Toc43471222)

[4.4 Exploring the data 15](#_Toc43471223)

[CHAPTER FIVE 19](#_Toc43471224)

[EVALUATION AND RESULTS 19](#_Toc43471225)

[5.1 Creating the Prediction Models using Machine Learning Algorithm 19](#_Toc43471226)

[5.2 Building the Prediction Models 19](#_Toc43471227)

[5.3 Results 24](#_Toc43471228)

[CHAPTER 6 26](#_Toc43471229)

[CONCLUSION 26](#_Toc43471230)

[REFERENCES 27](#_Toc43471231)

# CHAPTER ONE

# OBTAINING THE DATA

The dataset was downloaded from the Internship Environment in the form of an Excel file which contained two main sheets of data.

* Employees who left data set: 3571 observations, 10 variables
* Existing Employees data set: 11429 observations, 9 variables

On obtaining the data, a quick glimpse at the data head() of both datasets showed that:

1. **Ex-employee data set head():**

* Most of the ex-employees had low salary.
* There were no promotions in the last 5 years.
* The department is mostly sales.
* There were no work accidents.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Emp ID | satisfaction\_ level | last\_ evaluation | number\_ project | average\_ montly\_ hours | time\_ spend\_ company | Work\_ accident | promotion\_ last\_ 5years | dept | salary |
| 2001 | 0.58 | 0.74 | 4 | 215 | 3 | 0 | 0 | sales | low |
| 2002 | 0.82 | 0.67 | 2 | 202 | 3 | 0 | 0 | sales | low |
| 2003 | 0.45 | 0.69 | 5 | 193 | 3 | 0 | 0 | sales | low |
| 2004 | 0.78 | 0.82 | 5 | 247 | 3 | 0 | 0 | sales | low |

1. **Existing Employees dataset head():**

* The salary ranges from low-medium.
* Sales is still the predominant department.
* There have been no promotions in the past 5 years
* Existing employees have had no work accidents**.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Emp ID | satisfaction\_ level | last\_ evaluation | number\_ project | average\_ montly\_ hours | time\_ spend\_ company | Work\_ accident | promotion\_ last\_ 5years | dept | salary |
| 1 | 0.38 | 0.53 | 2 | 157 | 3 | 0 | 0 | sales | low |
| 2 | 0.8 | 0.86 | 5 | 262 | 6 | 0 | 0 | sales | medium |
| 3 | 0.11 | 0.88 | 7 | 272 | 4 | 0 | 0 | sales | medium |
| 4 | 0.72 | 0.87 | 5 | 223 | 5 | 0 | 0 | sales | low |

Thereafter, the actual datasets were visualized on Tableau

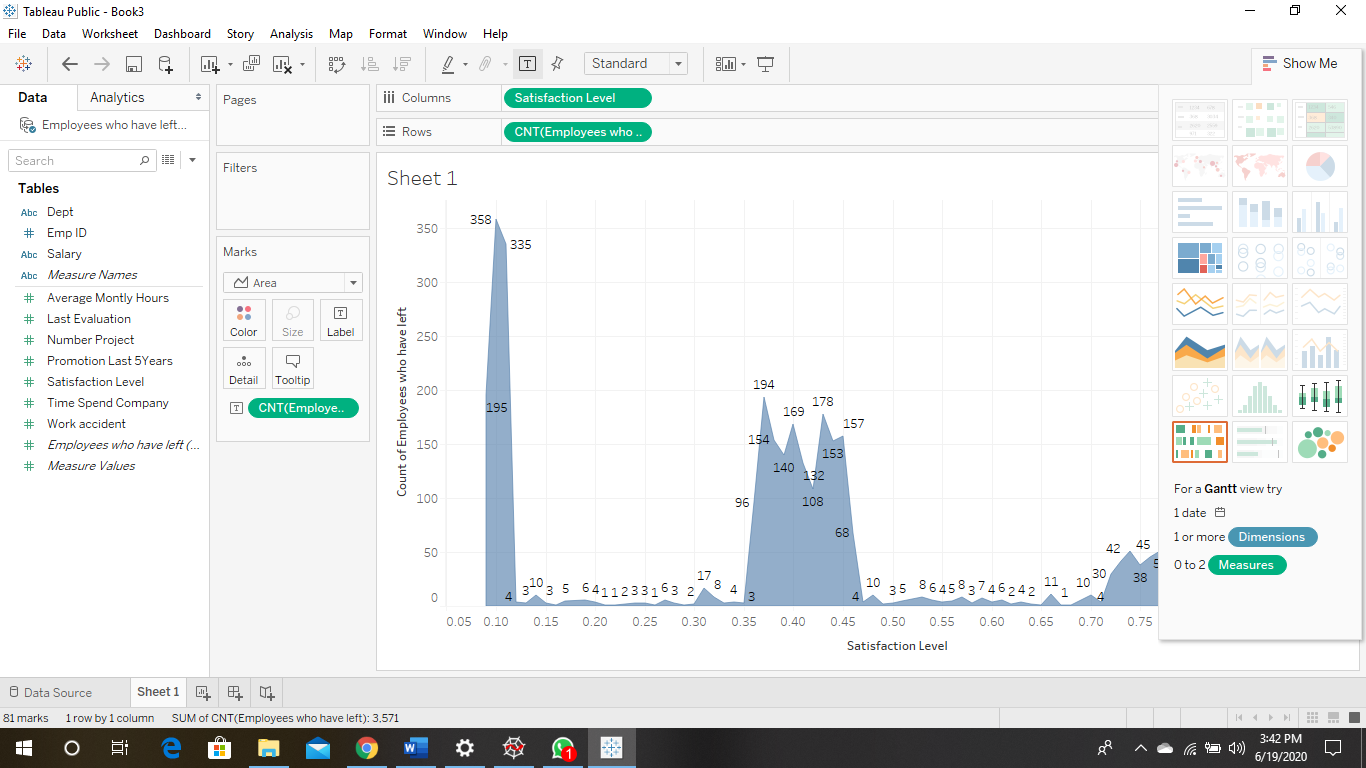
# CHAPTER 3

# VISUALIZING THE DATASET USING TABLEAU

To understand the data set further, a quick Tableau visualization was carried out. This involved visualizing Ex-employee data set; understanding the relationship between variables and the number of observations (ex-employees count) and understanding the relationship between variables if any.

