# IMPROVEMENT IN EMPLOYEE RETENTION AND RESOURCE CONSUMPTION THROUGH MACHINE LEARNING TECHNIQUES

**Prateek Jaiswal** and Deepankur Kansal

submission date:19/06/2020

## Abstract:

**Employee attrition is one of the major problems faced by many companies, start-ups nowadays. As loss of an employee incurs huge cost to the company in terms of lost productivity, recruitment and training costs. The biggest challenge faced by many organizations today is how to retain their employees. The aim of this research is to improve employee attrition or increasing the tenure of the employees with increasing the value of the employee to the company while maintaining the resource constraint. Employee attrition prediction, Employee value prediction and Resource consumption models are developed so as to work upon these and find the optimal solution for the attributes of an employee by maximizing employee value and minimizing resource consumption and thus increase the tenure of the employees. Predictive Models are developed based on several machine learning techniques: Random Forest, Gaussian Naive Bayes, Multinomial Naive Bayes, Support Vector Regression, Logistic Regression, Lasso Regression, Elastic Net Regression, Linear Regression, Ridge Regression, Bayesian Ridge Regression, Huber Regression, Theil-Sen Regression, Least-angle regression and XGB classifier. Employee Value Prediction model uses K-means Clustering algorithm. Employee Data (consisting of 35 features) is collected from IBM R & D lab released data-set on Employee attrition and performance. Results show that the expected tenure of 84.83% of the total employees have increased by on an average of approx. 174.17%, which is very significant.**

## 1. Introduction:

Employees are the backbone of an organization. Hence, the retention of the employees is important in keeping the organization on track. In order to retain the best talents, strategies aimed at satisfying employee’s needs are implemented, regardless of global companies or small-sized firms [1].

Employee attrition incurs huge cost in terms of lost productivity, recruitment and training costs. Between costs associated with separation, loss of productivity, recruitment, interviewing, training, and on boarding, the loss of a single employee is estimated to cost the company one third of that individual’s annual salary [2].There could be various factors responsible for employee leaving the company. The major challenge faced by many organizations is to retain their employees. This research paper aims to develop several models to solve this problem. Firstly, A predictive model for employee Tenure prediction using several machine learning techniques such as Random Forest, Gaussian Naive Bayes, Multinomial Naive Bayes, Support Vector Regression, Logistic Regression, Lasso Regression, Elastic Net Regression, Linear Regression, Ridge Regression, Bayesian Ridge Regression, Huber Regression, Theil-Sen Regression, Least-angle regression and XGB classifier is developed. Mode of all the predictions is taken so as to obtain an optimal tenure prediction score. After predicting the expected tenure of the employees, a model generating the value of an employee for the company is developed by clustering the employees into many clusters using K-means clustering algorithm. The employee value is generated considering the employee attributes (from data-set) and the predicted tenure forming one of the attributes. Each employee is treated as an individual entity and the suggested preventive attrition measures are at an individual level. Resource Consumption model is further developed, using the weighted employee attributes, which evaluates the resources being used by an employee.

Then, Retention Technique selection model is developed to find the optimal solution for the attributes of an employee by maximizing the employee value and minimizing resource consumption. Finally, An updated tenure prediction model is developed which uses the generated optimal solution of employee attributes to predict the newly improved employee tenure using the previously developed predictive models.

The remainder of our paper is structured as follows. In the next section we discuss about literature review . In section 3, we describe the data. Section 4 states the objective of the research. Section 5 tells about the software used. Then, section 6 describes the methodology, sub-divided into two sections: Human Resource Management Models and machine learning Algorithms. Section 7 states the hypothesis, and then we present our results in section 8 and then finally section 9 draws together our main conclusions. Section 10 and section 11 are the references and appendix respectively.

## 2. Literature Review:

Omer and Laura (2015) said successful employee retention is essential to an organization's stability, growth and revenue. Organizations can achieve employee retention by developing few strategies. To improve employee training, management is expected to continue to develop training materials adjusted to the work program. And the employee training is a determining factor that can give effect on employee performance. However, when providing employee training, it is necessary to have standardized reference that can be used effectively and efficiently. To create employee satisfaction including task and work aspect, salary aspect, employee relationship, and employee promotion, the company's management is suggested to make improvements in an effort to improve employee satisfaction that affects employee performance [4]

In Regresion Analysis (2012), an effective human resource management practices namely employee empowerment, training and development, appraisal system compensation are the main factor for the success of a firm on employee retention. By using a multiple regression analysis, training and development, appraisal system compensation are significant to employee retention. Employees are the backbone of an organization. Hence, the retention of the employees is important in keeping the organization on track. In order to retain the best talents, strategies aimed at satisfying employee's needs are implemented, regardless of global companies or small-sized firms. Generally, organization would retain their personnel for a specified period to utilize their skills and competencies to complete certain projects or execute tasks. In another word, we can understand it as employee retention where the scope of task, is however, often larger than a simple task and more preferably a job in real world. [1]

In the book, "Factors Affecting Employee Attrition and Predictive Modelling Using IBM HR Data", establishing a predictive model for employee attrition involves data preprocessing with chi square versus logistic regression for feature selection, machine learning models and their comparison using the confusion matrix, precision, recall and f1-scores based on IBM HR Data Set.[6]

Analysis of RF, ANN, and SVM Regression Models are compared by Beijing Research Center accounting for the application of various regressive models for a specific use case of estimation (2017)[5]

Lucas' (2013) report that employers don't understand the expense of high employee turnover. Recruiting new staff is costly due to advertising and administrative expenses; time and resources for onboarding and training; as well as loss of productivity[3]

Dawley, Houghton, and Bucklew (2010) provide us with an examination on turnover decisions, or intentions, which are driven by how an employee perceives their fit in a company. In as much that person-organization fit enhances retention; it also impacts an employee's decision to leave a job. [13]

Hebenstreit (2008) shows job fit is relative to personality. He promoted the use of Enneagram, which is a personality and motivation system to recruit and retain employees. His article evaluated data from Enneagram and revealed nine personality subtypes, which interact to create a work environment conducive to retention.[16] Hebenstreit, R. P. (2008). A call to apply the principles of the enneagram in organizations to attract, retain and motivate employees. Enneagram Journal, 4-21.

As we continued our discussion on job fit, Cramer (1993) helps us to understand that tenure is an important indicator of turnover intention. Organizations with high levels of tenure have low turnover rates. Employees are more apt to stay if employers provide and invest training in their employees. [17] Cramer, D. (1993). Tenure commitment and satisfaction of college graduates in an engineering firm. Journal of Social Psychology, 133(6), 791-796.

Heathfield (2008) article supports our claim that greater satisfaction among employees is also contingent upon their training and development. They are key motivators encouraging employees to remain loyal and create a cohesive workforce.[18] Heathfield, S. M. (2008). Training and development for employee motivation and retention. The Lama Review, 20(2), 20

Gilmore and Turner (2010) reminds us that selection of new employees should be guided by person-organization fit, matching characteristics of individuals to a job and its culture.[14]

Gabriel et al (2014) tests the relationship between 120 Journal of Leadership, Accountability and Ethics Vol. 12(2) 2015 perceived fit and the overall attitude associated with an employee's positive experiences on the job. The study showed that there is direct correlation between Person-Organization fit and job satisfaction. They implied that direct assessment of fit initiated at recruitment improves fit as it aligns employee skills and values with that of the organization. They also noted that the perception of fit fluctuates over time. [15]

## 3. Objective:

Employee attrition incurs huge cost in terms of lost productivity, recruitment and training costs. Between costs associated with separation, loss of productivity, recruitment, interviewing, training, and on boarding, the loss of a single employee is estimated to cost the company one third of that individual's annual salary. The biggest challenge faced by many organizations today is how to retain their employees**.** The aim of this research is to:

In the first part, analyzing the data we develop models using various machine learning techniques to predict the expected tenure of the employees presently working in the company.

In the second part, clustering the employees into various clusters using k-means clustering, we develop the employee value prediction and employee resource consumption models and find the optimal value of attributes of an employee by maximizing the employee value and minimizing the resource consumption so as to improve the tenure of the employees using the updated attributes. This is aimed at retaining the employees.

## 4. Data:

### 4a).Raw data, source and Overview:

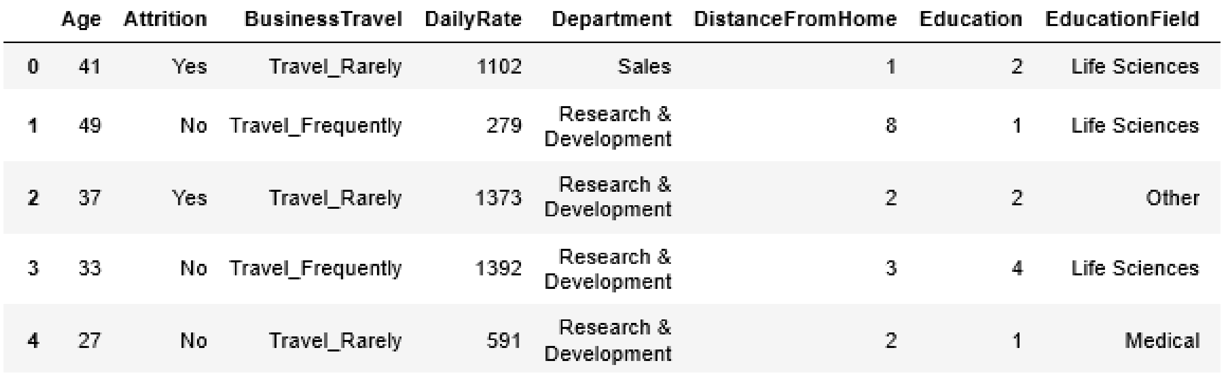
The Data taken is developed by IBM R and D lab to provide a standard for the HR management issues incorporated by a medium scale enterprise. It is developed to uncover the factors that lead to employee attrition and explore important questions and ideas leading to a fruitful outcome containing data features pertaining to every aspect of an employee concerning his/her professional life.

The Data itself is complete and is divided into two parts separating attrition and retention subsections of the employee. The initial training is done on attrition data set in the ratio of 80 percent train data and 20 percent test data and predicts values for the retention data set.

The data has total 35 features and 1470 employees (rows, columns)=(1470,35), the full data can be seen in this link:

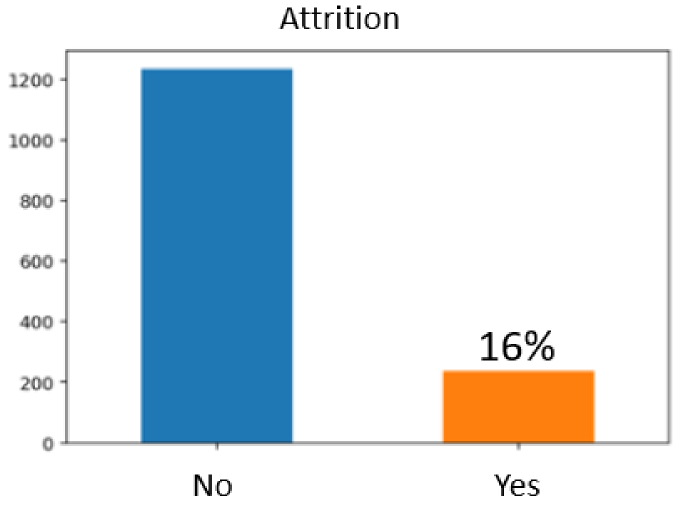
https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

Here is a small section of data (just first five examples) so as to give an overview how the data looks like.

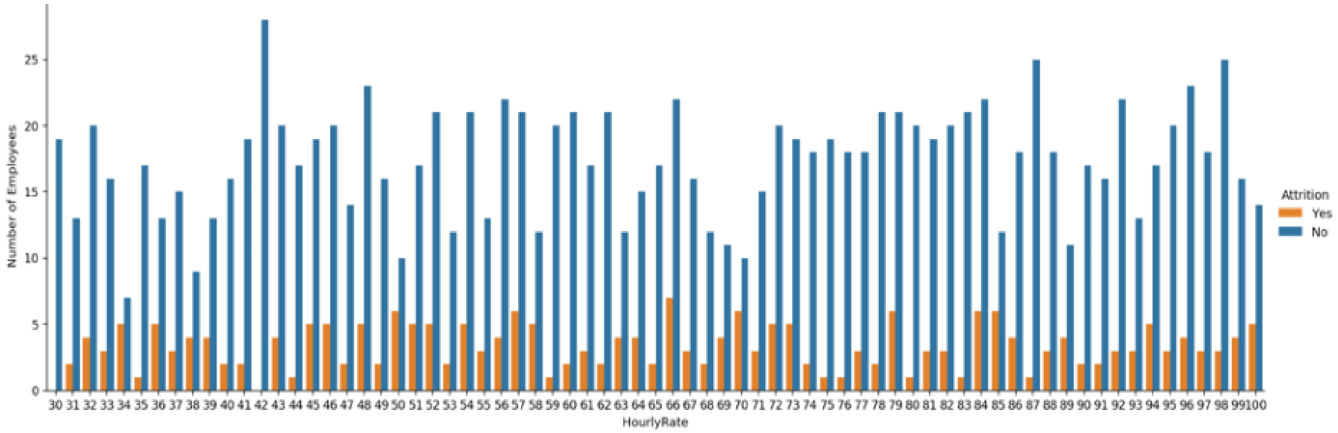


| | Age | Attrition | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationField |
|---|---|---|---|---|---|---|---|---|
| 0 | 41 | Yes | Travel_Rarely | 1102 | Sales | 1 | 2 | Life Sciences |
| 1 | 49 | No | Travel_Frequently | 279 | Research & Development | 8 | 1 | Life Sciences |
| 2 | 37 | Yes | Travel_Rarely | 1373 | Research & Development | 2 | 2 | Other |
| 3 | 33 | No | Travel_Frequently | 1392 | Research & Development | 3 | 4 | Life Sciences |
| 4 | 27 | No | Travel_Rarely | 591 | Research & Development | 2 | 1 | Medical |

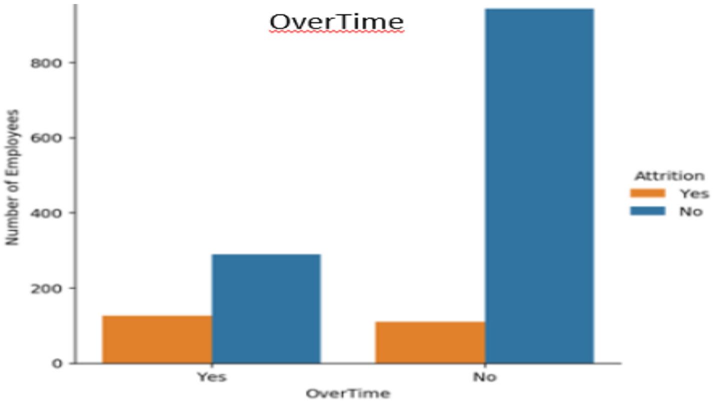
The dataset is well organised with no missing values. Target class is imbalance, with attrition rate of 16%.



Are employees leaving because they are poorly paid? Employees are paid an hourly rate of $30 to $100, and attrition seems to happen at every level regardless of employee hourly rate. This can be confirmed later at feature importance.



Overtime seems to be one of the key factors to attrition, as a larger proportion of overtime employees has departed.



## 3b).Data Preprocessing and Feature Engineering :

Data has to be preprocessed as machine learning models are better at reading numbers than words. Using label encoding, categorical data can be replaced with numbers.

Several new features have been added :

Features ['Department_0', 'Department_1', 'Department_2'] corresponding to ['Human Resources' 'Research & Development' 'Sales'] departments labeled as [0 1 2] respectively.

Features ['JobRole_0', 'JobRole_1', 'JobRole_2', 'JobRole_3','JobRole_4', 'JobRole_5', 'JobRole_6', 'JobRole_7', 'JobRole_8'] corresponding to ['Healthcare Representative' 'Human Resources' 'Laboratory Technician', 'Manager' 'Manufacturing Director' 'Research Director', 'Research Scientist' 'Sales Executive' 'Sales Representative'] job roles labeled as [0 1 2 3 4 5 6 7 8] respectively.

Features ['EducationField_0', 'EducationField_1', 'EducationField_2','EducationField_3', 'EducationField_4', 'EducationField_5'] corresponding to ['Human Resources' 'Life Sciences' 'Marketing' 'Medical' 'Other'] labeled as [0 1 2 3 4 5] respectively.

Features ['BusinessTravel_0', 'BusinessTravel_1','BusinessTravel_2'] corresponding to ['Non-Travel' 'Travel_Frequently' 'Travel_Rarely'] labeled as [0 1 2] respectively.

Features ['Gender_0','Gender_1'] corresponding to ['Female' 'Male'] labeled as [0 1] respectively.

Features ['MaritalStatus_0', 'MaritalStatus_1', 'MaritalStatus_2'] corresponding to ['Divorced' 'Married' 'Single'] labeled as [0 1 2] respectively.

Features ['Attrition_0','Attrition_1'] corresponding to ['No' 'Yes'] labeled as [0 1] respectively.

Features ['OverTime_0','OverTime_1'] corresponding to ['No' 'Yes'] labeled as [0 1] respectively.