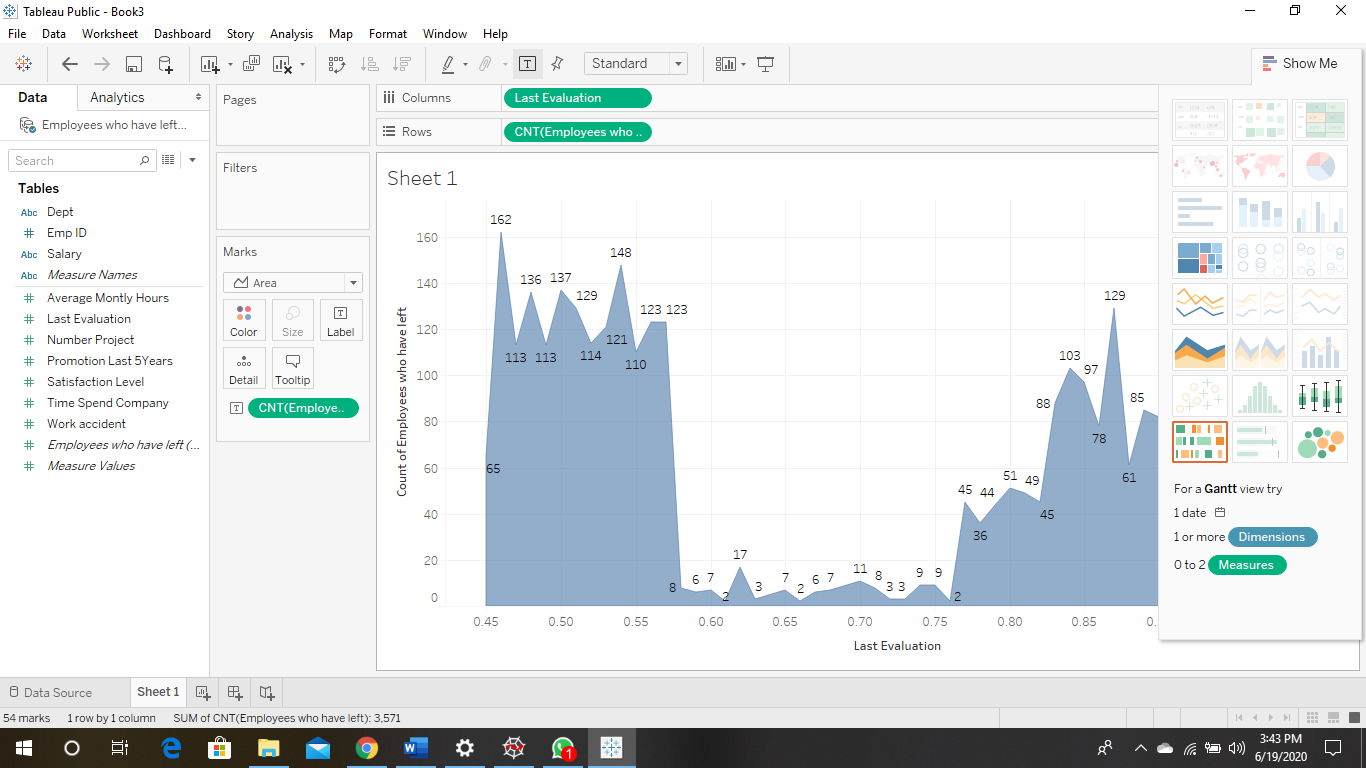
1. EX-EMPLOYEE COUNT VS SATISFACTION LEVEL

* Employees who left produced 3 major clusters (Little satisfaction, average satisfaction and High satisfaction) as shown the area chart visualization below.