The code for label encoding and feature engineering is:.

```
category_list=['Attrition','BusinessTravel','Department','EducationField','
Gender','JobRole','MaritalStatus','OverTime']
data =df
for s in category_list:
    label=LabelEncoder()
    data[s]=label.fit_transform(data[s])
    print(label.classes_)
    print(label.transform(label.classes_))
    data=pd.get dummies(data,columns=[s],prefix=[s])
df =data
print(df.columns)
df.head()
```

## 3b).Feature Selection:

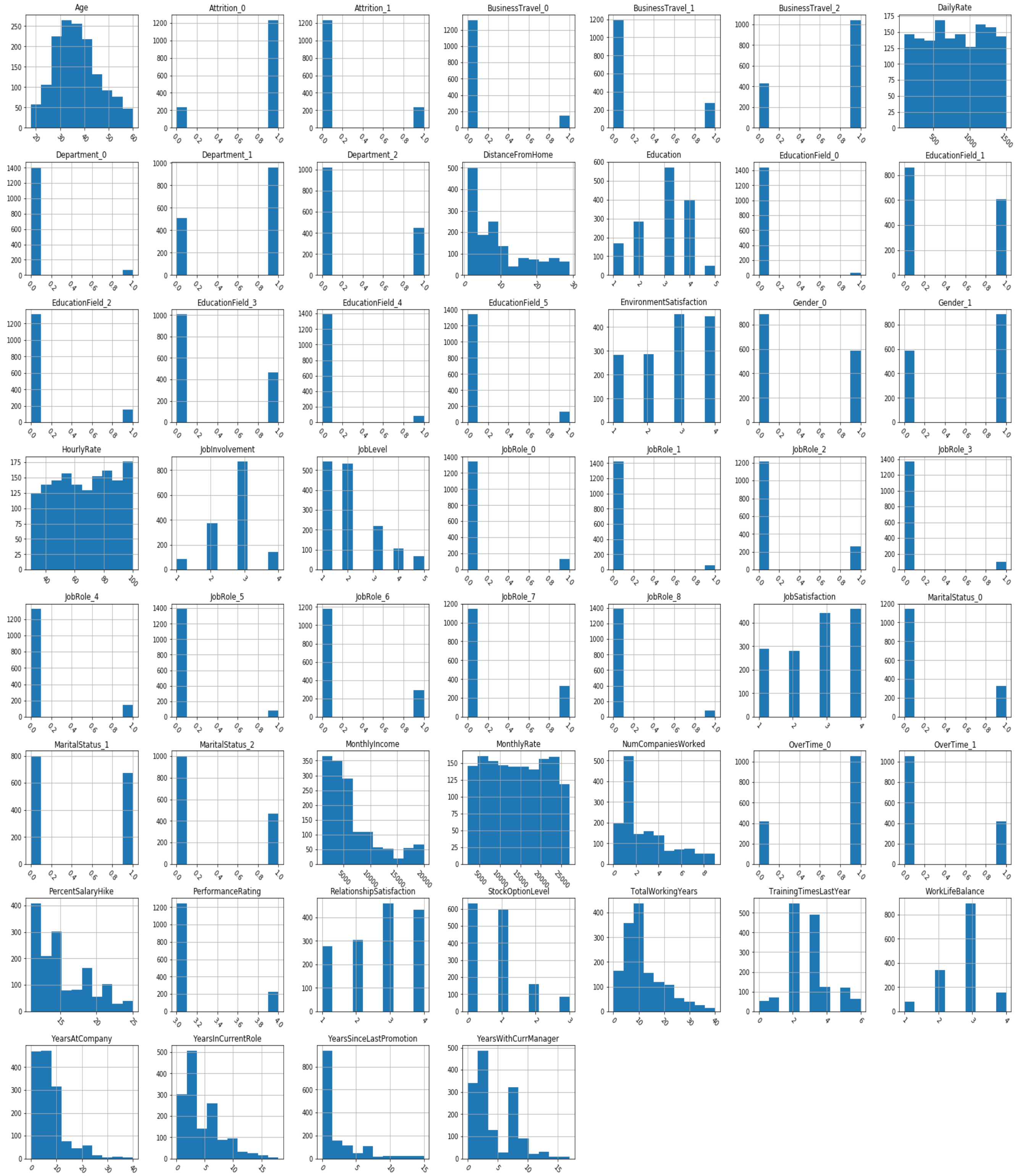
Few features which do not contain any useful information are discarded:

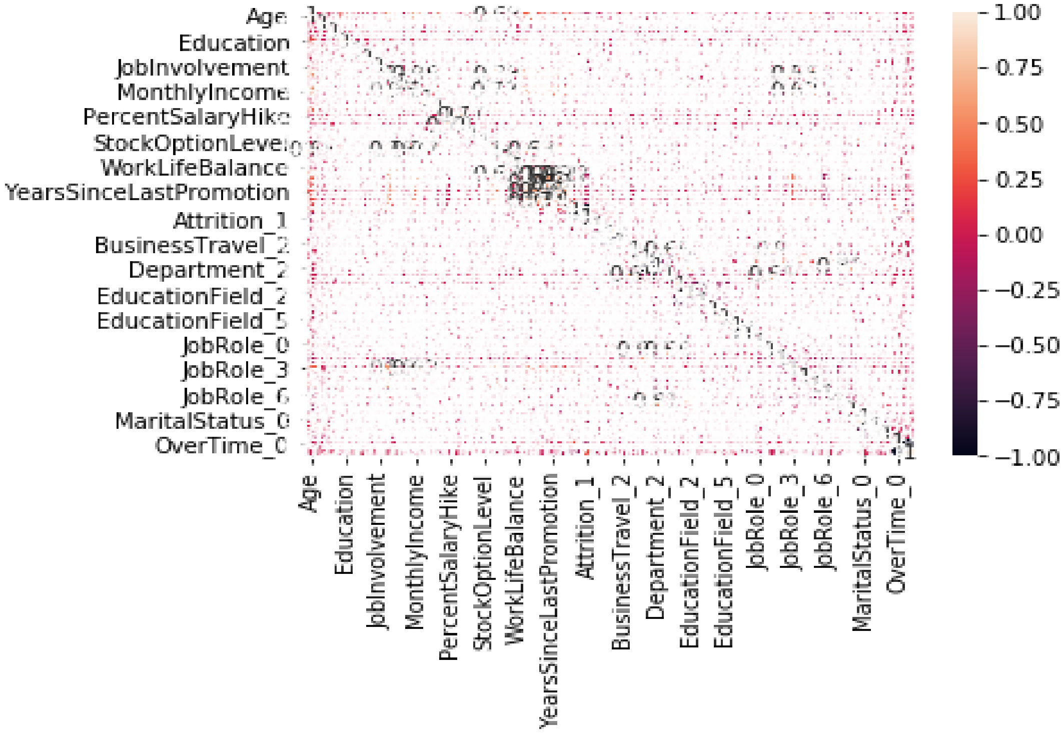
'EmployeeCount','StandardHours','Over18','EmployeeNumber'.

```
df=df.drop(['EmployeeCount','StandardHours','Over18','EmployeeNumber'],axis=1)
```

## 3c.)Data Visualisation

After data preprocessing, feature engineering and feature selection, the processed(ready to use) data can be visualized as:


Age
Attrition_0
Attrition_1
BusinessTravel_0
BusinessTravel_1
BusinessTravel_2
DailyRate
Department_0
Department_1
Department_2
DistanceFromHome
Education
EducationField_0
EducationField_1
EducationField_2
EducationField_3
EducationField_4
EducationField_5
EnvironmentSatisfaction
Gender_0
Gender_1
HourlyRate
JobInvolvement
JobLevel
JobRole_0
JobRole_1
JobRole_2
JobRole_3
JobRole_4
JobRole_5
JobRole_6
JobRole_7
JobRole_8
JobSatisfaction
MaritalStatus_0
MaritalStatus_1
MaritalStatus_2
MonthlyIncome
MonthlyRate
NumCompaniesWorked
OverTime_0
OverTime_1
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
1200
1400
20
30
40
50
60
500
1000
1500
5000
10000
15000
20000
25000



## 5. Software used:

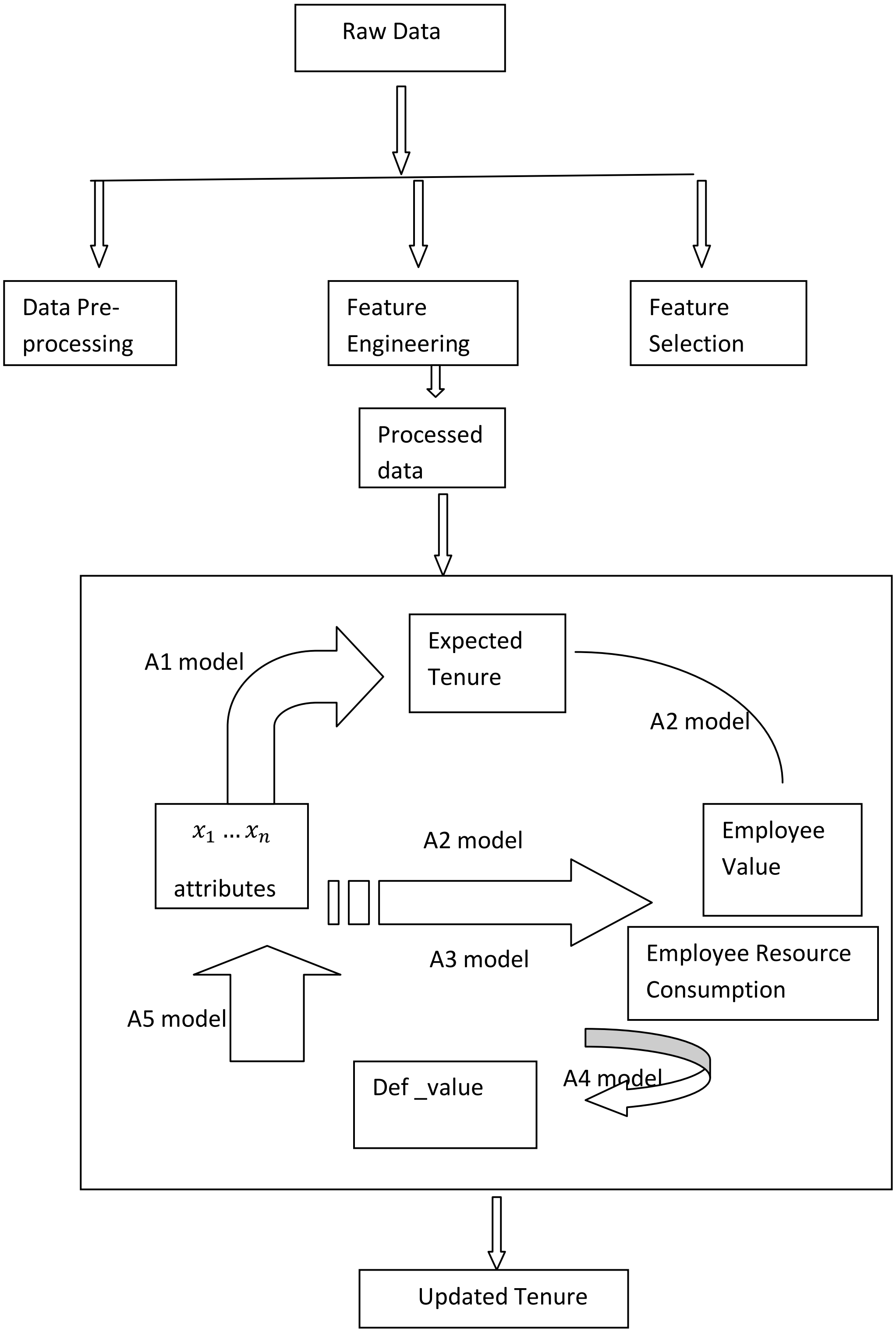
All the job of building the models by coding is being done in python notebook using python language codes on Google colaboratory.

Plotting graphs is also done in python notebook using python language codes on google colaboratory.

All statistical calculations like hypothesis testing calculation has been done in Excel.

(Note: The appendix section has all the links of the excel sheets and python notebooks which contains complete information about each and every process such as coding models ,graphs and statistical calculations. The idea to provide these links is to keep each and every process involved in the research work completely transparent and to ensure that all the items put in the research paper has been completely researched by own.)

## 6. Methodology



## *A. Human Resource Management Models*

### A1. Remaining Tenure Prediction Model

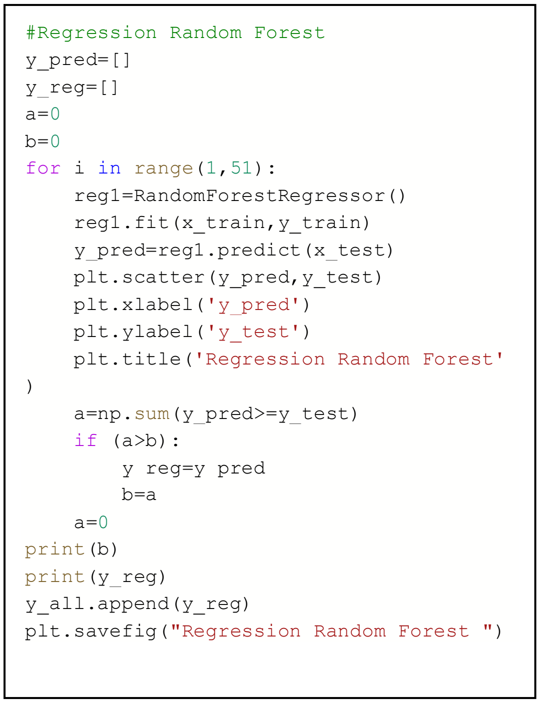
A predictive regression model is trained by ensemble learning by combining different regressive models (B1-B6) and generating the smallest mode range for the specific employee using the employee attrition data set(from above) and then the expected tenure of the employees currently working in the company is predicted using the regression model on the retention data set along with acting as the test condition for the model. As, the number of attrition-ed employees will always be less than currently working employees in a specific period of time therefore the test condition is that the Remaining predicted mode from the model for the currently working employees should be greater than 0 for the validation of the model.

*TenurePred($E_i$)>0*

Let us assume $x_1$,....,$x_n$ be the features of the employees from the data set, using which we are predicting the expected tenure(let us assume it to be $x_{n+1}$) and $E_i$ represents the $i^{th}$ employee.

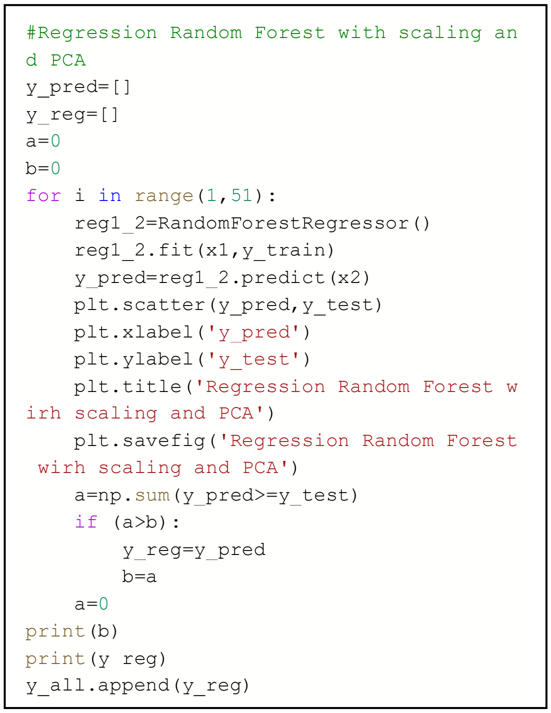
The machine learning based codes to develop the predictive model for employee tenure prediction and the visualization of all the models built for prediction are :

```
#Regression Random Forest
```

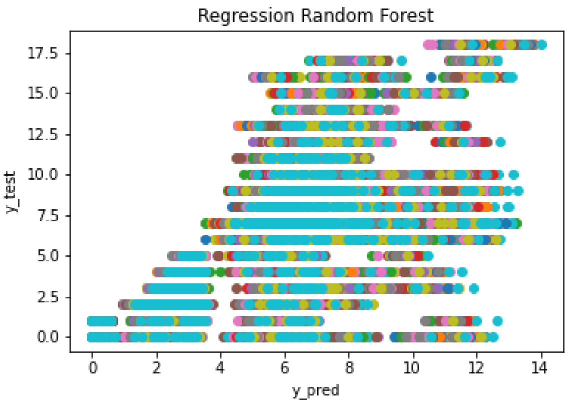


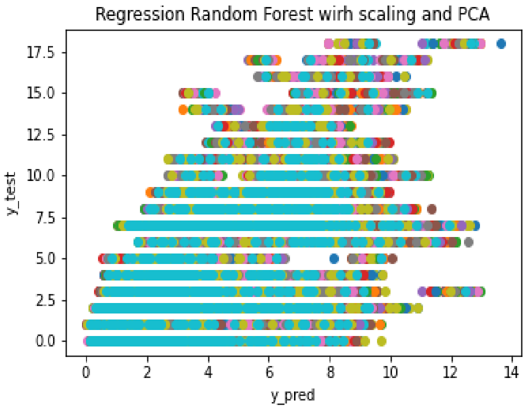
```
#Regression Random Forest
y_pred=[]
y_reg=[]
a=0
b=0
for i in range(1,51):
    reg1=RandomForestRegressor()
    reg1.fit(x_train,y_train)
    y_pred=reg1.predict(x_test)
    plt.scatter(y_pred,y_test)
    plt.xlabel('y_pred')
    plt.ylabel('y_test')
    plt.title('Regression Random Forest'
)
    a=np.sum(y_pred>=y_test)
    if (a>b):
        y reg=y pred
        b=a
    a=0
print(b)
print(y_reg)
y_all.append(y_reg)
plt.savefig("Regression Random Forest ")
```

```
#Regression Random Forest with scaling and PCA
```

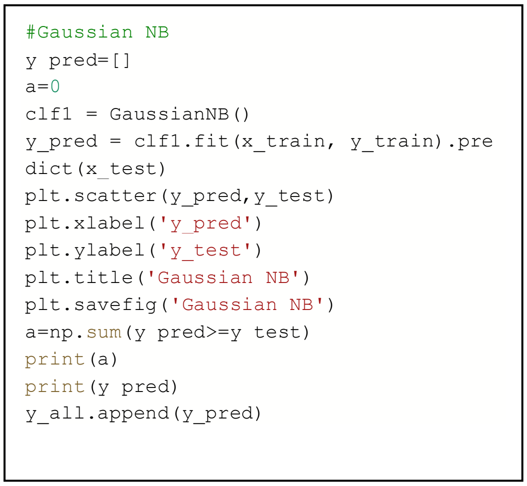


```
#Regression Random Forest with scaling an
d PCA
y_pred=[]
y_reg=[]
a=0
b=0
for i in range(1,51):
    reg1_2=RandomForestRegressor()
    reg1_2.fit(x1,y_train)
    y_pred=reg1_2.predict(x2)
    plt.scatter(y_pred,y_test)
    plt.xlabel('y_pred')
    plt.ylabel('y_test')
    plt.title('Regression Random Forest w
irh scaling and PCA')
    plt.savefig('Regression Random Forest
 wirh scaling and PCA')
    a=np.sum(y_pred>=y_test)
    if (a>b):
        y_reg=y_pred
        b=a
    a=0
print(b)
print(y reg)
y_all.append(y_reg)
```

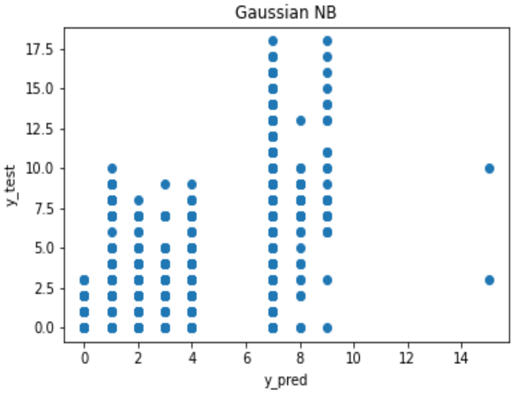




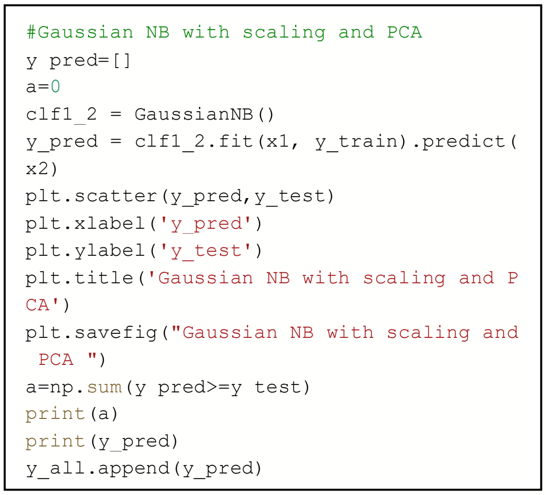
#Gaussian Naïve Bayes



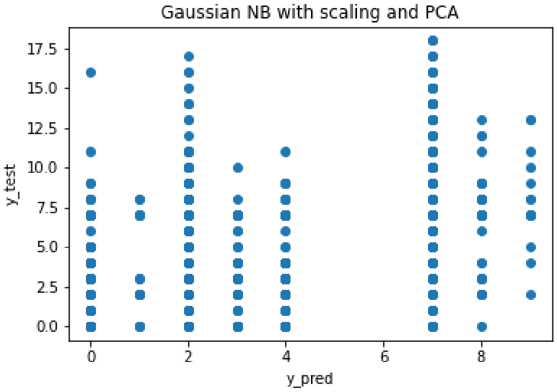
```
#Gaussian NB
y pred=[]
a=0
clf1 = GaussianNB()
y_pred = clf1.fit(x_train, y_train).pre
dict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Gaussian NB')
plt.savefig('Gaussian NB')
a=np.sum(y pred>=y test)
print(a)
print(y pred)
y_all.append(y_pred)
```



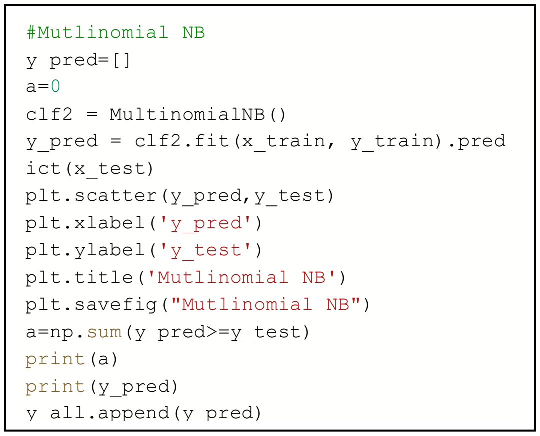
#Gaussian NB with scaling and PCA



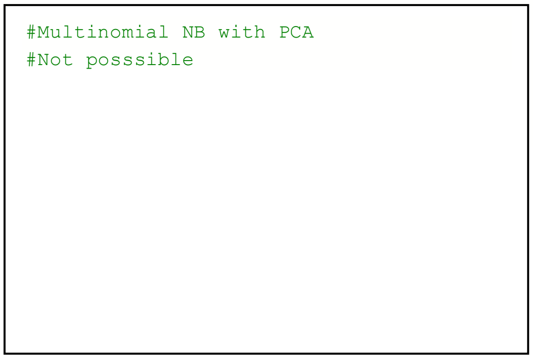
```
#Gaussian NB with scaling and PCA
y pred=[]
a=0
clf1_2 = GaussianNB()
y_pred = clf1_2.fit(x1, y_train).predict(
x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Gaussian NB with scaling and P
CA')
plt.savefig("Gaussian NB with scaling and
 PCA ")
a=np.sum(y pred>=y test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



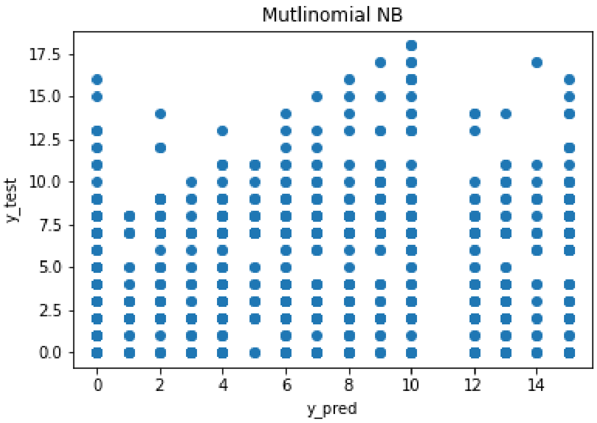
#Mutlinomial NB



```
#Mutlinomial NB
y pred=[]
a=0
clf2 = MultinomialNB()
y_pred = clf2.fit(x_train, y_train).pred
ict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Mutlinomial NB')
plt.savefig("Mutlinomial NB")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y all.append(y pred)
```

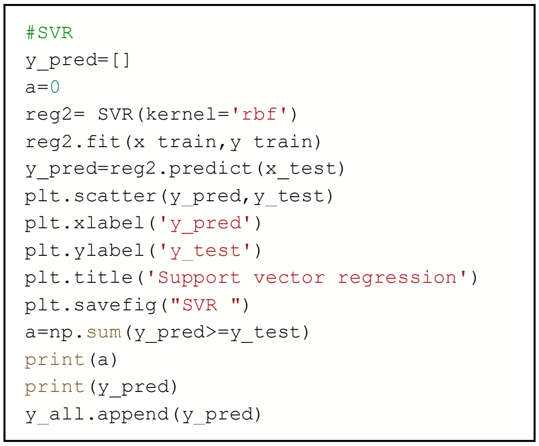


```
#Multinomial NB with PCA
#Not posssible
```

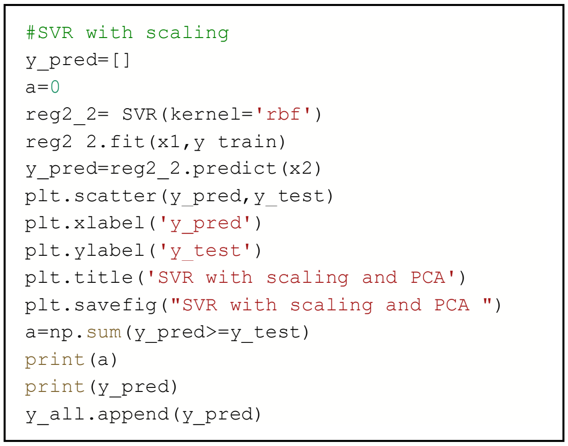


#Support Vector Regression

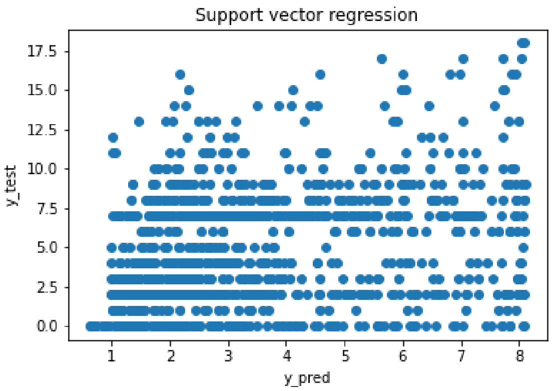
#SVR with scaling

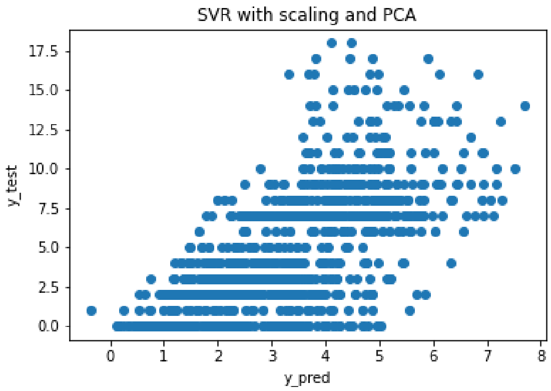


```
#SVR
y_pred=[]
a=0
reg2= SVR(kernel='rbf')
reg2.fit(x train,y train)
y_pred=reg2.predict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Support vector regression')
plt.savefig("SVR ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

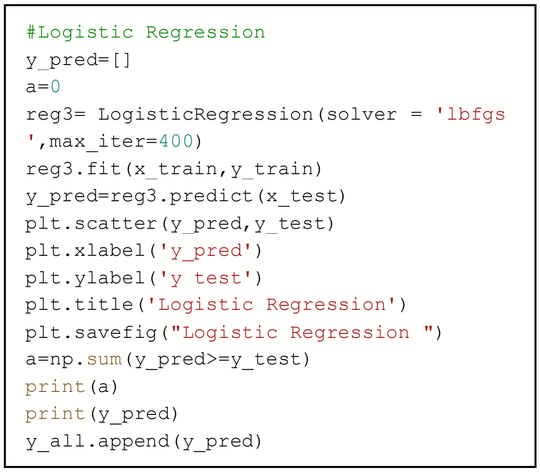


```
#SVR with scaling
y_pred=[]
a=0
reg2_2= SVR(kernel='rbf')
reg2 2.fit(x1,y train)
y_pred=reg2_2.predict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('SVR with scaling and PCA')
plt.savefig("SVR with scaling and PCA ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

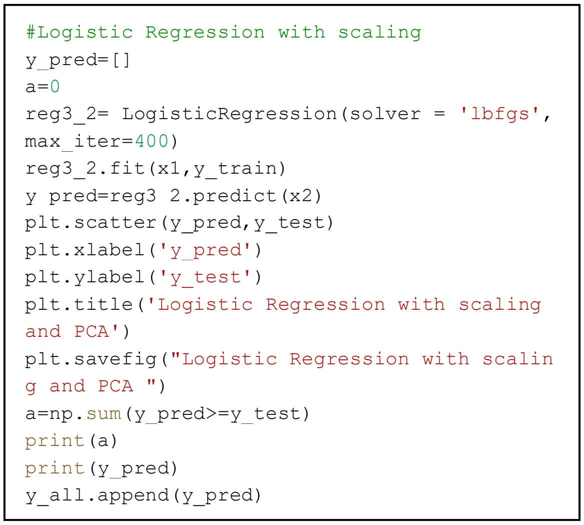




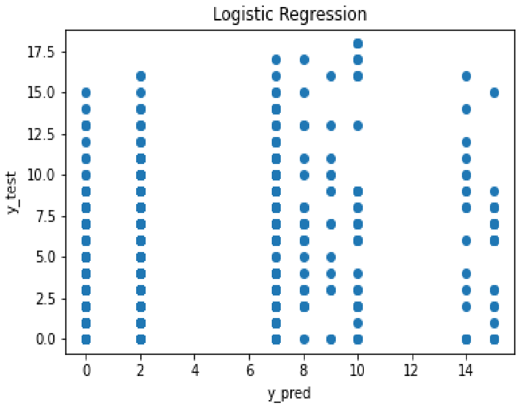
```
#Logistic Regression
```

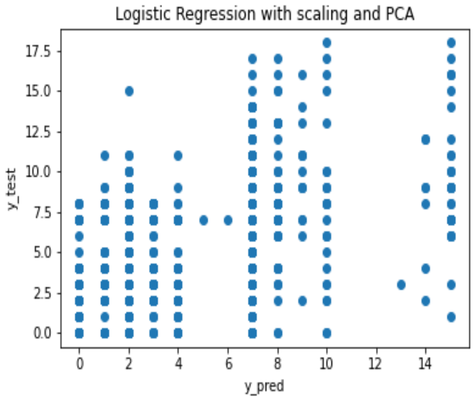


```
#Logistic Regression
y_pred=[]
a=0
reg3= LogisticRegression(solver = 'lbfgs
',max_iter=400)
reg3.fit(x_train,y_train)
y_pred=reg3.predict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y test')
plt.title('Logistic Regression')
plt.savefig("Logistic Regression ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



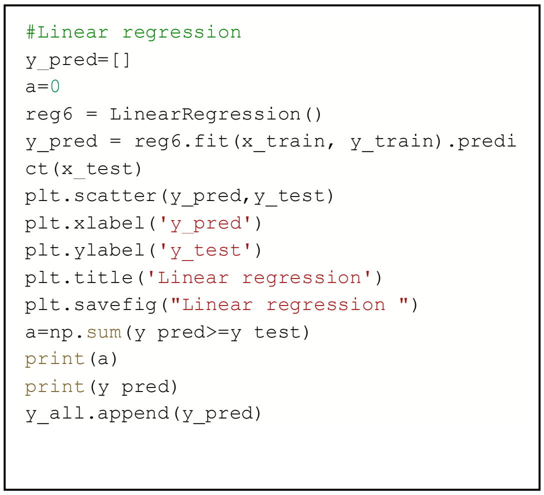
```
#Logistic Regression with scaling
y_pred=[]
a=0
reg3_2= LogisticRegression(solver = 'lbfgs',
max_iter=400)
reg3_2.fit(x1,y_train)
y pred=reg3 2.predict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Logistic Regression with scaling
and PCA')
plt.savefig("Logistic Regression with scalin
g and PCA ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



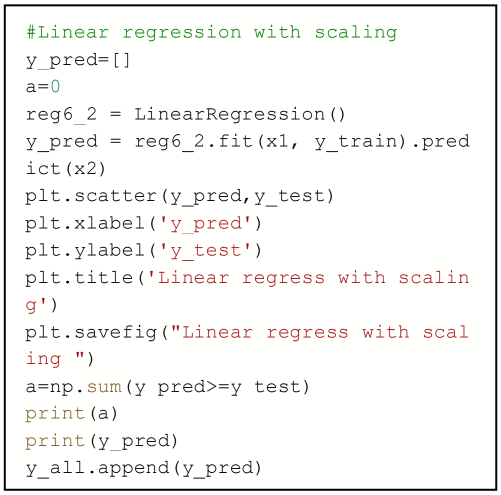


```
#Linear regression
```

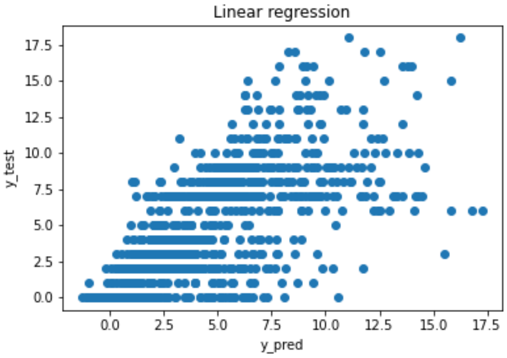
```
#Linear regression with scaling
```

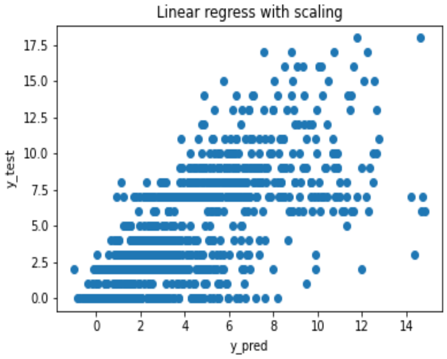


```
#Linear regression
y_pred=[]
a=0
reg6 = LinearRegression()
y_pred = reg6.fit(x_train, y_train).predi
ct(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Linear regression')
plt.savefig("Linear regression ")
a=np.sum(y pred>=y test)
print(a)
print(y pred)
y_all.append(y_pred)
```

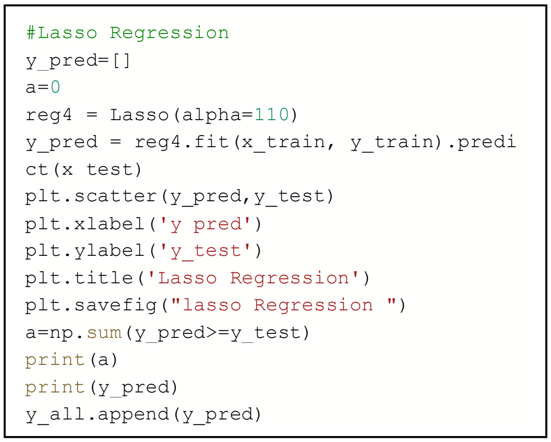


```
#Linear regression with scaling
y_pred=[]
a=0
reg6_2 = LinearRegression()
y_pred = reg6_2.fit(x1, y_train).pred
ict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Linear regress with scalin
g')
plt.savefig("Linear regress with scal
ing ")
a=np.sum(y pred>=y test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

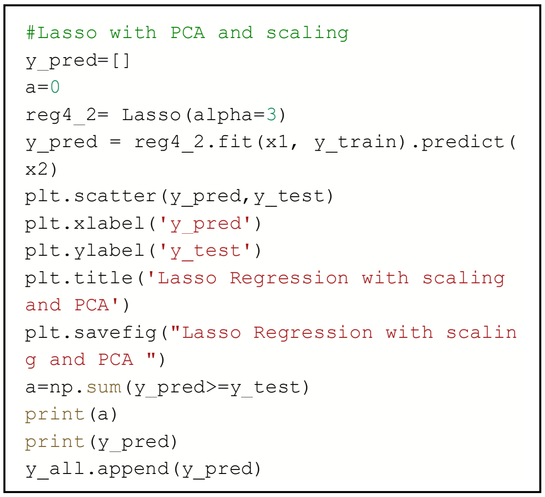




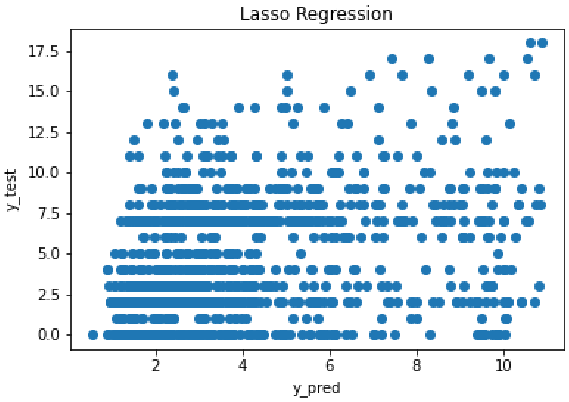
```
#Lasso Regression
```

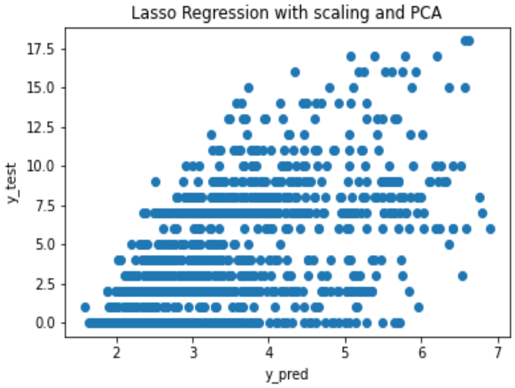


```
#Lasso Regression
y_pred=[]
a=0
reg4 = Lasso(alpha=110)
y_pred = reg4.fit(x_train, y_train).predi
ct(x test)
plt.scatter(y_pred,y_test)
plt.xlabel('y pred')
plt.ylabel('y_test')
plt.title('Lasso Regression')
plt.savefig("lasso Regression ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

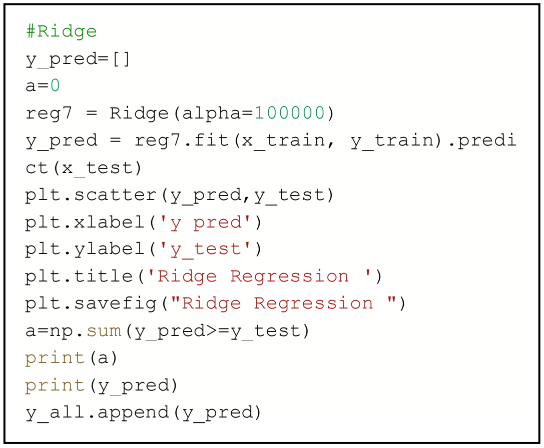