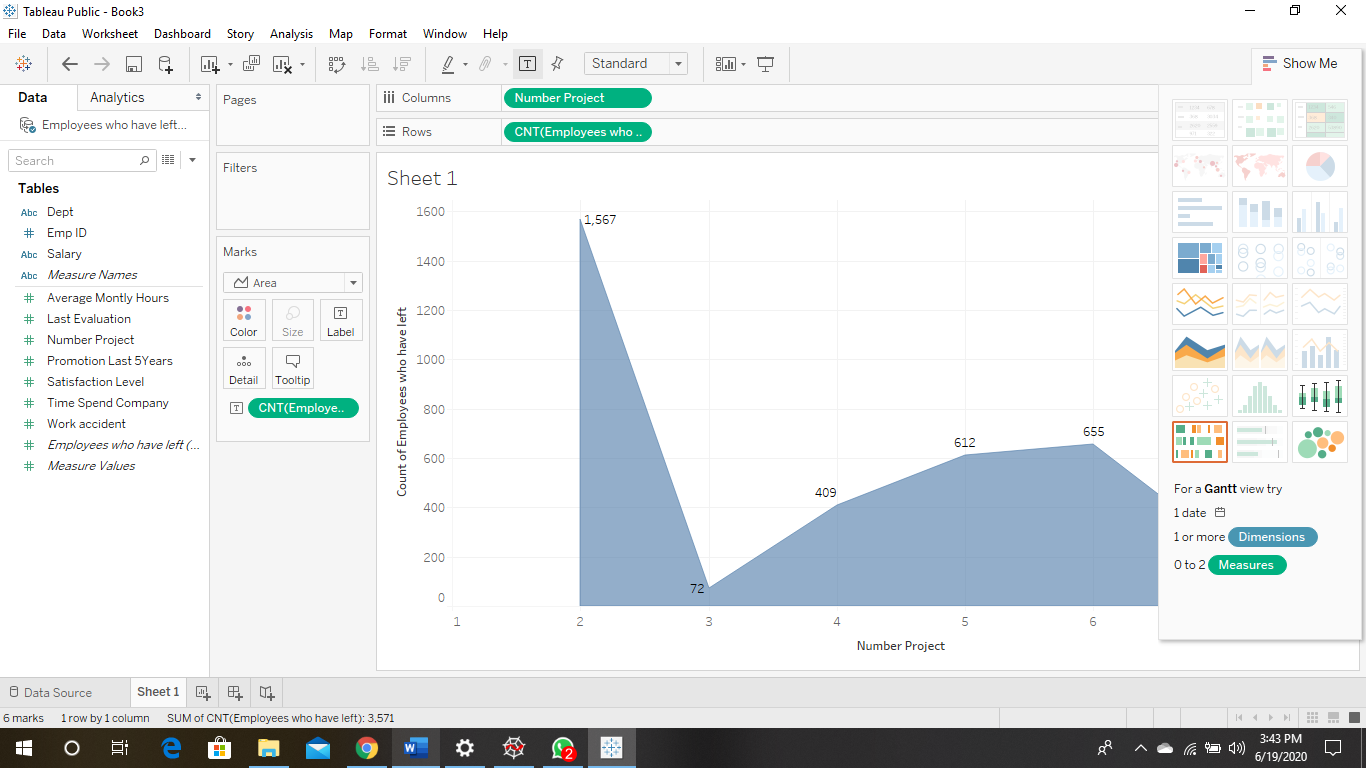
1. EX-EMPLOYEE COUNT VS LAST EVALUATION

* Employees who left had last evaluations that vary between low (0.4-0.6) and high (0.8-1.0) as shown in the area chart below.



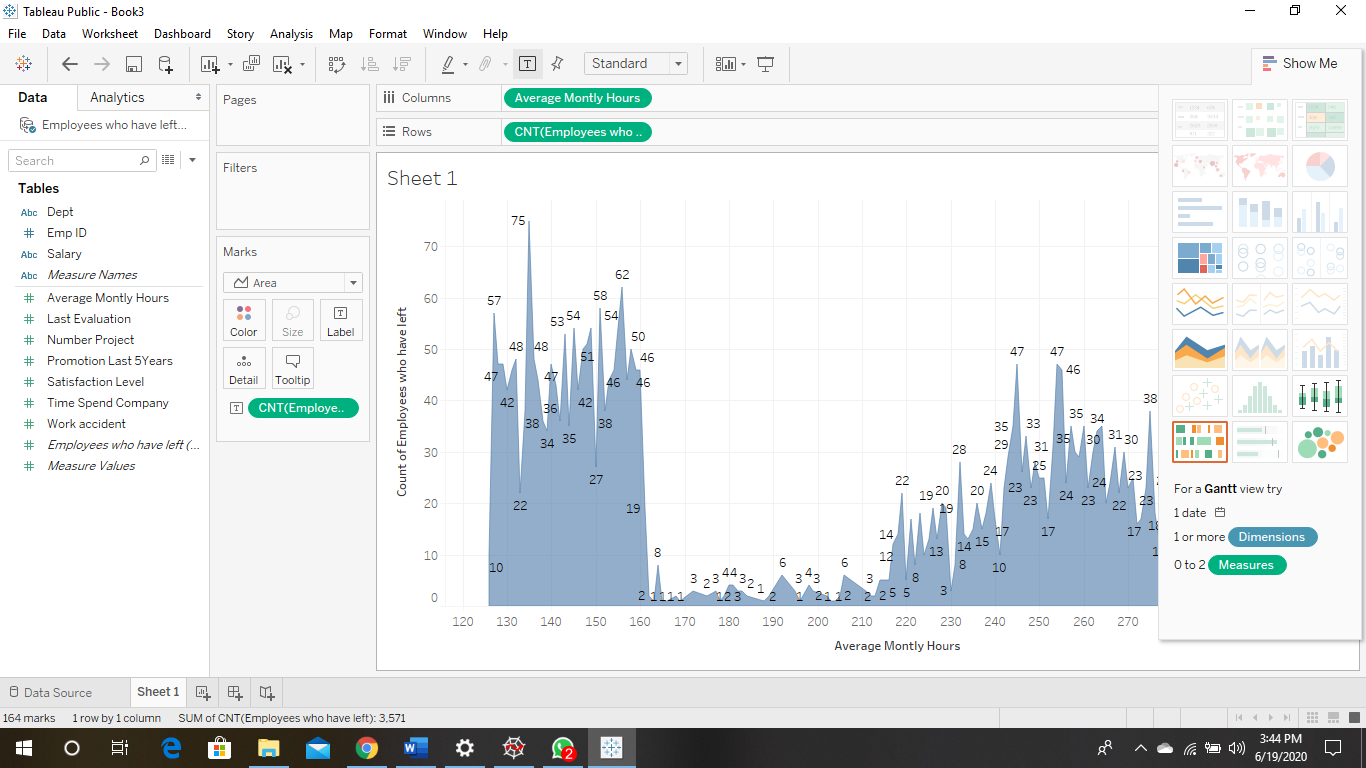
1. EX-EMPLOYEE COUNT VS NUMBER OF PROJECTS

* Employees involved in less projects (2) left and more employees left as the project number increased (from between 4-6).



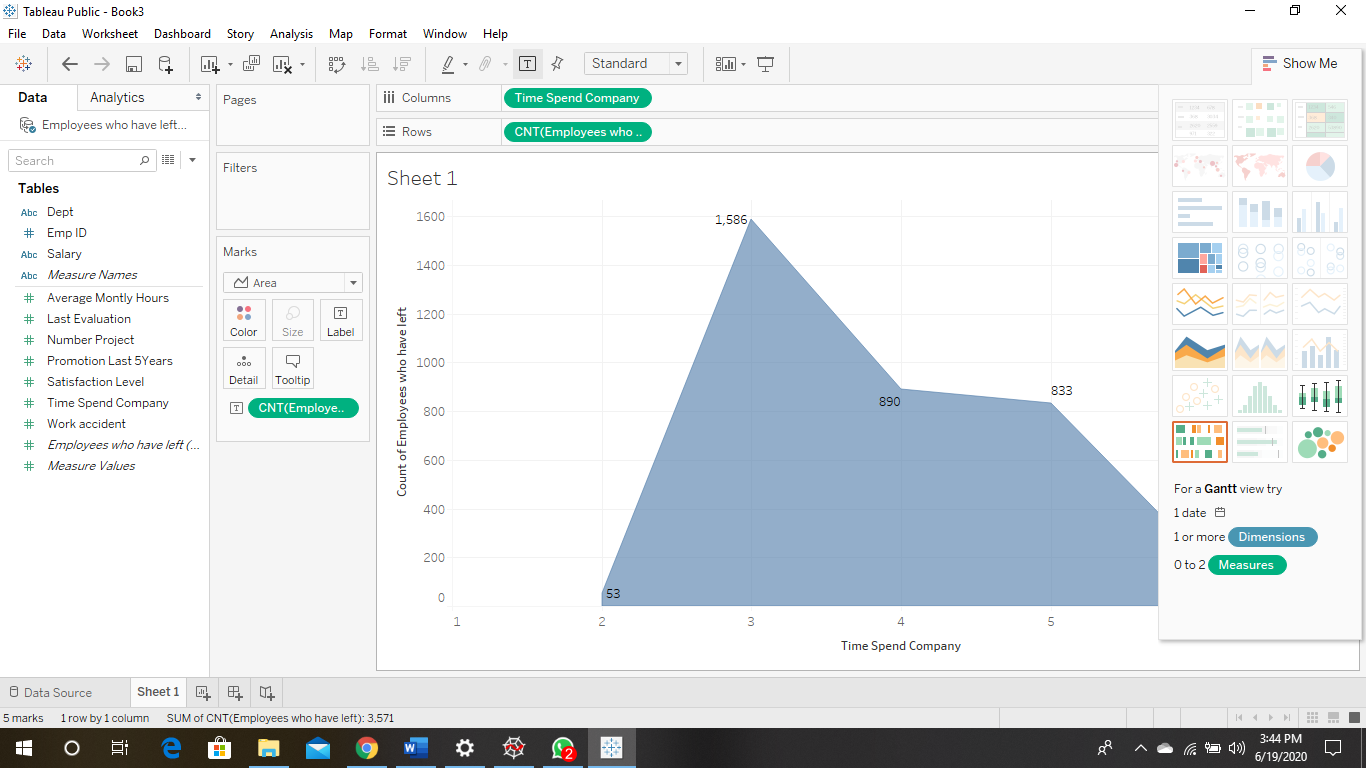
1. EX-EMPLOYEE COUNT VS AVERAGE HOURS MONTHLY

* Employees who left varied between low and high working hours.