```
#Lasso with PCA and scaling
y_pred=[]
a=0
reg4_2= Lasso(alpha=3)
y_pred = reg4_2.fit(x1, y_train).predict(
x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Lasso Regression with scaling
and PCA')
plt.savefig("Lasso Regression with scalin
g and PCA ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```


Lasso Regression

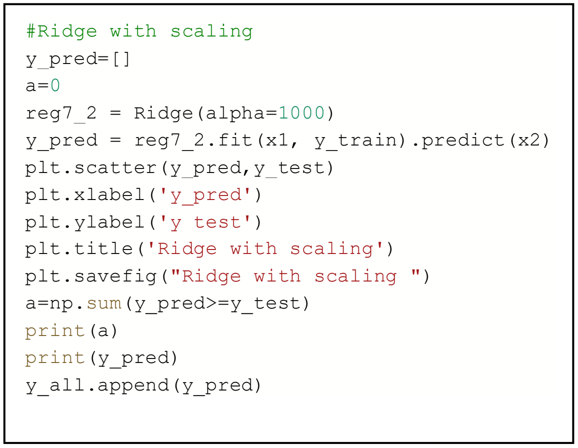

Lasso Regression with scaling and PCA

#Ridge Regression

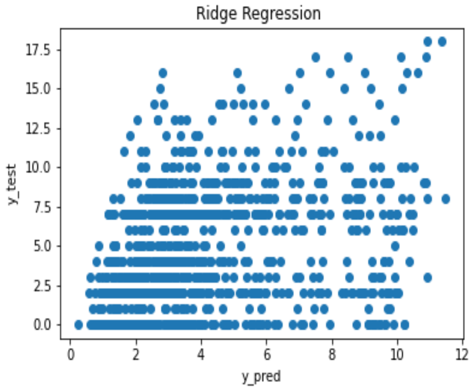


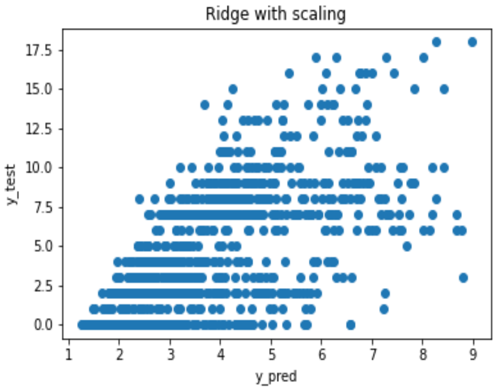
```
#Ridge
y_pred=[]
a=0
reg7 = Ridge(alpha=100000)
y_pred = reg7.fit(x_train, y_train).predi
ct(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y pred')
plt.ylabel('y_test')
plt.title('Ridge Regression ')
plt.savefig("Ridge Regression ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

#Ridge with scaling

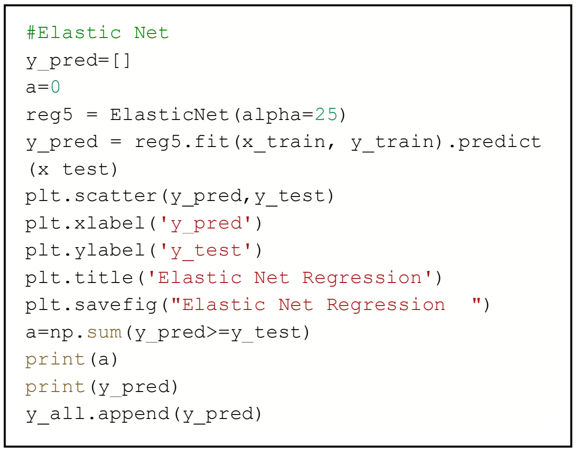


```
#Ridge with scaling
y_pred=[]
a=0
reg7_2 = Ridge(alpha=1000)
y_pred = reg7_2.fit(x1, y_train).predict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y test')
plt.title('Ridge with scaling')
plt.savefig("Ridge with scaling ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```


Ridge Regression

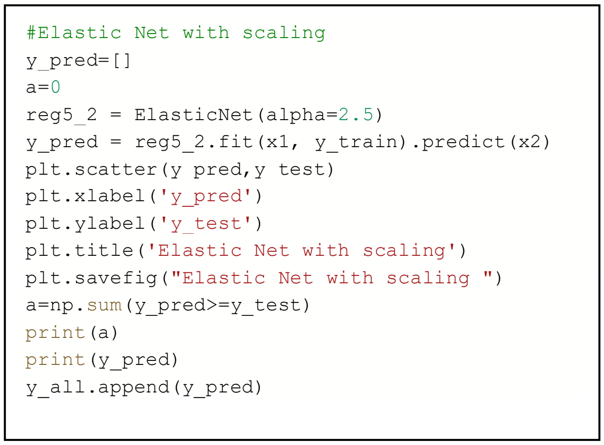

Ridge with scaling

#Elastic Net Regression

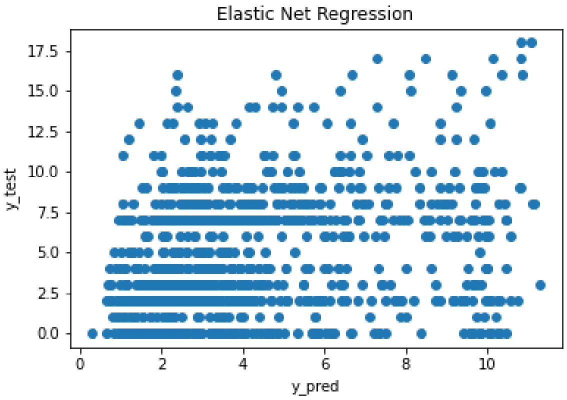


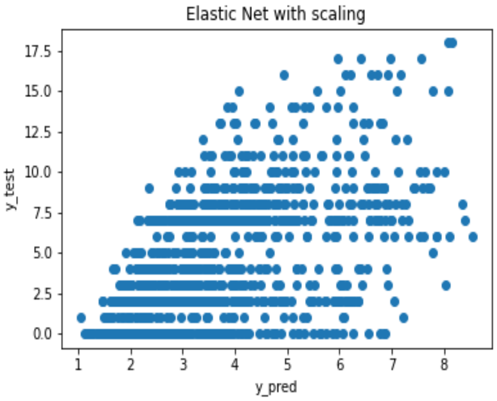
```
#Elastic Net
y_pred=[]
a=0
reg5 = ElasticNet(alpha=25)
y_pred = reg5.fit(x_train, y_train).predict
(x test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Elastic Net Regression')
plt.savefig("Elastic Net Regression  ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

#Elastic Net with scaling



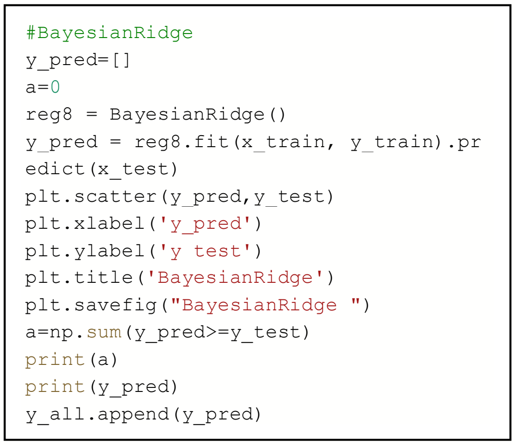
```
#Elastic Net with scaling
y_pred=[]
a=0
reg5_2 = ElasticNet(alpha=2.5)
y_pred = reg5_2.fit(x1, y_train).predict(x2)
plt.scatter(y pred,y test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('Elastic Net with scaling')
plt.savefig("Elastic Net with scaling ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



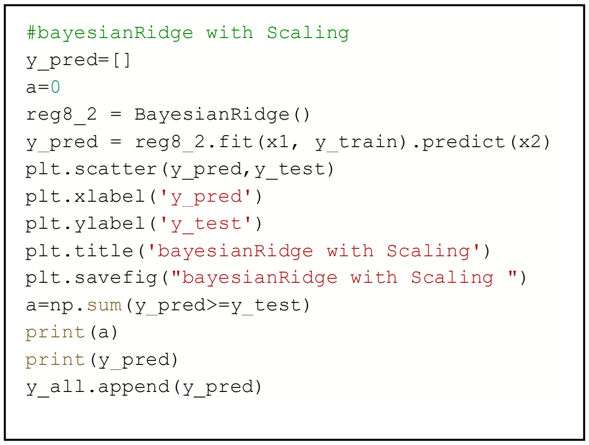


#BayesianRidge

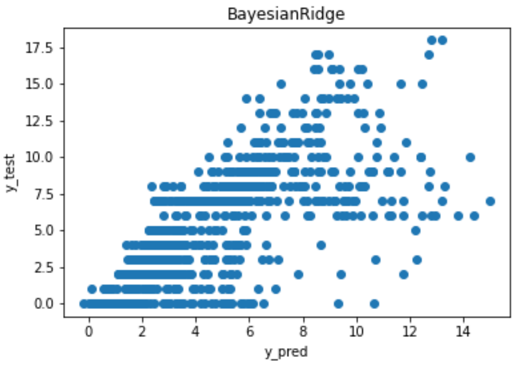
#bayesianRidge with Scaling

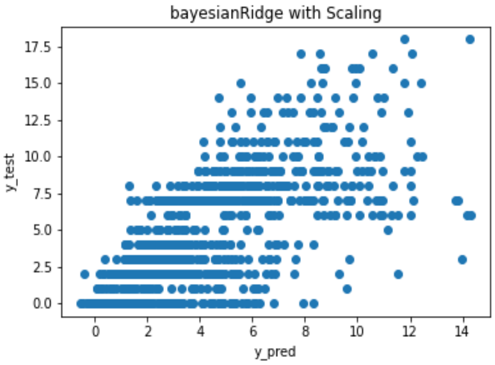


```
#BayesianRidge
y_pred=[]
a=0
reg8 = BayesianRidge()
y_pred = reg8.fit(x_train, y_train).pr
edict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y test')
plt.title('BayesianRidge')
plt.savefig("BayesianRidge ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



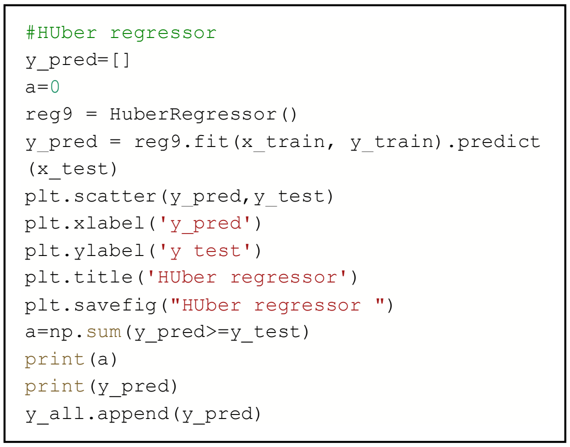
```
#bayesianRidge with Scaling
y_pred=[]
a=0
reg8_2 = BayesianRidge()
y_pred = reg8_2.fit(x1, y_train).predict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('bayesianRidge with Scaling')
plt.savefig("bayesianRidge with Scaling ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



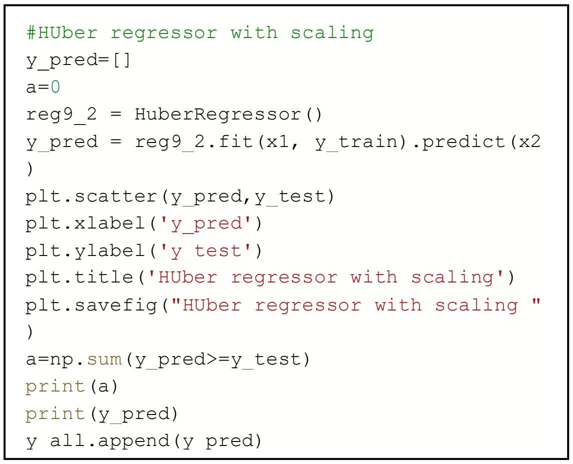


#HUber regressor

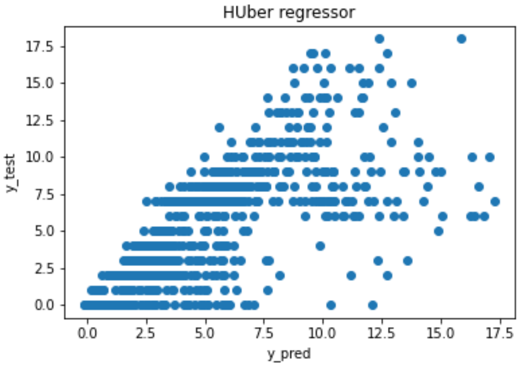
#HUber regressor with scaling

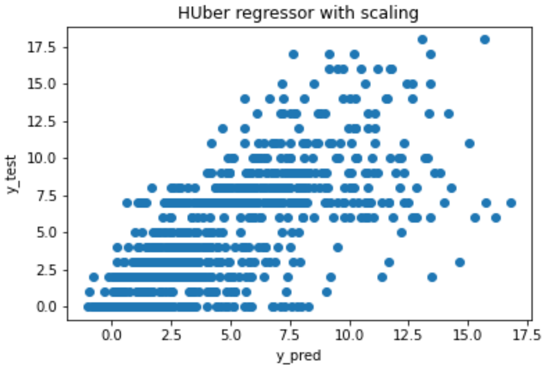


```
#HUber regressor
y_pred=[]
a=0
reg9 = HuberRegressor()
y_pred = reg9.fit(x_train, y_train).predict
(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y test')
plt.title('HUber regressor')
plt.savefig("HUber regressor ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

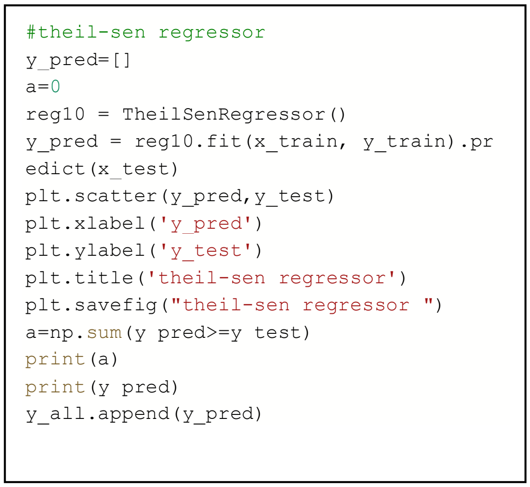


```
#HUber regressor with scaling
y_pred=[]
a=0
reg9_2 = HuberRegressor()
y_pred = reg9_2.fit(x1, y_train).predict(x2
)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y test')
plt.title('HUber regressor with scaling')
plt.savefig("HUber regressor with scaling "
)
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y all.append(y pred)
```



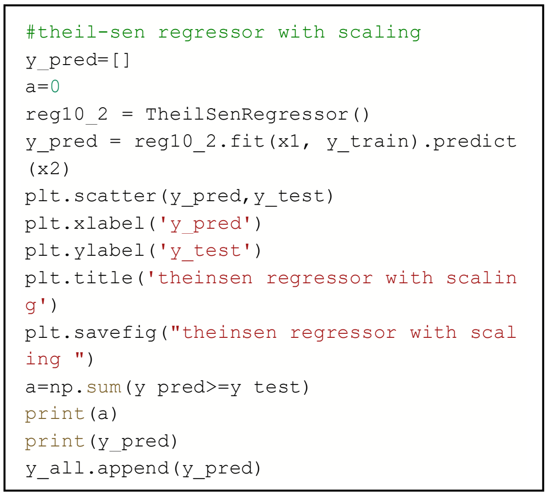


#theil-sen regressor

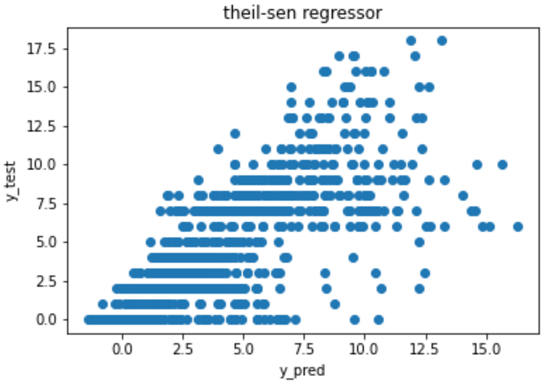


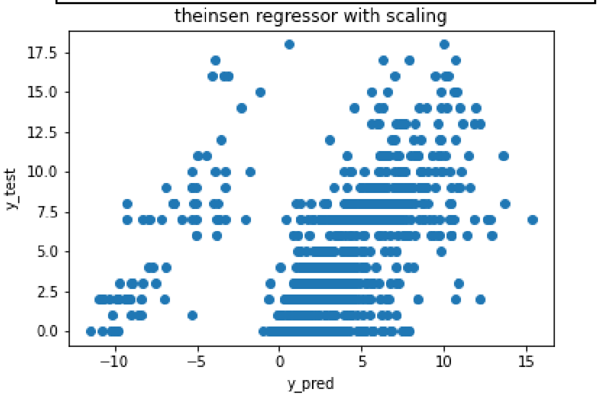
```
#theil-sen regressor
y_pred=[]
a=0
reg10 = TheilSenRegressor()
y_pred = reg10.fit(x_train, y_train).pr
edict(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('theil-sen regressor')
plt.savefig("theil-sen regressor ")
a=np.sum(y pred>=y test)
print(a)
print(y pred)
y_all.append(y_pred)
```

#theil-sen regressor with scaling

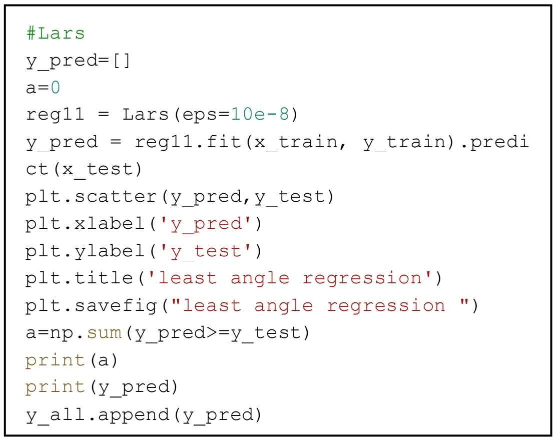