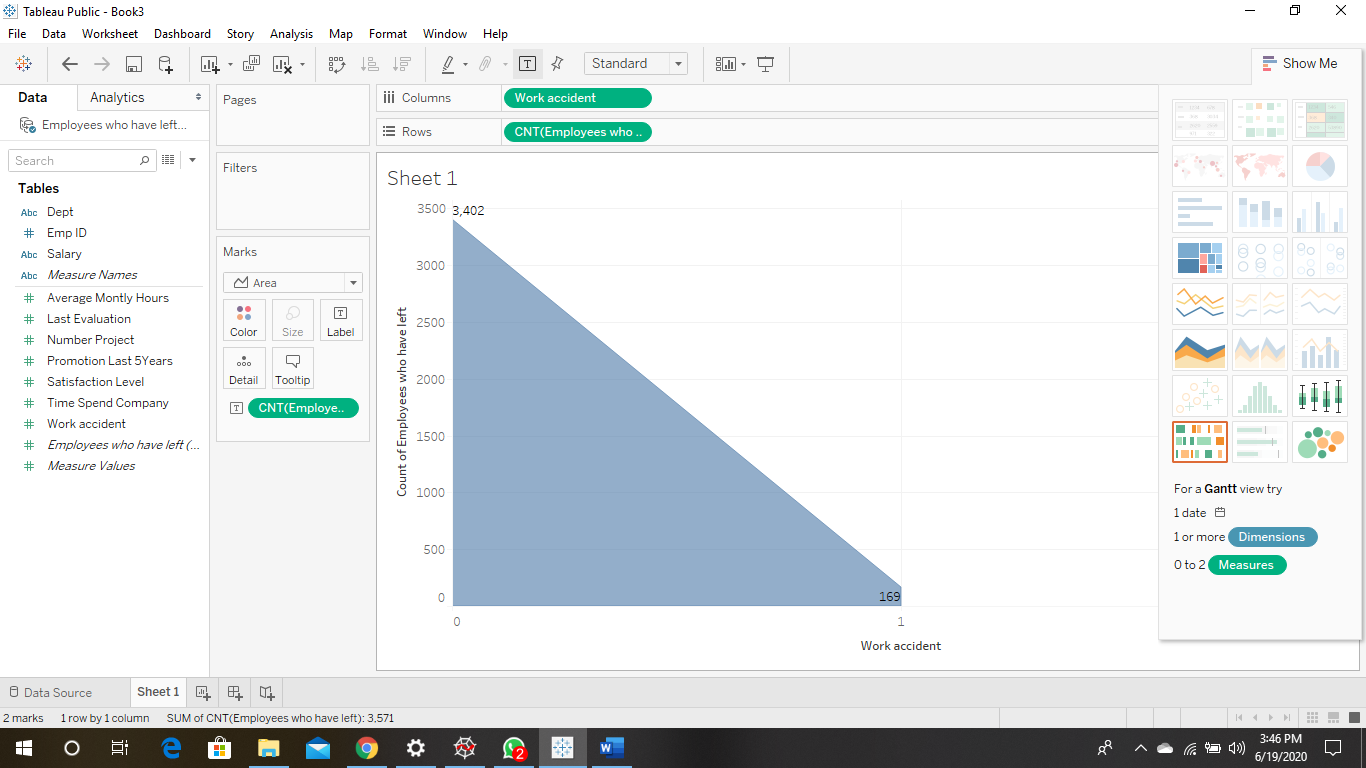
1. EX-EMPLOYEE COUNT VS TIME SPENT AT COMPANY

* Majority of employees who left had been at the company for between 3-5 years



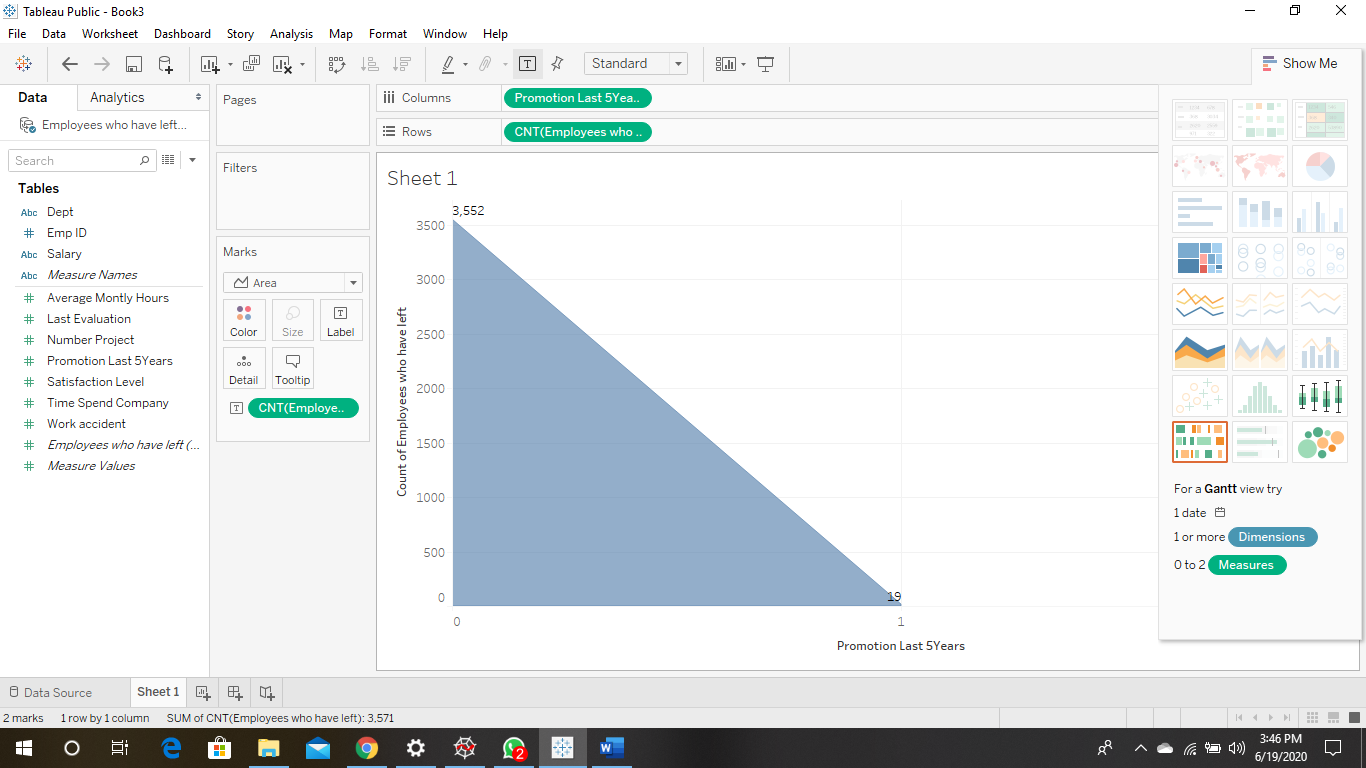
1. EX-EMPLOYEE COUNT VS WORK ACCIDENTS

* Employees decisions to leave were not largely based on work accidents.



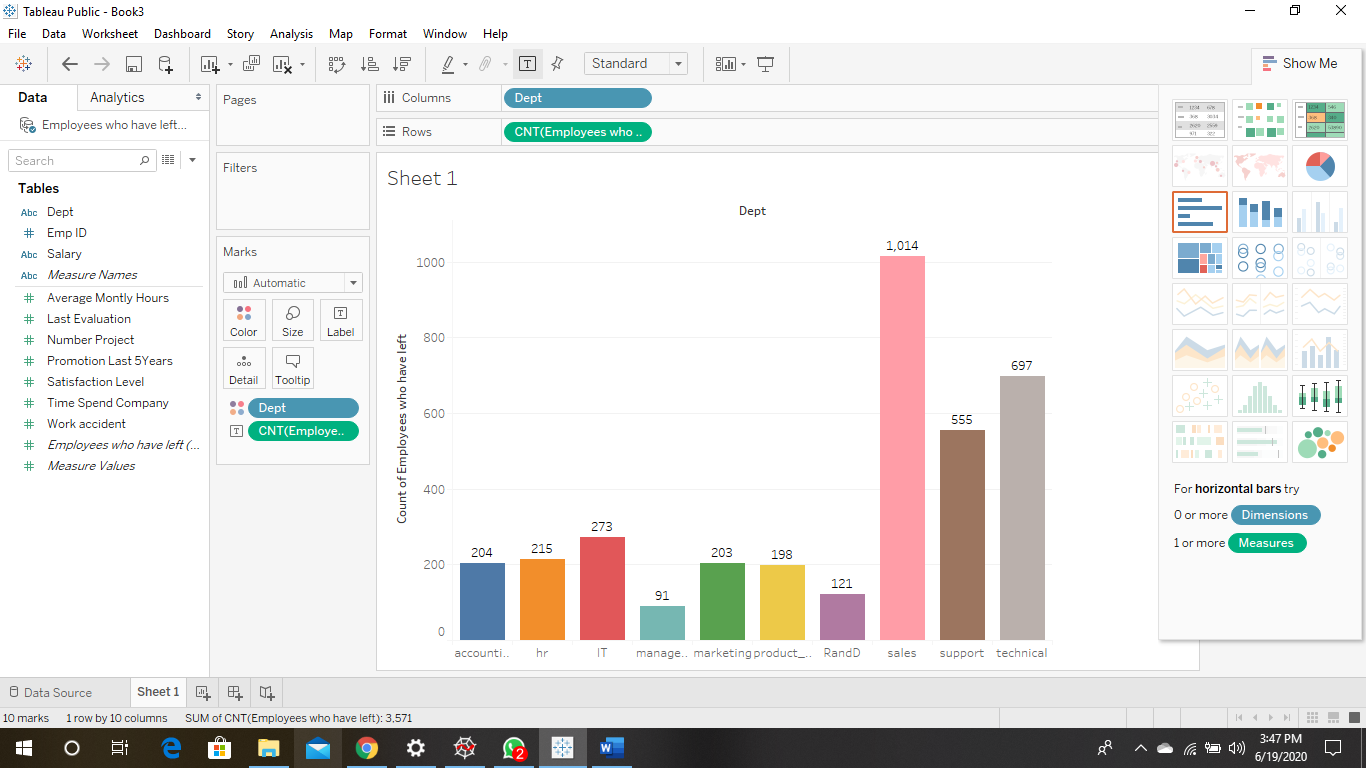
1. EX-EMPLOYEE COUNT VS PROMOTION IN LAST 5 YEARS

* Most Employees who left had not had a promotion in the last 5 years.



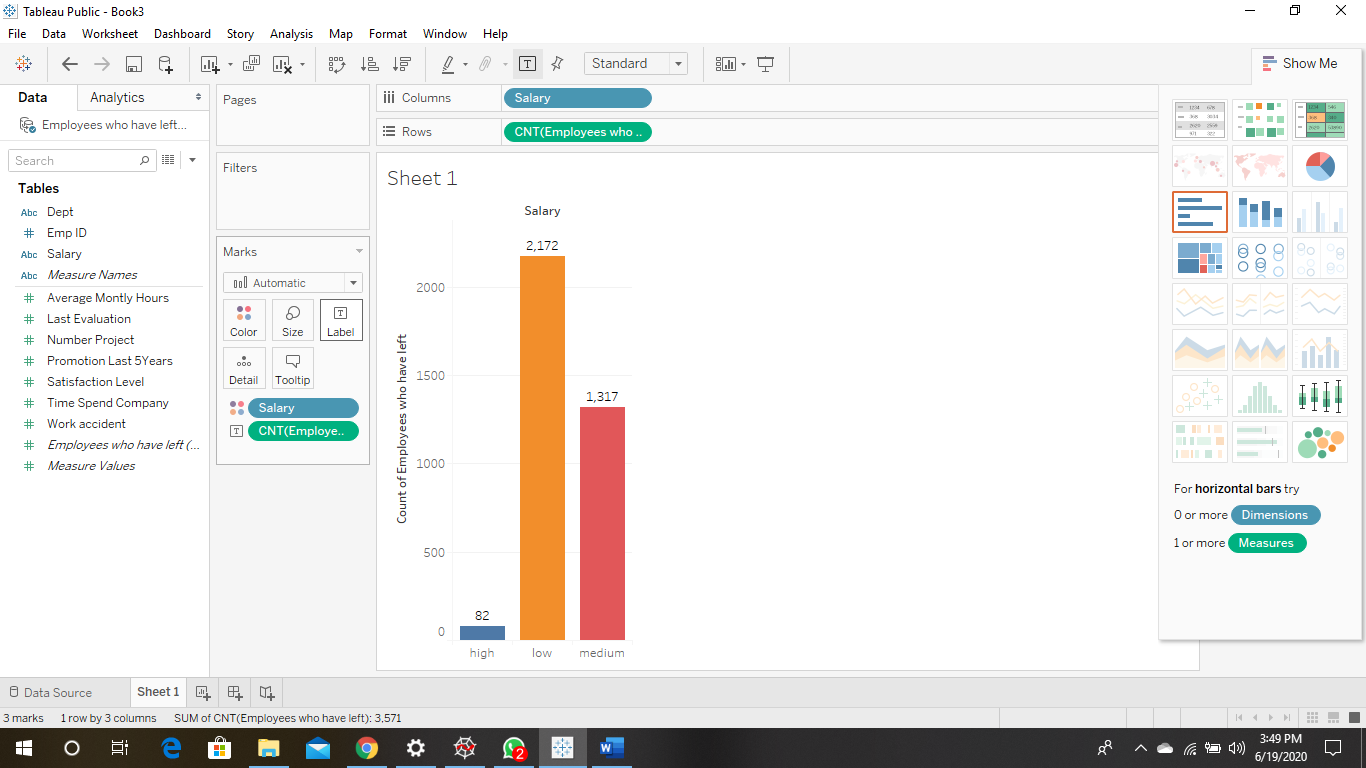
1. EX-EMPLOYEE COUNT VS DEPARTMENT

* Most employees who left were in the sales department, with employees in the technical and support departments following closely.



1. EX-EMPLOYEE COUNT VS SALARY

* Most employees who left fell in the low-medium salary range.



# CHAPTER 4

# DATA ANALYSIS WITH PYTHON

## 4.1 **Preparing the Datasets**

To analyze both datasets with python, it would be effective to examine the datasets to ensure its readability to avoid wrong results.

This involved:

* Checking for missing or Null data in the dataset.
* Converting the salary and department feature labels from string to numbers
* View statistical properties of the datasets
* Eliminate the ‘Employee Id’ column as it is not useful.
* Confirm number of observations in the datasets
* Plot a correlation matrix to show mathematical relations between variables.
* Use a Kernel Density Estimate plot to confirm the classification of clusters

To do this, both datasets were loaded into one sheet in Excel as a .csv file. This sheet was then called using the pd.read\_csv() into a pandas data frame in Python. For easy data manipulation, a column “Left” was added to the new .csv file created with 0 values for Existing employees and 1 for Ex-employees.