```
#theil-sen regressor with scaling
y_pred=[]
a=0
reg10_2 = TheilSenRegressor()
y_pred = reg10_2.fit(x1, y_train).predict
(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('theinsen regressor with scalin
g')
plt.savefig("theinsen regressor with scal
ing ")
a=np.sum(y pred>=y test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



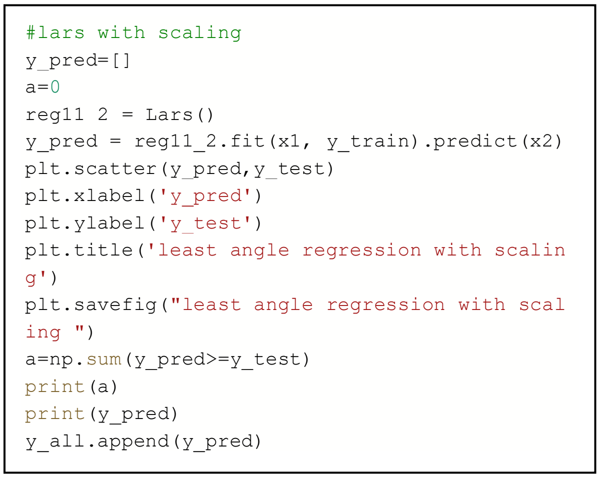


#Least angle regression

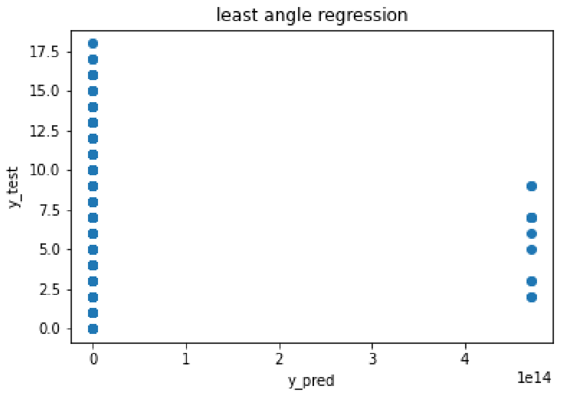


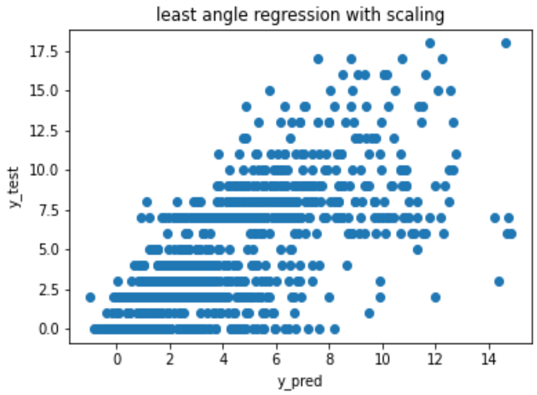
```
#Lars
y_pred=[]
a=0
reg11 = Lars(eps=10e-8)
y_pred = reg11.fit(x_train, y_train).predi
ct(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('least angle regression')
plt.savefig("least angle regression ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

#lars with scaling

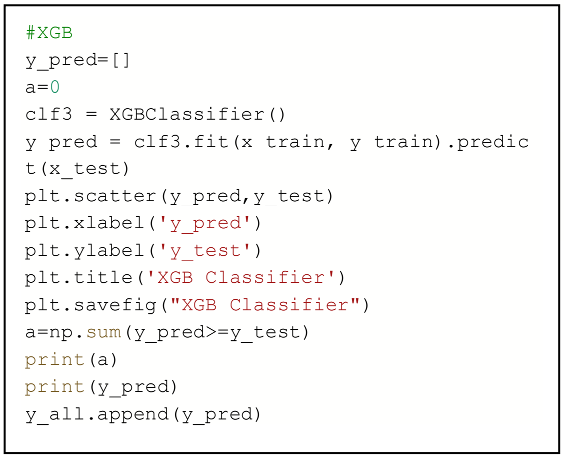


```
#lars with scaling
y_pred=[]
a=0
reg11 2 = Lars()
y_pred = reg11_2.fit(x1, y_train).predict(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('least angle regression with scalin
g')
plt.savefig("least angle regression with scal
ing ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```

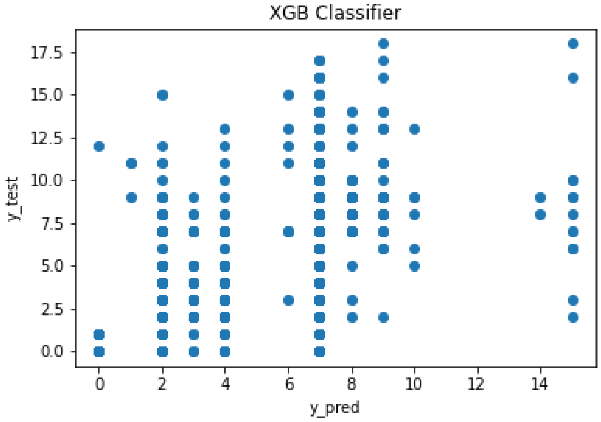




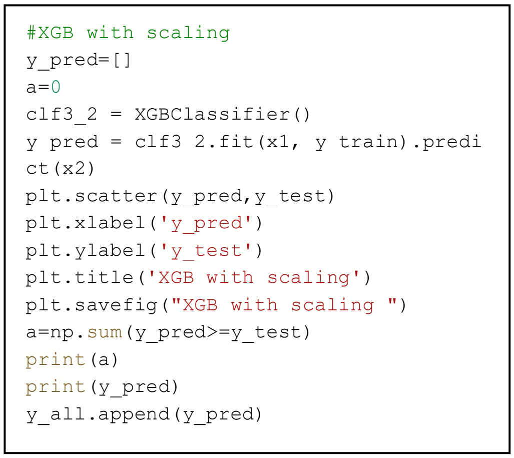
```
#XGB classifier
```



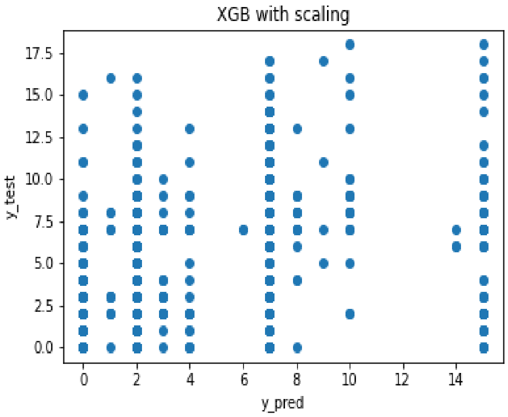
```
#XGB
y_pred=[]
a=0
clf3 = XGBClassifier()
y pred = clf3.fit(x train, y train).predic
t(x_test)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('XGB Classifier')
plt.savefig("XGB Classifier")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



```
#XGB with scaling
```

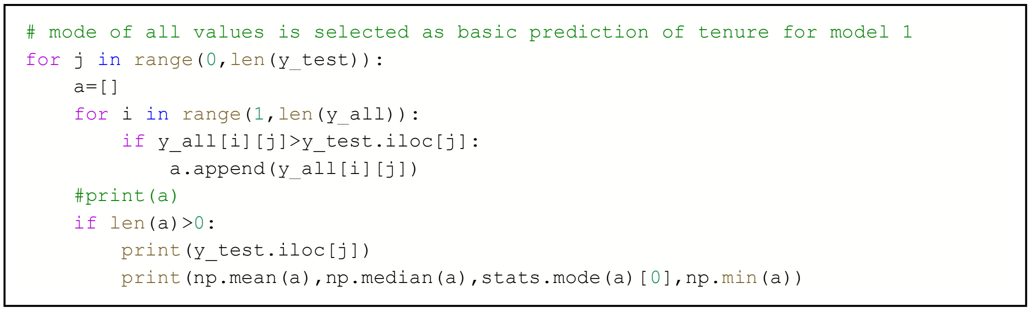


```
#XGB with scaling
y_pred=[]
a=0
clf3_2 = XGBClassifier()
y pred = clf3 2.fit(x1, y train).predi
ct(x2)
plt.scatter(y_pred,y_test)
plt.xlabel('y_pred')
plt.ylabel('y_test')
plt.title('XGB with scaling')
plt.savefig("XGB with scaling ")
a=np.sum(y_pred>=y_test)
print(a)
print(y_pred)
y_all.append(y_pred)
```



Then, using these machine learning models the final predictive model for employee tenure prediction is created and loaded. The detailed code for loading the model is in the ipython notebook(appendix).

Finally, mode of all values is selected as basic prediction for model 1.



```
# mode of all values is selected as basic prediction of tenure for model 1
for j in range(0,len(y_test)):
    a=[]
    for i in range(1,len(y_all)):
        if y_all[i][j]>y_test.iloc[j]:
            a.append(y_all[i][j])
    #print(a)
    if len(a)>0:
        print(y_test.iloc[j])
        print(np.mean(a),np.median(a),stats.mode(a)[0],np.min(a))
```

```
#either mode is considered if it exists or deafult value of tenure is accepted.
tenure_model1=[]
for j in range(0,len(y_test)):
    a=[]
    for i in range(1,len(y_all)):
        if y_all[i][j]>y_test.iloc[j]:
            a.append(y_all[i][j])
    #print(a)
    if len(a)>0:
        tenure_model1.append(stats.mode(a)[0][0])
    else:
        tenure_model1.append(y_test.iloc[j])
print(tenure_model1)
```

## A2. Employee Value Prediction Model

The employees are clustered into many clusters using the K Means algorithm, and each cluster is assigned a range mid-point of class size 5 denoting the value of the employee to the company and therefore defining the value of k to be 20.

For clustering, the value is generated considering the weighted n+ 1 features:

where $w_1$,......,$w_n$ , $w_{n+1}$ are the weights corresponding to each of the n+1 features $x_1$,....,$x_n$ , $x_{n+1}$ respectively and the n attributes($x_1$,.....,$x_n$) are from the attrition-ed employees data set and the predicted tenure forms the $(n+1)^{th}$ attribute used for the value-generationn in the employee value prediction model.

The equation for calculating the employee value is :

$$\begin{aligned}
&y_{employee\ value} = w_1Age + w_2DailyRate + w_3DistanceFromHome + w_4Education + \\
&w_5EnvironmentSatisfaction + w_6HourlyRate + w_7JobInvolvement + w_8JobLevel + \\
&w_9JobSatisfaction + w_{10}MonthlyIncome + w_{11}MonthlyRate + w_{12}NumCompaniesWorked + \\
&w_{13}PercentSalaryHike + w_{14}PerformanceRating + w_{15}RelationshipSatisfaction + \\
&w_{16}StockOptionLevel + w_{17}TotalWorkingYears + w_{18}TrainingTimesLastYear + \\
&w_{19}WorkLifeBalance + w_{20}YearsAtCompany + w_{21}YearsSinceLastPromotion + \\
&w_{22}YearsWithCurrManager + w_{23}BusinessTravel_0 + w_{24}BusinessTravel_1 + \\
&w_{25}BusinessTravel_2 + w_{26}Department_0 + w_{27}Department_1 + w_{28}Department_2 + \\
&w_{29}EducationField_0 + w_{30}EducationField_1 + w_{31}EducationField_2 + w_{32}EducationField_3 + \\
&w_{33}EducationField_4 + w_{34}EducationField_5 + w_{35}Gender_0 + w_{36}Gender_1 + w_{37}JobRole_0 + \\
&w_{38}JobRole_1 + w_{39}JobRole_2 + w_{40}JobRole_3 + w_{41}JobRole_4 + w_{42}JobRole_5 + w_{43}JobRole_6 + \\
&w_{44}JobRole_7 + w_{45}JobRole_8 + w_{46}MaritalStatus_0 + w_{47}MaritalStatus_1 + w_{48}MaritalStatus_2 + \\
&w_{49}OverTime\_0 + w_{50}OverTime\_1 + w_{51}Expected\_tenure
\end{aligned}$$

Where the vector $[w_1, w_2 \dots w_{51}]$ is [0,-1,0,1,0,-1,1,1,0,-1,-1,1,-1,1,0,1,1,1,0,1,1,0,0,2,1,0,0,0,0,0,0,0,0,0,0,0,1,3,2,3,5,5,4,2,1,0,0,0,-1,1,1]

The code to develop the Employee Value Prediction model is:

```
#adding generated tenure to data points
x_test_new = x_test.copy()
x_test_new['Expected_tenure']=tenure_model1
print(x_test.shape)
print(x_test_new.shape)
```

```
#clustering the employees
kmeans=KMeans(n_clusters=20).fit(x_test_new)
cluster=kmeans.cluster centers
cluster.shape
```

```
#employee value calculation model
#some attributes have positive effects with increasing value and some negative
# +1 means positive, 0 means no effect,-1 negative
print(x_test_new.columns)
attr=np.array([0,-1,0,1,0,-1,1,1,0,-1,-1,1,-
1,1,0,1,1,1,0,1,1,0,0,2,1,0,0,0,0,0,0,0,0,0,0,0,1,3,2,3,5,5,4,2,1,0,0,0,-1,1,1])
print(len(attr))
def value_predictor(x):
    return x.dot(np.transpose(attr))
emp_cluster_score=value_predictor(cluster)
alpha=np.min(emp_cluster_score)
beta=np.max(emp_cluster_score)
emp_cluster_imp_perc=(emp_cluster_score-alpha)*(1/(beta-alpha))
emp_cluster_imp_perc
```

## A3. Resource consumption Model

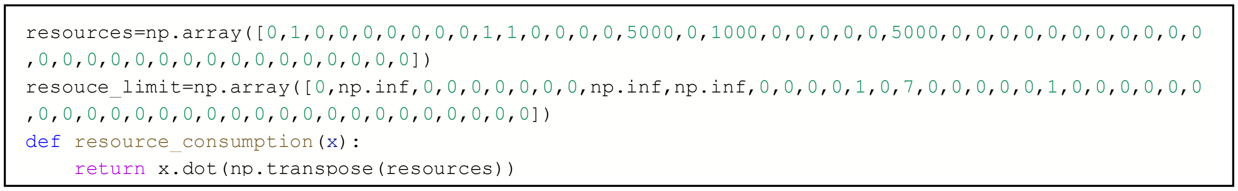
This model evaluates a function $A3_i$ depending on all the attributes of an employee $x_1 … x_n$ in a weighted manner representing the resources being used according to the current attributes.

The equation for calculating the Resource consumption is:

$$\begin{aligned} y_{employee\ value} = {} & w_1 Age + w_2 DailyRate + w_3 DistanceFromHome + w_4 Education \\ & + w_5 EnvironmentSatisfaction + w_6 HourlyRate + w_7 JobInvolvement \\ & + w_8 JobLevel + w_9 JobSatisfaction + w_{10} MonthlyIncome + w_{11} MonthlyRate \\ & + w_{12} NumCompaniesWorked + w_{13} PercentSalaryHike \\ & + w_{14} PerformanceRating + w_{15} RelationshipSatisfaction \\ & + w_{16} StockOptionLevel + w_{17} TotalWorkingYears + w_{18} TrainingTimesLastYear \\ & + w_{19} WorkLifeBalance + w_{20} YearsAtCompany + w_{21} YearsSinceLastPromotion \\ & + w_{22} YearsWithCurrManager + w_{23} BusinessTravel_0 + w_{24} BusinessTravel_1 \\ & + w_{25} BusinessTravel_2 + w_{26} Department_0 + w_{27} Department_1 \\ & + w_{28} Department_2 + w_{29} EducationField_0 + w_{30} EducationField_1 \\ & + w_{31} EducationField_2 + w_{32} EducationField_3 + w_{33} EducationField_4 \\ & + w_{34} EducationField_5 + w_{35} Gender_0 + w_{36} Gender_1 + w_{37} JobRole_0 \\ & + w_{38} JobRole_1 + w_{39} JobRole_2 + w_{40} JobRole_3 + w_{41} JobRole_4 + w_{42} JobRole_5 \\ & + w_{43} JobRole_6 + w_{44} JobRole_7 + w_{45} JobRole_8 + w_{46} MaritalStatus_0 \\ & + w_{47} MaritalStatus_1 + w_{48} MaritalStatus_2 + w_{49} OverTime\_0 + w_{50} OverTime\_1 \\ & + w_{51} Expected\_tenure \end{aligned}$$

Where the vector $[w_1, w_2 \dots w_{51}]$ is
[0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,5000,0,1000,0,0,0,0,0,5000,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

The code to develop the Resource consumption Model is:



```
resources=np.array([0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,5000,0,1000,0,0,0,0,0,5000,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
resouce_limit=np.array([0,np.inf,0,0,0,0,0,0,0,np.inf,np.inf,0,0,0,0,1,0,7,0,0,0,0,0,1,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
def resource_consumption(x):
    return x.dot(np.transpose(resources))
```

## A4. Retention Technique Selection Model

The above calculated value is considered and for the employees with more than 50 percent importance value will be considered for this model. This model in turn carries various Retention Techniques which are unique for each $x_i$, updates the features $x$ and re-evaluates the cluster in the model $A2$ for $A2_i$ generating a different importance value. Also the function $A3_i$ has to be minimized while updating and recurring for the suitable solution. This continues till all the combination of the techniques is exhausted and generates the best combination of technique possible for the each employee $A2_i$ being considered along with the most optimal Resource possible. Therefore the argument is to $\max\,(\frac{A2_i}{A3_i})$ where the $Res_i$ represents the Resource consumption of an employee $i$. This can be written as:-

$$\max\,(\frac{A2_i}{A3_i}) = \max\,(\frac{A2_i | \, x_1 \dots x_{n+1}}{A3_i \mid \, x_1 \dots . x_n})$$

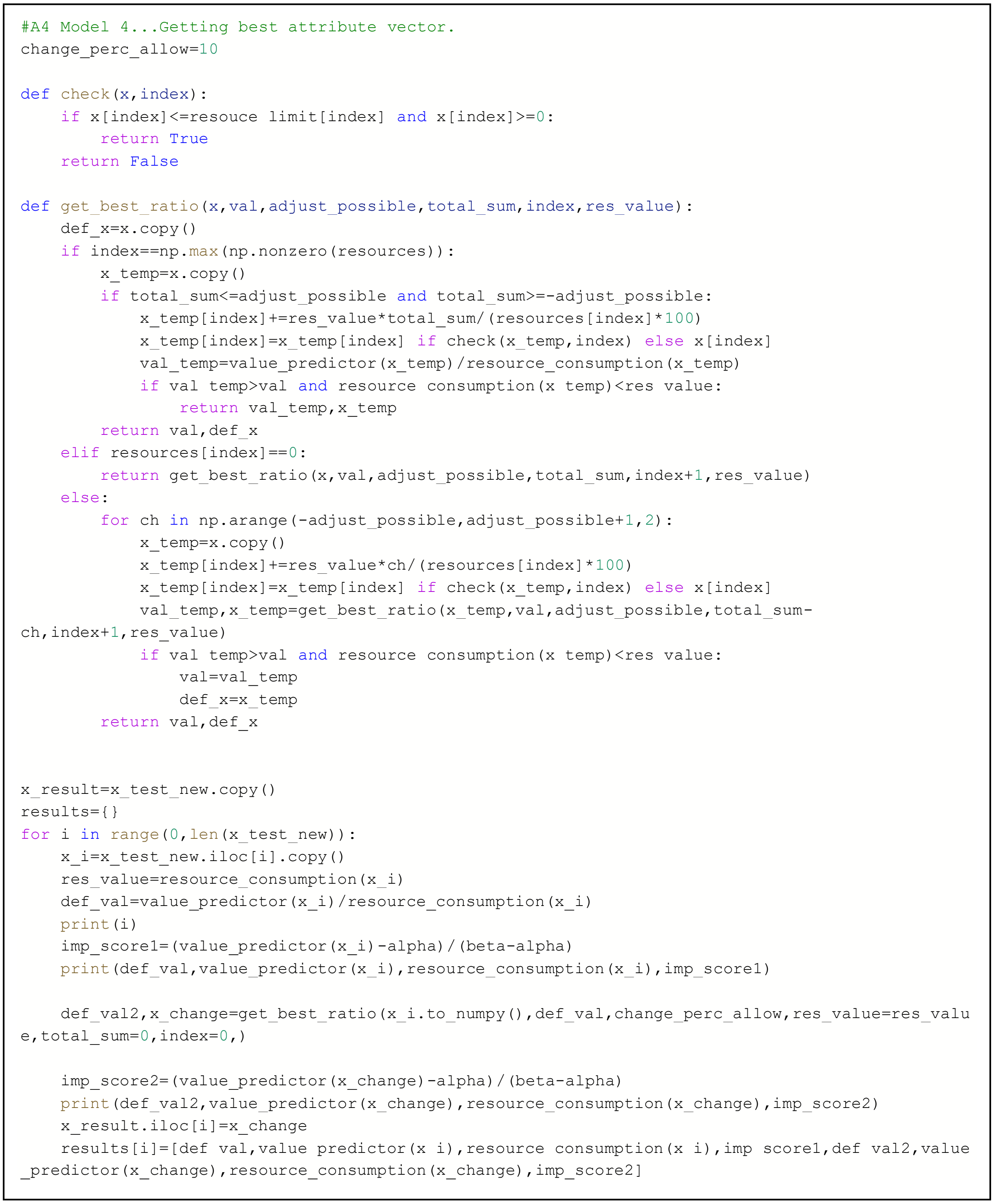
$$= \max\,(\frac{(A2_i \mid \, x_1 \, \dots . x_{n+1}\,) * (x_{n+1} \mid x_1 \dots . x_n)}{(A3_i \mid \, x_1 \dots . x_n)})$$

$$= (x_{n+1} \mid \, (x_1 \, \dots . \, x_n)\,)\, * \, \max\,(\frac{A2_i | \, x_1 \dots x_n}{A3_i \mid \, x_1 \dots . \, x_n})$$

$x_{n+1}$ represents the predicted tenure which is considered to be constant for the whole iterative process considered and therefore can be taken out.

$$= \,(A1)\, * \, max(\frac{A2_i}{A3_i})$$

The code to develop the Retention Technique Selection Model is writtten below .The output of this code gives the best attribute vector consisting of employee tenure, value, resource consumption and importance score obtained before and after updating the optimal scores for features.



```
#A4 Model 4...Getting best attribute vector.
change_perc_allow=10

def check(x,index):
    if x[index]<=resouce limit[index] and x[index]>=0:
        return True
    return False

def get_best_ratio(x,val,adjust_possible,total_sum,index,res_value):
    def_x=x.copy()
    if index==np.max(np.nonzero(resources)):
        x_temp=x.copy()
        if total_sum<=adjust_possible and total_sum>=-adjust_possible:
            x_temp[index]+=res_value*total_sum/(resources[index]*100)
            x_temp[index]=x_temp[index] if check(x_temp,index) else x[index]
            val_temp=value_predictor(x_temp)/resource_consumption(x_temp)
            if val temp>val and resource consumption(x temp)<res value:
                return val_temp,x_temp
        return val,def_x
    elif resources[index]==0:
        return get_best_ratio(x,val,adjust_possible,total_sum,index+1,res_value)
    else:
        for ch in np.arange(-adjust_possible,adjust_possible+1,2):
            x_temp=x.copy()
            x_temp[index]+=res_value*ch/(resources[index]*100)
            x_temp[index]=x_temp[index] if check(x_temp,index) else x[index]
            val_temp,x_temp=get_best_ratio(x_temp,val,adjust_possible,total_sum-
ch,index+1,res_value)
            if val temp>val and resource consumption(x temp)<res value:
                val=val_temp
                def_x=x_temp
        return val,def_x


x_result=x_test_new.copy()
results={}
for i in range(0,len(x_test_new)):
    x_i=x_test_new.iloc[i].copy()
    res_value=resource_consumption(x_i)
    def_val=value_predictor(x_i)/resource_consumption(x_i)
    print(i)
    imp_score1=(value_predictor(x_i)-alpha)/(beta-alpha)
    print(def_val,value_predictor(x_i),resource_consumption(x_i),imp_score1)

    def_val2,x_change=get_best_ratio(x_i.to_numpy(),def_val,change_perc_allow,res_value=res_valu
e,total_sum=0,index=0,)

    imp_score2=(value_predictor(x_change)-alpha)/(beta-alpha)
    print(def_val2,value_predictor(x_change),resource_consumption(x_change),imp_score2)
    x_result.iloc[i]=x_change
    results[i]=[def val,value predictor(x i),resource consumption(x i),imp score1,def val2,value
_predictor(x_change),resource_consumption(x_change),imp_score2]
```

## A5. Updated Tenure Prediction Model

After calculating the optimal solution for the attributes of the employee we generate the updated predicted tenure according to our Model A1 i.e. $x_0$.Now, the difference between $x_0$ - $x_{n+1}$ is calculated and represents the Final Result for our Algorithm along with the Resource consumption value and the updated attributes which can be used to reflect the actual retention strategies used for each employee.

The code to develop Updated Prediction Model is:

```
#Model 5 and analysis.
new tenure=[]
x_results=x_result.drop(['Expected_tenure'],axis=1)
new_tenure.append(reg1.predict(x_results))
new_tenure.append(reg2.predict(x_results))
new_tenure.append(reg3.predict(x_results))
new_tenure.append(reg4.predict(x_results))
new_tenure.append(reg5.predict(x_results))
new_tenure.append(reg6.predict(x_results))
new_tenure.append(reg7.predict(x_results))
new_tenure.append(reg8.predict(x_results))
new_tenure.append(reg9.predict(x_results))
new_tenure.append(reg10.predict(x_results))
new tenure.append(reg11.predict(x results))
new_tenure.append(clf1.predict(x_results))
new_tenure.append(clf2.predict(x_results))
new_tenure.append(clf3.predict(x_results))
X_result=torch.DoubleTensor(x_results.values)
new_tenure.append(model(X_result).detach().numpy()[:,0])
new_tenure.append(model2(X_result).detach().numpy()[:,0])
len(new_tenure)
```

## *B. Machine Learning Algorithms*

### B1. Support Vector Regression [7]

As a supervised-learning approach, SVR trains using a symmetrical loss function, which equally penalizes high and low misestimates. One of the main advantages of SVR is that its computational complexity does not depend on the dimensionality of the input space.

More specifically, SVR is formulated as an optimization problem by first defining a convex $\varepsilon$-insensitive loss function to be minimized and finding the flattest tube that contains most of the training instances. Hence, a multiobjective function is constructed from the loss function and the geometrical properties of the tube. Then, the convex optimization, which has a unique solution, is solved, using appropriate numerical optimization algorithms.

In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM in incorporated. This constrained quadratic optimization problem can be solved by finding the Lagrangian (see Equation 4-8). The Lagrange multipliers, or dual variables, are $\lambda, \lambda^*, \alpha, \alpha^*$ and are nonnegative real numbers.

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i + \xi_i^*, \qquad (4\text{-}7)$$

Subject to

$$\begin{aligned} y_i - w^T x_i \le \varepsilon + \xi_i^* \quad & i = 1 \ldots N \\ w^T x_i - y_i \le \varepsilon + \xi_i \quad & i = 1 \ldots N \\ \xi_i, \xi_i^* \ge 0 \quad & i = 1 \ldots N \end{aligned}$$

$$\begin{aligned} \mathcal{L}(w, \xi^*, \xi, \lambda, \lambda^*, \alpha, \alpha^*) = {} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i + \xi_i^* + \sum_{i=1}^{N} \alpha_i^* \left( y_i - w^T x_i - \varepsilon - \xi_i^* \right) \\ & + \sum_{i=1}^{N} \alpha_i \left( -y_i + w^T x_i - \varepsilon - \xi_i \right) - \sum_{i=1}^{N} \lambda_i \xi_i + \lambda_i^* \xi_i^* \end{aligned} \qquad (4\text{-}8)$$

The minimum of Equation 4-8 is found by taking its partial derivatives with respect to the variables and setting them equal to zero, based on the *Karush-Kuhn-Tucker* (KKT) conditions.

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta w} &= w - \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) x_i = 0 \\ \frac{\delta \mathcal{L}}{\delta \xi_i^*} &= C - \lambda_i^* - \alpha_i^* = 0 \\ \frac{\delta \mathcal{L}}{\delta \xi_i} &= C - \lambda_i - \alpha_i = 0 \\ \frac{\delta \mathcal{L}}{\delta \lambda_i^*} &= \sum_{i=1}^{N} \xi_i^* \le 0 \\ \frac{\delta \mathcal{L}}{\delta \lambda_i} &= \sum_{i=1}^{N} \xi_i \le 0 \\ \frac{\delta \mathcal{L}}{\delta \alpha_i^*} &= y_i - w^T x_i - \varepsilon - \xi_i^* \le 0 \\ \frac{\delta \mathcal{L}}{\delta \alpha_i} &= -y_i + w^T x_i - \varepsilon - \xi_i \le 0 \end{aligned} \qquad (4\text{-}9)$$

$$\left. \begin{aligned} \alpha_i \left( -y_i + w^T x_i - \varepsilon - \xi_i \right) &= 0 \\ \alpha_i^* \left( y_i - w^T x_i - \varepsilon - \xi_i^* \right) &= 0 \\ \lambda_i \xi_i &= 0, \\ \lambda_i^* \xi_i^* &= 0 \end{aligned} \right. \quad \forall i \qquad (4\text{-}10)$$

$$w = \sum_{i=1}^{N_{sv}} \left( \alpha_i^* - \alpha_i \right) x_i \qquad (4\text{-}11)$$

The expected values are:-

$$y = \sum_{i=1}^{N} (a_i - a_j) * K(x_i, x) + b$$

The kernel function is:-

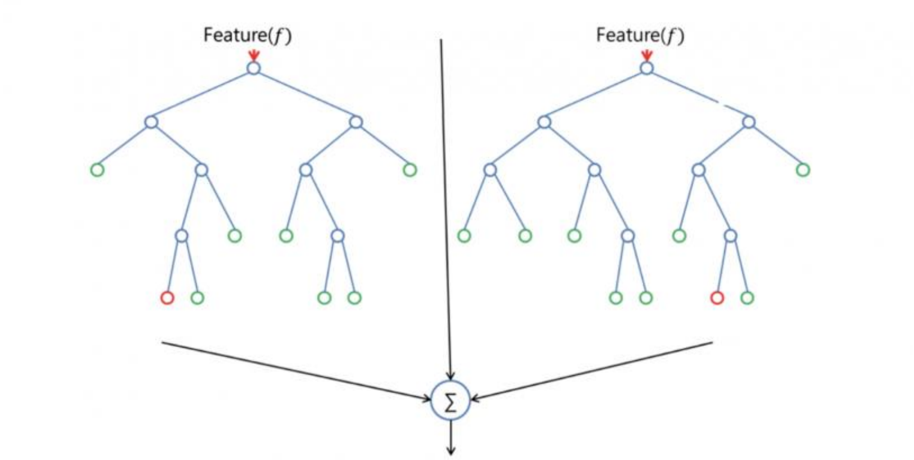
$$k(x_i, x_j) = e^{-(\frac{|x_i - x_j|^2}{2\sigma^2})}$$

**B2. Random Forest** [8]

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1 , \ldots , x_n$ with responses $Y = y_1, \ldots , y_n$ bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples $x'$ can be made by averaging the predictions from all the individual regression trees on $x'$:

$$f = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.



Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

The prediction of the whole forest is

$$F(X) = \frac{1}{M} \sum_{m=1}^{M} T_m(X) = \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{n} W_{im}(X) Y_i = \sum_{i=1}^{n} \Bigg( \frac{1}{M} \sum_{m=1}^{M} W_i$$

which shows that the random forest prediction is a weighted average of the Yi 's, with weights

$$W_i(X) = \frac{1}{M} \sum_{m=1}^{M} W_{im}(X)$$

## B3. Linear Regression:

The simple relation of linear regression is:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

It aims to predict the y value such that the error difference between the predicted value and the true value is minimum. The $\beta$ values are updated such that it minimizes the error (Cost function).

The Cost Function is:

$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2$$

To find the optimal values for the $'s$ , gradient descent optimization technique is used.

Batch Gradient descent algorithm:

Repeat {

$$\beta_i : \beta_i - \alpha . \sum_{j=1}^{n} \left( h_\theta \left( X^j \right) - y^j \right) . X_i^j$$

}

## B4. Lasso Regression Algorithm[11]

Lasso regression adds regularisation penalty term to the ordinary least-squares objective. But the results are noticeably different**.** With lasso regression, a subset of the coefficients is forced to be precisely zero which is a kind of automatic feature selection, since with the weight of zero the features are essentially ignored completely in the model. This sparse solution where only a subset of the most important features is left with non-zero weights also makes the model easier to interpret which is a huge advantage.

$$RSS_{LASSO}(w, b) = \sum_{\{i=1\}}^{N} (y_i - (w . x_i + b))^2 + \alpha \sum_{\{j=1\}}^{p} |w_j|$$

## B5. Ridge Regression Algorithm:

$$RSS_{RIDGE}(w, b) = \sum_{\{i=1\}}^{N} (y_i - (w . x_i + b))^2 + \alpha \sum_{\{j=1\}}^{p} (w_j)^2$$

As shown in the above image, the RSS modified by imposing that sum of squares penalty on the size of the W coefficients. The super power of Ridge Regression is that it minimize the RSS by enforce the W coefficients to be lower, but it does not enforce them to be zero**-** minimize their impact on the trained model to simplify the statistical model**.**

## B6. Elastic Net Regression Algorithm[10]:

Elastic Net aims at minimizing the following loss function:-

$$\frac{\sum_{i=1}^{n}(y_i-x_i^J\beta)^2}{2n}+\delta(\frac{1-\alpha}{2}\sum_{j=1}^{m}\beta_j^2+\alpha\sum_{j=1}^{m}|\beta_j|)$$

## **B7. Logistic Regression**:

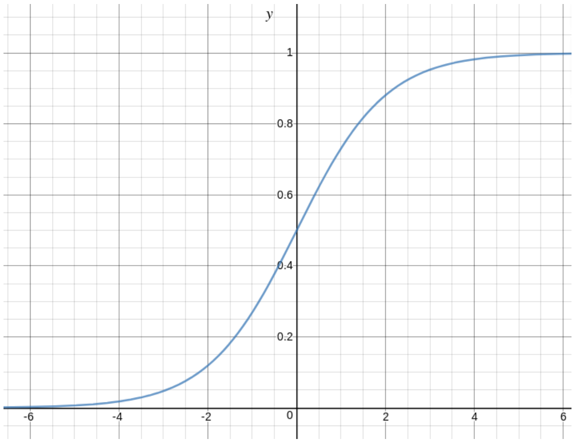
Similar to Linear Regression, Logistic regression is also a regression algorithm which gives output a label like “1”.

It models the data using sigmoid function:

$$g(z)=\frac{1}{1+e^{-z}}$$

The hypothesis function for classification is:

$$h(x_i)=g(\beta^T x_i)=\frac{1}{1+e^{-\beta^T x_i}}$$



Here is a plot showing g(z) on the left side.

We can infer from graph that:

- g(z) tends towards 1 as
- g(z) tends towards 0 as
- g(z) is always bounded between 0 and 1

The probability function could be written as:

$$P(y_i|x_i;\beta)=(h(x_i))^{y_i}\,(1-h(x_i))^{1-y_i}$$

## **B8.Gaussian Naïve Bayes**[11]:

If in a data set most of the attributes are continues then Gaussian Naive Bayes is used. It is assumed in this algorithm that predictor values are samples from Gaussian distribution.

Hence, Formula for conditional Probability becomes:-

$$P(x_i|y)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2})}$$

## **B9. Multinomial Naïve Bayes**:

With a multinomial event model,samples(feature vectors) represent the frequencies with which certain events have been generated by a multinomial $p_1, \ldots., p_n$ where $p_i$ is the probability that event $i$ occurs (

or $K$ such multinomials in the multiclass case). A feature vector $x = (x_1, \ldots., x_n)$ is then a histogram with $x_i$ counting the number of times event $i$ was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document. The likelihood of observing a histogram $x$ is given by

$$p(x|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

The multinomial naïve Bayes classifier becomes a  linear classifier when expressed in log-space.

$$\log\ p(C_k|x) \propto \log\ (p(C_k \prod_{i=1}^{n} p_{ki}^{x_i})$$

$$=\log p(C_k) + \sum_{i=1}^{n} x_i \,.\log p_{ki}$$

$$= b + w_k^T\ x$$

Where $b = \log \mathrm{p(C_k)}$ and $w_{ki} = \log p_{ki}$ .

## **B10.Bayesian Ridge Regression**:

Bayesian regression allows a natural mechanism to survive insufficient data or poorly distributed data by formulating linear regression using probability distributors rather than point estimates. The output or response ‘y’ is assumed to drawn from a probability distribution rather than estimated as a single value.

Mathematically, to obtain a fully probabilistic model the response y is assumed to be Gaussian distributed around $X_w$ as follows:

$$p(y|X, w, \alpha) = N(y|X_w, \alpha)$$

One of the most useful type of Bayesian Ridge regression which estimates a probabilistic model of the regression problem. The prior for the coefficient $w$ is given by spherical Gaussian as follows-

$$p(w|\gamma) = N\left(w\middle|0, \gamma^{-1} I_p\right)$$

This resulting model is Bayesian Ridge regression.

## **B11. Huber Regression**:

Huber regression is a regression technique that is robust to outliers, The idea is to use a different loss function rather than the traditional least-squares; we solve

$$_{i=1}^{m}(y_i - x_i^T)$$

For variable $\beta \in R^n$ where the loss $\emptyset$ is the Huber function with threshold $> 0$ ,

$$\emptyset(\mu) = \begin{cases} \mu^2 & if\ |\mu| \le M \\ 2M\mu - M^2 & if\ |\mu| > M \end{cases}$$

This function is identical to the least squares penalty for small residuals, but on large residuals, its penalty is lower and increases linearly rather than quadratically. it is thus more forgiving of outliers.

## **B12. Theil-sen Regression**:[12]

Theil's regression is a nonparametric method which is used as an alternative to robust methods for data sets with outliers. Although the nonparametric procedures perform reasonably well for almost any possible distribution of errors and they lead to robust regression lines, they require a lot of computation. it is proved to be useful when outliers are suspected, but when there are more than few variables, the application becomes difficult.

for a simple linear regression model to obtain the slope of a line that fits the data points, the set of all slopes of lines joining pairs of data points $(x_i,y_i)$ and $(x_j,y_j)$ , $x_j \neq x_i$ , for $1 \leq i \leq j \leq n$ should be calculated by;

$$b_{ij} = \frac{y_j - y_i}{x_j - x_i} \quad \text{-(1)}$$

Thus $b^*$ is the median of all Equation (1).

Hence, in this study, for n observations, we have $\frac{n(n-1)}{2}$ algebraic distinct $b_{ij} = b_{ji}$

But $a^*$ is the median of all $a_i = y_i - b^* x_i$ .

The mean square error is given in equation (2)

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y)^2}{n-k}$$

## **B13. Least angle regression**:

Least angle regression is an algorithm for fitting linear regression models to high-dimensional data. Suppose we expect a response variable to be determined by a linear combination of a subset of potential covariates. Then the LARS algorithm provides a means of producing an estimate of which variables to include, as well as their coefficients.

Algorithm:

- start with all coefficients $\beta$ equal to 0.
- Find the predictor $x_j$ most correlated to $y$.
- Increase the coefficient $\beta_j$ in the direction of the sign of its correlation with $y$.take residuals $r = y - y^{\wedge}$ along the way. Stop when some other predictor $x_k$ has a s much correlation with $r$ as $x_j$ has.
- Increase ($\beta_j$,$\beta_k$) in their joint least squares direction, until some other predictor $x_m$ has as much correlation with the residual $r$.

- Increase $(\beta_j, \beta_k, \beta_m)$ in their joint least squares direction, until some other predictor $x_n$ has as much correlation with the residual $r$.
- Continue until: all predictors are in the model.

**B14. XGB classifier**:

XGBoost is one of the most popular and efficient implementations of the Gradient Boosted Trees algorithm, a supervised learning method that is based on function approximation by optimizing specific loss functions as well as applying several regularization techniques.

The objective function (loss function and regularization) at iteration $t$ that we need to minimize is the following:

$$\varphi^{(t)} = \sum_{i=1}^{n} l\left(y_i, y_i^{\wedge(t-1)} + f_t(x_i)\right) + \omega(f_t)$$

XGBoost objective using second-order Taylor approximation:

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2$$

$$\varphi^{(t)} \cong \sum_{i=1}^{n}[l\left(y_i, y_i^{\wedge(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t)$$

Where:

$$g_i = \partial_{y^{\wedge(t-1)}} l\left(y_i, y_i^{\wedge(t-1)}\right) \text{ and } h_i = \partial^2_{y^{\wedge(t-1)}} l\left(y_i, y_i^{\wedge(t-1)}\right)$$

First and second order gradient statistics of the loss function.

Finally, if we remove the constant parts, we have the following simplified objective to minimize at step ***t***:

$$\varphi^{\sim(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t)$$

XGBoost simplified objective

The above is a sum of simple quadratic functions of one variable and can be minimized by using known techniques, so our next goal is to find a learner that minimizes the loss function at iteration ***t***.

$$argmin_x Gx + \frac{1}{2} Hx^2 = -\frac{G}{H}, H > 0 \qquad min_x\, Gx + \frac{1}{2} Hx^2 = -\frac{1}{2}\frac{G^2}{H}$$

Minimizing a simple quadratic function

$$\boldsymbol{\varphi}^{\sim(\mathbf{t})}(\mathbf{q}) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\sum_{i\in I_j} g_i{}^{2}}{\sum_{i\in I_j} h_i+\tau} + \gamma T$$

The tree learner structure q scoring function

Let’s take the case of binary classification and log loss objective function:

$$\boldsymbol{yln(p) + (1-y)ln(1-p)} \;\; \boldsymbol{where\ p} = \frac{1}{(1+e^{-x})}$$

Binary classification with Cross Entropy loss function

,where ***y*** is the real label in ***{0,1}*** and ***p*** is the probability score.

***p*** (score or pseudo-probability) is calculated after applying the famous sigmoid function into the *output of the* ***GBT model x***.
The output *x* of the model is the sum across the the CART tree learners.
So, in order to minimize the log loss objective function we need to find its 1st and 2nd derivatives (gradient and hessian) with respect to ***x***.

## B15. K-means Clustering:

This clustering algorithm separates data into the best suited group based on the information the algorithm already has. Data is separated in *k* different clusters, which are usually chosen to be far enough apart from each other spatially, in Eucledian Distance, to be able to produce effective data mining results. Each cluster has a center, called the centroid, and a data point is clustered into a certain cluster based on how close the features are to the centroid.

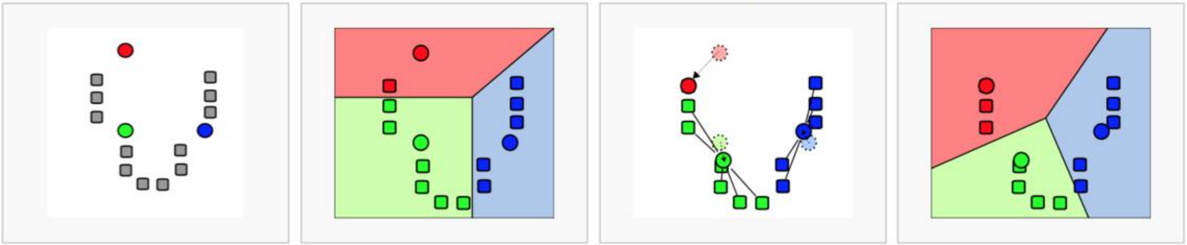
K-means algorithm iteratively **minimizes** the distances between every data point and its centroid in order to find the most optimal solution for all the data points.

1. *k* random points of the data set are chosen to be centroids.
2. Distances between every data point and the *k* centroids are calculated and stored.
3. Based on distance calculates, each point is assigned to the nearest cluster

4. New cluster centroid positions are updated: similar to finding a mean in the point locations
5. If the centroid locations changed, the process repeats from step 2, until the calculated new center stays the same, which signals that the clusters' members and centroids are now set.

Finding the minimal distances between all the points implies that data points have been separated in order to form the most compact clusters possible, with the least variance within them. In other words, no other iteration could have a lower average distance between the centroids and the data points found within them.