The codes run are as follows:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import matplotlib as matplot

import seaborn as sns

df = pd.read\_csv(r'C:\Users\AYODEJI\.spyder-py3\problemcase.csv')

print(df)

df = df.rename(columns={'satisfaction\_level': 'satisfaction',

'last\_evaluation': 'evaluation',

'number\_project': 'projectCount',

'average\_montly\_hours': 'averageMonthlyHours',

'time\_spend\_company': 'yearsAtCompany',

'Work\_accident': 'workAccident',

'promotion\_last\_5years': 'promotion',

'sales' : 'department',

'left' : 'turnover'

})

print(df)

# Check to see if there are any missing values in our data set

print(df.isnull().any())

# Get a quick overview of what we are dealing with in our dataset

print(df.head())

#data exploration

# The dataset contains 10 columns and 14999 observations

print(df.shape)

# Check the type of our features.

print(df.dtypes)

# Looks like about 76% of employees stayed and 24% of employees left.

# NOTE: When performing cross validation, its important to maintain this turnover ratio

turnover\_rate = df.turnover.value\_counts() / len(df)

print(turnover\_rate)

# Display the statistical overview of the employees

t=df.describe()

print(t)

from sklearn import preprocessing

#creating labelEncoder

le = preprocessing.LabelEncoder()

# Convert these variables into categorical variables

df["dept"] = df["dept"].astype('category').cat.codes

df["salary"] = df["salary"].astype('category').cat.codes

#Correlation Matrix

corr = df.corr()

sns.heatmap(corr,annot=True,cmap='seismic',

xticklabels=corr.columns.values,

yticklabels=corr.columns.values)

plt.title('Heatmap of Correlation Matrix')

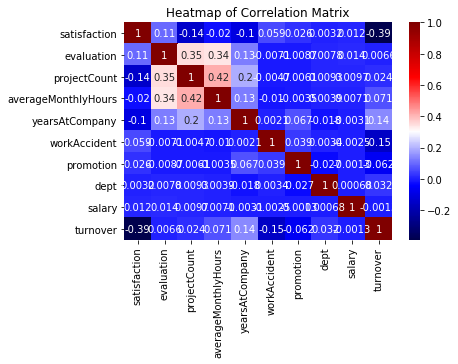
plt.show()

print(corr)

The code ran successfully with no errors. The result of the code is as follows:

* No null data were identified.
* The ‘Emp ID’ column was successfully deleted.
* The Salary and Department columns now have integers as elements.
* The dataset is confirmed to have 14999 observations in total, 11428 Existing Employees and 3571 ex-employees.
* A correlation matrix was generated.

4.2 Analyzing the Correlation Matrix generated



From the correlation matrix generated,

From the heatmap, there is a **positive(+)** correlation between projectCount, averageMonthlyHours, and evaluation. Which could mean that the employees who spent more hours and did more projects were evaluated highly.  
For the **negative(-)** relationships, turnover and satisfaction are highly correlated. I'm assuming that people tend to leave a company more when they are less satisfied.

This further supports our initial inferences from our Tableau Visualization.

**4.4 Exploring the data**

To further explore the data and create a prediction model, it was imperative that we use a decision tree classifier to rank the importance of each variable to enable us select the most important for modeling. The following code was run to achieve that:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv(r'C:\Users\AYODEJI\.spyder-py3\problemcase.csv')

print(df)

from sklearn import preprocessing

#creating labelEncoder

le = preprocessing.LabelEncoder()

# Convert these variables into categorical variables

df["dept"] = df["dept"].astype('category').cat.codes

df["salary"] = df["salary"].astype('category').cat.codes

#spliting features into independent and dependent variables

X= df.iloc[:,[0,1,2,3,4,5,6,7,8]].values

Y= df.iloc[:,9].values

print(df)

#split dataset into train and test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,Y,train\_size=0.76,test\_size=0.24,random\_state=1)

from sklearn.tree import DecisionTreeClassifier

#fitting decision tree to training set

classifier1=DecisionTreeClassifier(criterion='entropy', random\_state=1)

classifier1.fit(X\_train, Y\_train)

#predicting test set result

Y\_pred= classifier1.predict(X\_test)

print(Y\_pred)

var\_prob= classifier1.predict\_proba(X\_test)

var\_prob[0,:]

#confusion matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(Y\_test, Y\_pred)

#getting accuracy of the test set

Accuracy=classifier1.score(X\_test,Y\_test)

print(Accuracy)

## plot the importances ##

importances = classifier1.feature\_importances\_

feat\_names = df.drop(['turnover'],axis=1).columns

indices = np.argsort(importances)[::-1]

plt.figure(figsize=(12,6))

plt.title("Feature importances by DecisionTreeClassifier")

plt.bar(range(len(indices)), importances[indices], color='lightblue', align="center")

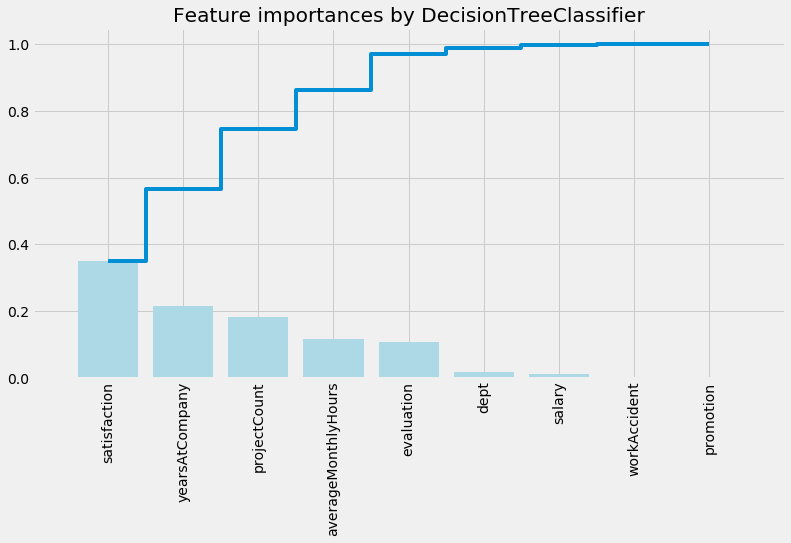
plt.step(range(len(indices)), np.cumsum(importances[indices]), where='mid', label='Cumulative')

plt.xticks(range(len(indices)), feat\_names[indices], rotation='vertical',fontsize=14)

plt.xlim([-1, len(indices)])

plt.show()

The code ran successfully without any errors. The plot of the Features by level of importance is as follows:



In a tabular format, the features and corresponding levels of importance are:

|  |  |
| --- | --- |
| **FEATURE** | **IMPORTANCE** |
| Satisfaction | 0.355906 |
| yearAtCompany | 0.198967 |
| Project Count | 0.176788 |
| Average monthly hours | 0.142487 |
| Evaluation | 0.104795 |
| Dept | 0.012754 |
| Salary | 0.00571052 |
| WorkAccident | 0.00235654 |
| Promotion | 0.000236112 |

The top 5 most important features will be used to create our Prediction Models using Machine Learning Algorithms.

**CHAPTER FIVE**

**EVALUATION AND RESULTS**

**5.1 Creating the Prediction Models using Machine Learning Algorithm**

5 Machine Learning Algorithms were used to create Prediction Models with the data sets. The prediction models also showed the probabilities for each employee in the test set to either leave or remain with the company based on the features selected.

1. **Logistic Regression:** This machine learning algorithm has binary probability outcomes. This was the first choice model as the dataset already has binary outcomes (0,1) for Existing and Ex-Employees.
2. **Decision Tree Classification**: This is a supervised machine learning algorithm mostly used in classification problems. It splits population datasets into two or more homogenous sets based on most significant splitter in variables.

**5.2 Building the Prediction Models**

Since the data set was to be same for the predictions, all codes were run in one for ease.

The code run was as follows:

import pandas as pd

import numpy as np

df = pd.read\_csv(r'C:\Users\AYODEJI\.spyder-py3\problemcase.csv')

print(df)

df = df.rename(columns={'satisfaction\_level': 'satisfaction',

'last\_evaluation': 'evaluation',

'number\_project': 'projectCount',

'average\_montly\_hours': 'averageMonthlyHours',

'time\_spend\_company': 'yearsAtCompany',

'Work\_accident': 'workAccident',

'promotion\_last\_5years': 'promotion',

'sales' : 'department',

'left' : 'turnover'

})

from sklearn import preprocessing

#creating labelEncoder

le = preprocessing.LabelEncoder()

# Convert these variables into categorical variables

df["dept"] = df["dept"].astype('category').cat.codes

df["salary"] = df["salary"].astype('category').cat.codes

#spliting features into independent and dependent variables

X= df.iloc[:,[0,1,2,3,4,5,6,7,8]].values

Y= df.iloc[:,9].values

print(df)

#split dataset into train and test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X,Y,train\_size=0.76,test\_size=0.24,random\_state=1)

#feature scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_tain = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state=1, solver='lbfgs',max\_iter=300)

classifier.fit(X\_train, Y\_train)

#predict test set result

Y\_pred = classifier.predict(X\_test)

var\_prob= classifier.predict\_proba(X\_test)

var\_prob[0,:]

#confusion matrix

from sklearn.metrics import confusion\_matrix

cm =confusion\_matrix(Y\_test,Y\_pred)

from sklearn.metrics import accuracy\_score

print(accuracy\_score(Y\_test,classifier.predict(X\_test)))

from sklearn.metrics import classification\_report

print(classification\_report(Y\_test, classifier.predict(X\_test)))

from sklearn.tree import DecisionTreeClassifier

#fitting decision tree to training set

classifier1=DecisionTreeClassifier(criterion='entropy', random\_state=1)

classifier1.fit(X\_train, Y\_train)

#predicting test set result

Y\_pred= classifier1.predict(X\_test)

print(Y\_pred)

var\_prob= classifier1.predict\_proba(X\_test)

var\_prob[0,:]

#confusion matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(Y\_test, Y\_pred)

#getting accuracy of the test set

Accuracy=classifier1.score(X\_test,Y\_test)

print(Accuracy)

## plot the importances ##

importances = classifier1.feature\_importances\_

feat\_names = df.drop(['turnover'],axis=1).columns

indices = np.argsort(importances)[::-1]

plt.figure(figsize=(12,6))

plt.title("Feature importances by DecisionTreeClassifier")

plt.bar(range(len(indices)), importances[indices], color='lightblue', align="center")

plt.step(range(len(indices)), np.cumsum(importances[indices]), where='mid', label='Cumulative')

plt.xticks(range(len(indices)), feat\_names[indices], rotation='vertical',fontsize=14)

plt.xlim([-1, len(indices)])

plt.show()

The code was run successfully without any errors.

5.**3 Results**

The various Accuracies, Precision and Confusion matrix for each model used is presented in a tabular format as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Logistic Regression** | | **Decision Tree Classification** | |
| Model Accuracy | 64.2% | | 97.80% | |
| Precision | 48.57% | | 94.66% | |
|  |
| Confusion Matrix |
| Actual | 0 | 1 | 0 | 1 |
| 0 | 1693 | 1034 | 2677 | 50 |
| 1 | 255 | 618 | 29 | 844 |

The model with the best accuracy and precision is the **Decision Tree Classification** as it has the least **False Positives and False Negatives**.

**CHAPTER 6**

**CONCLUSION**

In conclusion, this report highlights the analysis done on the Employee Attrition Datasets to explain reasons why employees leave, the types of employees prone to leave and the use of its features to create prediction models for future Employee turnover.

Most of the visualization was carried out using Tableau, with codes run in Python using its numpy, pandas, matplotlib, seaborn, scikit libraries.

In summary,

**Reasons Employees Leave:**  
1. Employee satisfaction, Time Spent at Company, and number of projects were the three most important features in determining turnover.   
2. Underworking and Overworking an employee could cause them to leave; employees that had 2,6, or 7 project count were at risk of leaving the company.  
3. Employees with either really high or low evaluations should be taken into consideration for high leave rate.  
4. Employees with low to medium salaries are the bulk of employees leaving.  
5. The most common departments of Ex-Employees—Sales Executive, Support and Technical.  
 **The Types of Employees Prone to Leave are:**  
1) Employees with Low Satisfaction and High Evaluation.  
2) Employees with Moderate Satisfaction and Moderate Evaluation  
3) Employees with High Satisfaction and High Evaluation.  
  
**Predicting Future Employees Leaving:**  
The Decision Tree has the most Accuracy and Precision to predict whether an Existing employee would leave or not.

**REFERENCES**

[1][<https://medium.com/@monikapdb/employee-attrition-analysis-using-machine-learning-methods-73564358e87f>](http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/images/348px-IR_Sensor_Brightness_Principles.png)