Demonstration of the standard algorithm



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

The K-means algorithm defined above aims at minimizing an objective function, which in this case is the squared error function.

The objective function for the K-means clustering algorithm is the squared error function:

$J = \sum_{i=1}^{k} \sum_{j=1}^{n} (||x_i - v_j||)^2 = 1$

where,

$||x_i - v_j||$ is the Eucledian distance between a point, $x_i$ and a centroid, $v_j$, iterated over all $k$ points in the $i^{ih}$ cluster, for all $n$ clusters.

[5]

## 7. Hypothesis:

null hypothesis: The average expected tenure after applying the retention technique selection model is same as before.

Alternative hypothesis: The average expected tenure after applying the retention technique selection model is greater than before.

## 8. Results:

After applying all the models and finally getting the newly updated tenure of the employees, we are comparing the previous and new tenures to examine whether our primary aim of the research to increase the employee tenure has been achieved or not.

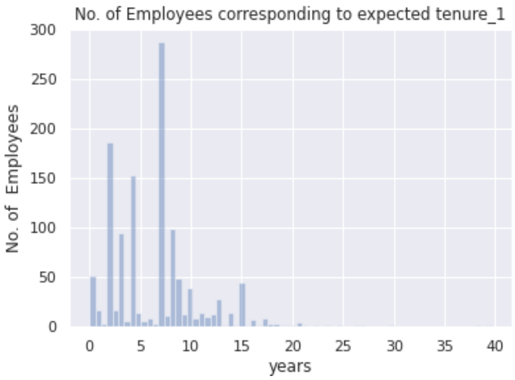
1046 employees out of total no. of employees 1233 , i.e 84.83% of total employees tenure has been increased, this is a very significant result of our model.

### 8a). Expected Tenure:

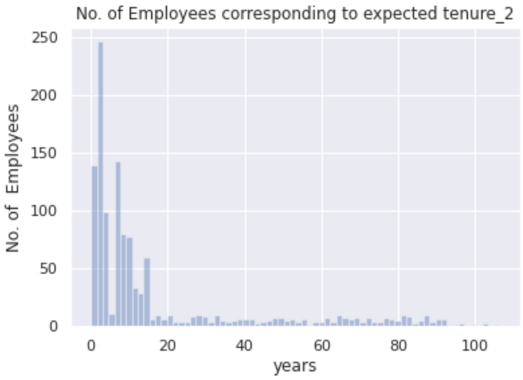
Let’s have a look at some of the statistical parameters of the results obtained for the expected tenure:

| | Tenure_1 | Tenure_2 | Percentage change |
|---|---|---|---|
| Mean | 6.41 | 17.59 | 174.17% |

The Percentage change of 174.17% between the average tenure before (Tenure_1) and the average tenure after(Tenure_2) is a very impressive figure. This suggests that our models aimed at improving the employee tenure so as to solve the problem of employee attrition have worked very well and on an average, there is an increase of more than double in the expected tenure of the employees. So, the company will be able to retain their employees for a much longer period using our suggested techniques.



Fig(1)

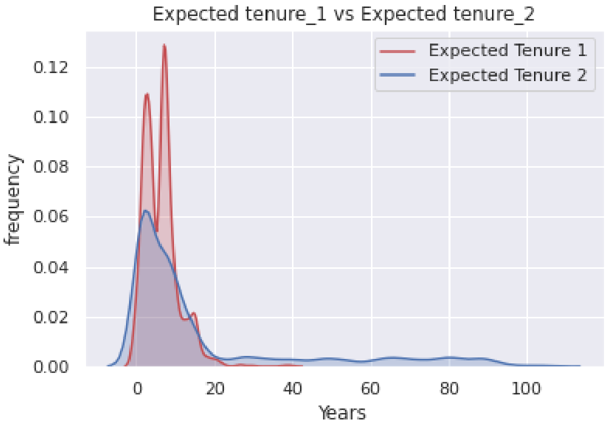


Fig(2)

Fig(1) on the left side,is a graph between expected tenure_1(in years) on x- axis vs it's frequency i.e number of employees on the y-axis.

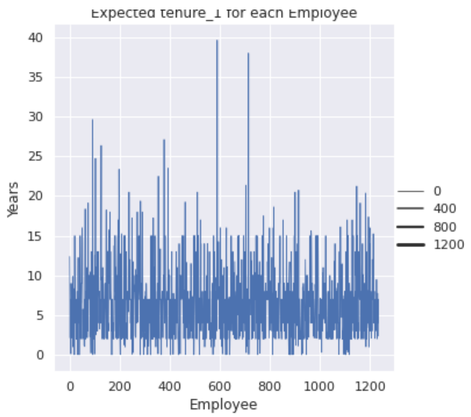
And Fig(2) on the right side, is a graph between expected tenure_2(in years) on x- axis vs it's frequency i.e number of employees on the y-axis.

From these two graphs, it can be observed that in the first graph of expected tenure_1 ,the majority of the employee's expected tenure are between 0-8 years and there are hardly any employee beyond 15 years. While in the second graph on the right of the expected tenure_2, while still there are employees between 0-8 years range but many employees are there now in the 8-20 range too and the employee's tenure seems to be distributed across all years as there are a descent number of employees between 20-80 years range too, while previously there was hardly any. So, the expected tenure graph has shifted towards right direction towards the increasing number of years and is distributed well now. Thus, this graph shows the improvement in employee tenure.

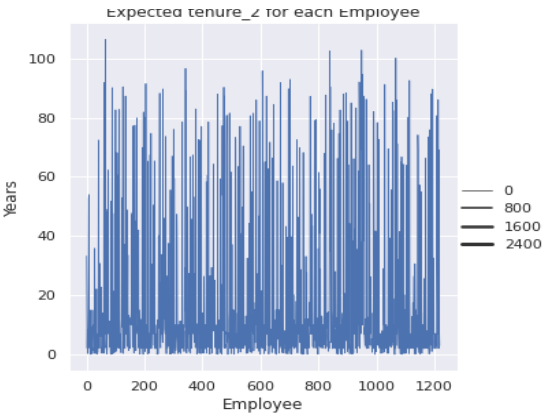


Fig(3)

Fig(3) is a line curve between years vs frequency for expected tenure_1 and expected tenure_2 . The red area which is expected tenure_1 has all it's area covered between 0-20 years range while the blue area which is expected tenure_2 has it's area spread across all years from 0-100 years. Thus, this is shows a significant improvement of expected tenures.



Fig(4)

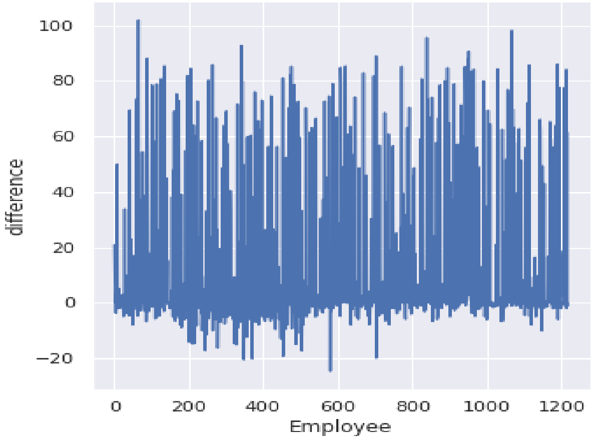


Fig(5)

Fig(4) on the left side,is a graph between Employee on x- axis vs expected tenure_1(in years) on the y-axis.

And Fig(5) on the right side, is a graph between  Employee on x- axis vs expected tenure_2(in years) on the y-axis.

In the first graph,the year corresponding to each employee is lesser as compared to in the second graph. The years corresponding to each employee in first graph is mostly between 0-20 , while in the second graph , it is much scattered at a higher range of years, many employees have expected years higher than 40 and 60. This shows that for all employees , at an individual level the expected tenure year has increased.



Fig(6)

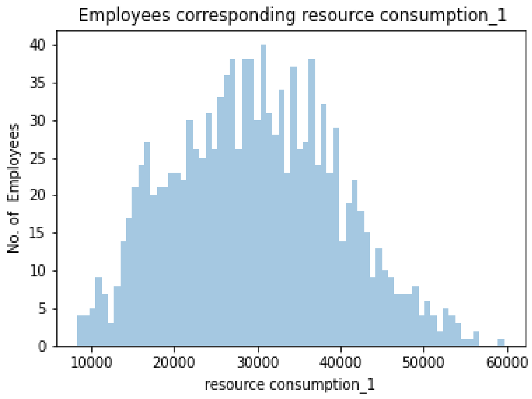
Fig(6) is the graph between Employee on x-axis and the difference between expected tenure_2 and expected tenure_1 plotted. This can be clearly seen from the graph that for almost everyone, the difference is positive above 0 line except a few. Hence, the expected tenure has increased at each employee level.

## 8b). Employee Resource consumption:

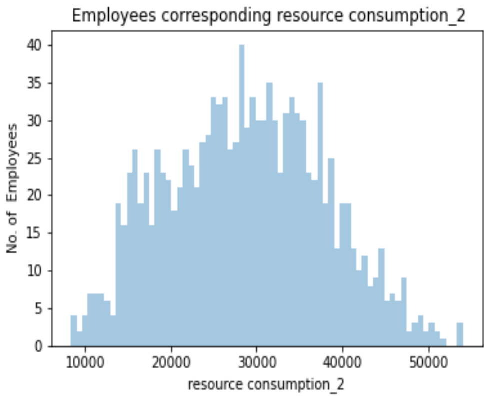
Let's have a look at some of the statistical parameters of the results obtained for the resource consumption:

| | Resource consumption_1 | Resource consumption_2 | Percentage change |
|---|---|---|---|
| Mean | 29812.89 | 28729.8 | -23.97 |

This table shows the average value of Employee resource consumption before and after applying the model. It can be clearly seen that after applying the model, the percentage change in the average value of employee resource consumption is 23.97%, by this much percentage the resource consumed by employees has been reduced.

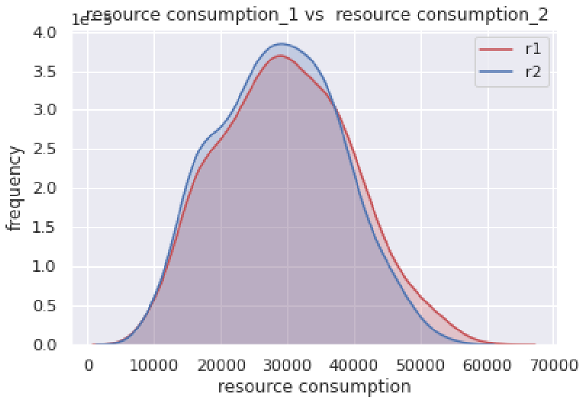


Fig(7)



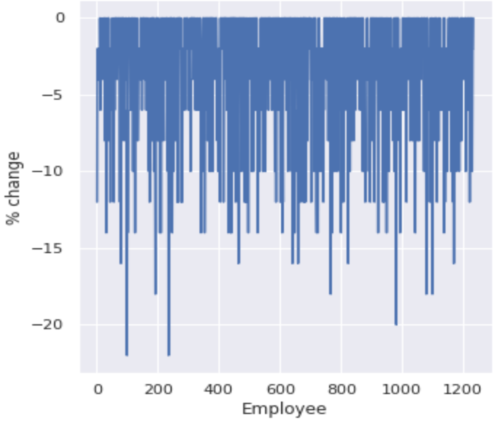
Fig(8)

Fig(7) is the graph between resource consumption_1 of an employee on x-axis vs no. of employees on y-axis.In Fig(7), the graph is centered near 3000 , representing maximum no. of employees consuming resource around 300 while in the next graph, the graph is centered around  2500. So, there is a shift towards left side from Fig(7) to Fig(8) which shows that our model has successfully minimized resources corresponding to employees.



Fig(9)

In Fig(9), a line plot of resource consumption_1 and resource consumption_2 with their frequencies is plotted. Clearly, the blue curve(resource consumption_2) is more shifted towards left side as compared to red curve(resource consumption_1) which implies after applying model the resource consumption of the employees has decreased, indicating successful results.



Fig(10)

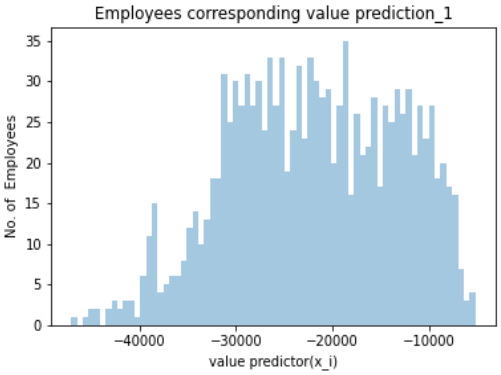
Fig(10) is the graph between employee on x-axis and %change in resource consumption_2 and resource consumption_1 on y-axis. For each employee, the %change in the graph is a non-positive  value and %changes upto 10% to 15% too , indicating each employee's resource is minimized or constant and secondly, minimized to a good extent(10-15% and even 20% and above too

## 8c). Employee value:

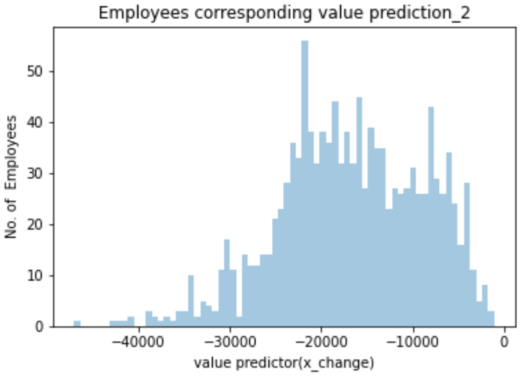
Let’s have a look at some of the statistical parameters of the results obtained for the Employee value:

| | Employee value_1 | Employee value_2 | Percentage change |
|---|---|---|---|
| Mean | -21943.14 | -16682.36 | 23.97 |

This table shows the averge score of the employee value before and after applying the model. It can be clearly seen that after applying the model, the value of employees for the company has increased by 23.97%.



Fig(11)

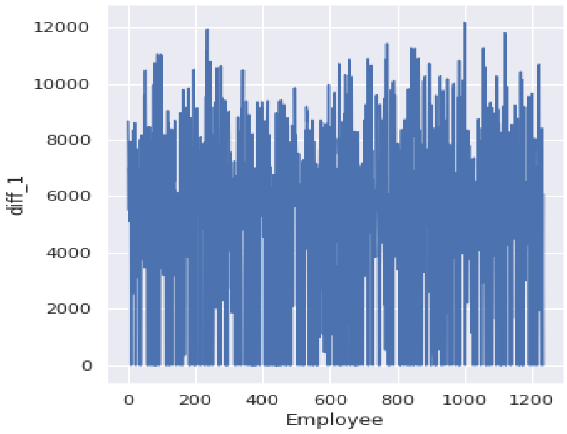


Fig(12)

Fig(11) is the plot of employee value prediction_1 on x-axis vs no. of employees on the y-axis.

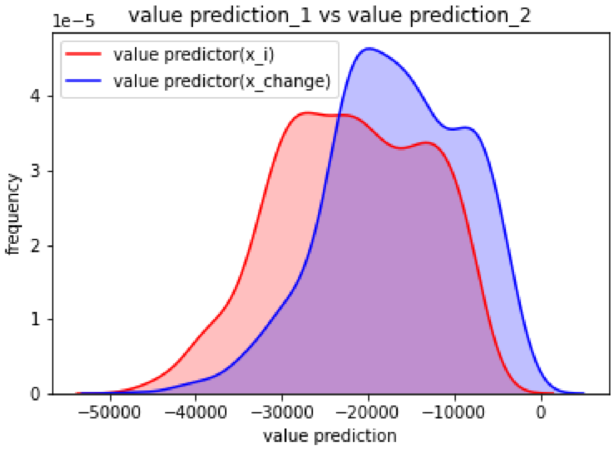
Fig(12) is the plot of employee value prediction_2 on x-axis vs no. of employees on the y-axis.

In Fig(11), the graph is centered between -2000 and -3000 i.e there are many employees having the value score in this range while in the Fig(12) ,the there is a shift of these employees of range -2000 and -3000 in fig(12) to range -2000 and -1000 i.e. this graph is centered around this range and there are very few employees only having range value range between - 2000 and -3000 . So, as evident there is a lot improvement in the employee value score.



Fig(13)

Fig(13) is a plot between each employee on x-axis and the difference between employee value_2 and employee value_1 on the y-axis.  For all the employees the difference is greater than 0 an dto a very significant extent.



Fig(14)

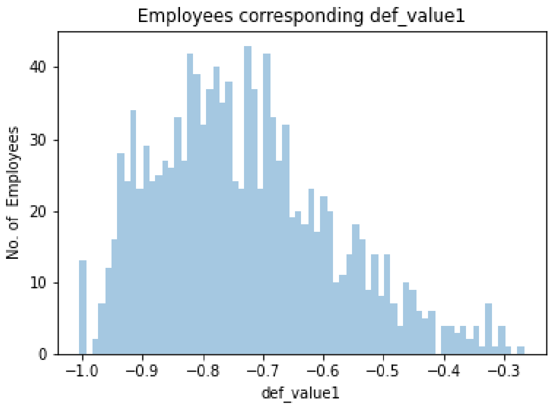
Fig(14) is graph between employee value prediction and frequency. The blue curve is the employee value_2 and has shifted towards right i.e the greater value score and by a great extent. This clearly implies that there is a significant improvement in the value score.

## 8d.) Employee def_value:

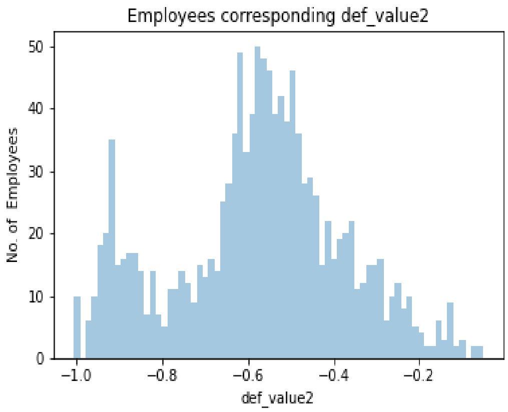
Let's have a look at some of the statistical parameters of the results obtained for the def_value

| | **def_value1** | **Def_value2** | **Percentage change** |
|---|---|---|---|
| **Mean** | -0.73 | -0.57 | 21.54 |

This table clearly shows that there is an increase of 21.54% in def_value after applying the model.



Fig(15)

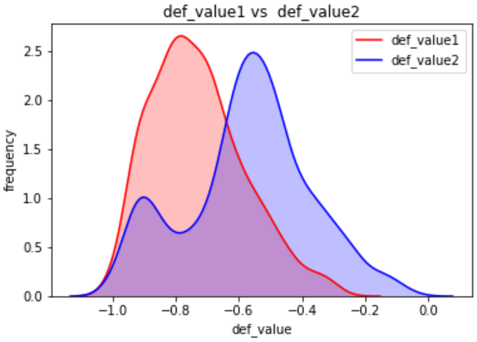


Fig(16)

Fig(15) is plot between def_value on x-axis and no. of employees on y-axis.

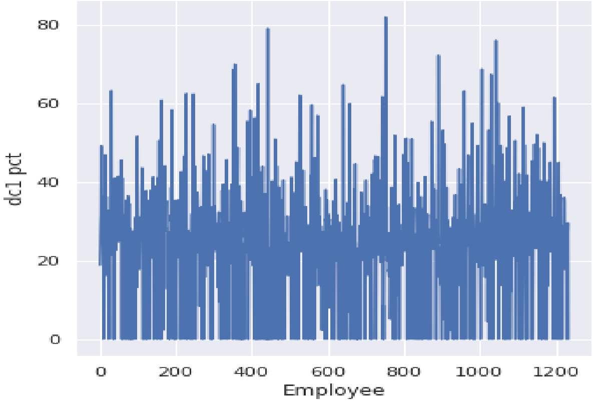
Fig(16) is a plot between def_value on x-axis and no. of employees on y-axis.

Fig(15) is centered around -0.8 while Fig(16) is centered around -0.5. So, this is a siginificant improvement in the def_value.



Fig(17)

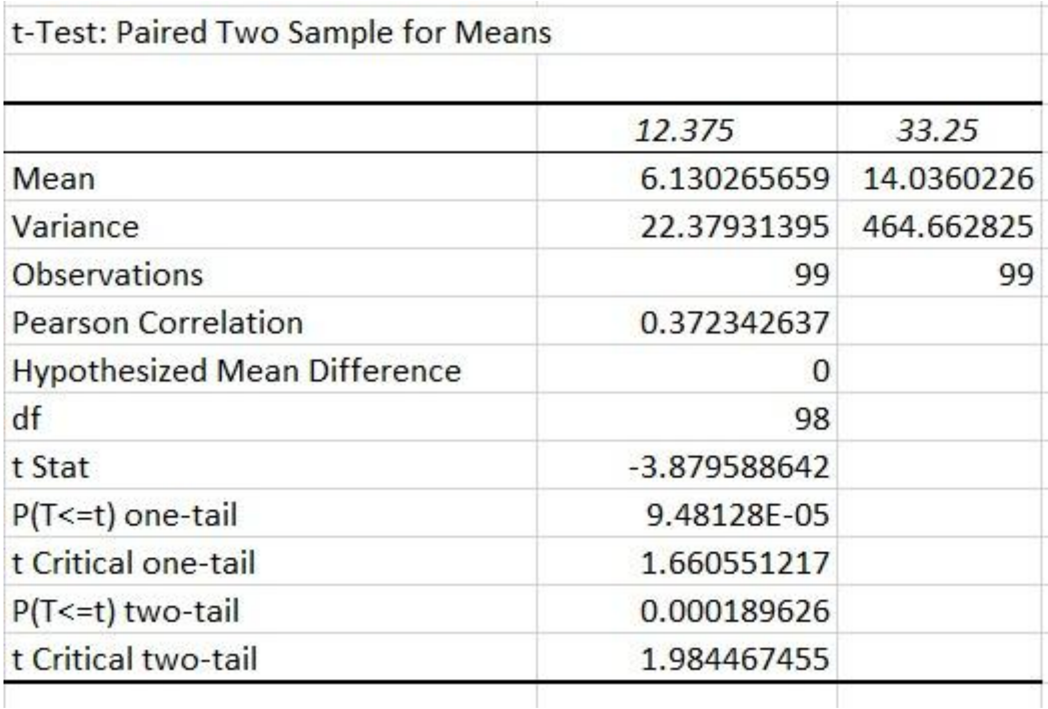
Fig(17) is the plot between def_valu1 and def_value2 with it's frequencies. The blue curve i.e. def_value 2 has shifted significantly towards right which indicates our model has worked very well.



Fig(18)

Fig(18) is the plot between each employee on the x-axis vs the percentage change in def_value2 and def_value1 on y-axis. Clearly, for each employee the percentage change is greater than 0 and to a significant extent too.

## <u>8e).Hypothesis testing:</u>



t-Test: Paired Two Sample for Means

| | *12.375* | *33.25* |
|---|---|---|
| Mean | 6.130265659 | 14.0360226 |
| Variance | 22.37931395 | 464.662825 |
| Observations | 99 | 99 |
| Pearson Correlation | 0.372342637 | |
| Hypothesized Mean Difference | 0 | |
| df | 98 | |
| t Stat | -3.879588642 | |
| P(T<=t) one-tail | 9.48128E-05 | |
| t Critical one-tail | 1.660551217 | |
| P(T<=t) two-tail | 0.000189626 | |
| t Critical two-tail | 1.984467455 | |

Performing hypothesis testing, the statistical measures obtained are in the table above.

p two –tail value of our testing is **0.00018** which is very very less than the significant level 0.05. Hence, our null hypothesis i.e "The average expected tenure after applying the retention technique selection model is same as before". Our obtained p-value too be much lesser than than significance level forms the evidence for the alternative hypothesis i.e *The average expected tenure after applying the retention technique selection model is greater than before*.

Thus, hypothesis testing clearly suggests that our model has worked perfectly well.

## 9.Conclusion:

Employee attrition has become a very big issue in the corporate all around the world. As, Employee attrition incurs huge cost in terms of lost productivity, recruitment and training costs. Between costs associated with separation, loss of productivity, recruitment, interviewing, training, and on boarding, the loss of a single employee is estimated to cost the company one third of that individual's annual salary .Nowadays, Every company is looking for strategies to retain employees. While there has been several studies being conducted to analyse the employee attrition but no significant research had been done about how to retain employee.

In our study, firstly we have analysed the employee attrition and built efficient models to predict the employee expected tenure then we built model: the retention technique selection model using the resource consumption and employee value prediction function, on how to retain the employees i.e to improve their expected tenure while maintaining the resource consumption and employee value constraints. Statistical and graphical analysis suggest that our models have every significantly improved the expected tenure of the employees, the value of an employee for the company while minimizing the resource constraint.

Therefore, we have shown that in enterprises our predicted model will provide its extended help to the Human Resource Management Department in taking better and determined decisions towards a specific employee and therefore benefit the company by a significant margin by retaining the existing hired workforce and reducing a large amount of resources to be spent on scouting, hiring and training.

## 10.References:

[1] Eric Ng Chee Hong ,Lam Zheng Hao,Ramesh Kumar,Charles Ramendran ,Vimala Kadiresan.An Effectiveness of Human Resource Management Practices on Employee Retention in Institute of Higher learning: - A Regression Analysis. International Journal of Business Research and Management (IJBRM), Volume (3) : Issue (2) : 2012.

[2] Matthew O’Connell and Mavis (Mei-Chuan) Kung. “The Cost of Employee Turnover.” In: Industrial Management (2007), pp. 14–19.

[3] . Lucas, S. (2013). How much employee turnover really cost you.

[4] Omer Cloutier,Laura Felusiak,Calvin Hill,Enda Jean Pemberton-Jones.The Importance of Developing Strategies for Employee Retention.Journal of Leadership, Accountability and Ethics Vol. 12(2) 2015.

[5] Yuan, H.; Yang, G.; Li, C.; Wang, Y.; Liu, J.; Yu, H.; Feng, H.; Xu, B.; Zhao, X.; Yang, X. Retrieving Soybean Leaf Area Index from Unmanned Aerial Vehicle Hyperspectral Remote Sensing: Analysis of RF, ANN, and SVM Regression Models. Remote Sens. 2017, 9, 309.

[6] Khan, Emad Afaq; Hayat Khan, Sumaira Muhammad.Factors Affecting Employee Attrition and Predictive Modelling Using IBM HR Data. Journal of Computational and Theoretical Nanoscience, Volume 16, Number 8, August 2019, pp. 3379- 3383(5).

[7] Awad M., Khanna R. (2015) Support Vector Regression. In: Efficient Learning Machines. Apress, Berkeley, CA

[8] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

[10] Hui Zou and Trevor Hastie(2005). Regularization and variable selection via the Elastic Net.Journal of the Royal Statistical Society, Series B. 301-320

[11].Hybrid of Naive Bayes and Gaussian Naive Bayes for Classification: A Map Reduce Approach Shikha Agarwal, Balmukumd Jha, Tisu Kumar, Manish Kumar, Prabhat Ranjan. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-6S3, April 2019

[12] COMPARISONS OF THEIL'S AND SIMPLE REGRESSION ON NORMAL AND NON-NORMAL DATA SET WITH DIFFERENT SAMPLE SIZES; International Journal of Management and Applied Science, ISSN: 2394-7926; 1ESEMOKUMO PEREWAREBO AKPOS, 2OPARA, JUDE

[13]. Dawley, D., Houghton, J.D.& Bucklew, N.S. (2010). Perceived organizational support and turnover intention: the mediating effects of personal sacrifice and job fit. The Journal of Social Psychology, 150(3), 238-257. Dibble, S. (1999).

[14]. Gilmore, D.C. & Turner, M. (2010). Improving executive recruitment and retention. Psychologist - Manager Journal (Taylor & Francis Ltd.), 13, 125-128.

[15].Gabriel, A.S., Diefendorff, J.M., Moran, C.M. & Greguras, G.J. (2014). The dynamic relationships of work affect and job satisfaction with perceptions of fit. Personnel Psychology(67), 389-420

## 11.Appendix:

These are the links which contains complete information about each and every process such as coding models ,graphs and statistical calculations. The idea to provide these links is to keep each and every process involved in the research work completely transparent and to ensure that all the items put in the research paper has been completely research by own.

ipython notebook containing the full codes for all the models:

https://colab.research.google.com/drive/1F_csNAlLrXX81P3ZsF-V7l2jowtXp4MI?usp=sharing

ipython notebook containing the full codes for all the graphs of expected tenure:

https://colab.research.google.com/drive/1Qol9FPXCOTdlwjzghjfWqAEeLuhCkTLC?usp=sharing ipython notebook containing the full codes for all the graphs of resource consumption:

https://colab.research.google.com/drive/1u6hOkb69QWby96BG6b1jrXJr6sq6L98L?usp=sharing

ipython notebook containing the full codes for all the graphs of employee value and def_value:

https://colab.research.google.com/drive/1GZ0LhnaChzWW5sbcRNG0XTFNiQEMCRq-?usp=sharing

My github repository which contains all the excel and csv files used for graphs(aouund 22 in total), result storage , statistical calculation such as p-value calculation.

https://github.com/Prateek190/data.git