w1. INTODUCTION

1.1 Overview

- The objective of this project is to predict the attrition rate for each employee, to find out who's more likely to leave the organization.
- It will help organization to find ways to prevent attrition or to plan in advance the hiring of new candidate.

1.2 Purpose

- Attrition proves to be costly and time consuming problem for the organization and it also leads to loss of productivity.
- The scope of the project extends to companies in all industries.

2. LITERATURE SURVEY

2.1 Existing Problem

Bill Gates was once quoted as saying,
" You take away our top 20 employees and we [Microsoft] become a mediocre company”.
His statement cuts to the core of a **major problem: employee attrition**. An organization is only as good as its employees, and these people are the true source of its competitive advantage. **Organizations face huge costs resulting from employee turnover**. Some costs are tangible such as training expenses and the time it takes from when an employee starts to when they become a productive member. However, the most important costs are intangible. Consider what’s lost when a productive employee quits: new product ideas, great project management, or customer relationships.

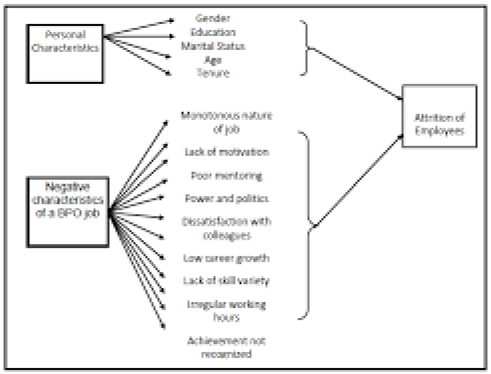
2.2 Proposed Solution

With advances in machine learning and data science, its possible to **not only predict employee attrition but to understand the key variables that influence turnover.**

3. THEORITICAL ANALYSIS

3.1 Block Diagram

The factors influencing the employee attrition in BPO can be illustrated as below:

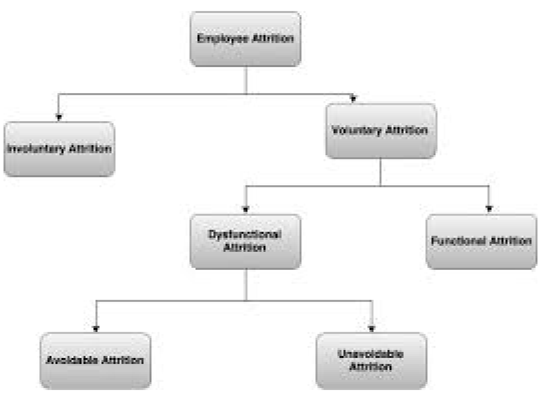


4. EXPERIMENTAL ANALYSIS

The dataset included various important features including average number of monthly hours, number of projects, years spent in the company and whether the employee received a promotion in the last five years. There were a total of nine features, out of which two were categorical and seven were numeric

- Data preprocessing
- Model Validation
- System Environment Specification

5. FLOWCHART



6. RESULT

- The dataset doesnot feature any missing values or any reductant features.
- The strongest positive correlations with the target features are Distance from home, Job Satisfaction, marital status ,overtime and business travel
- The strongest negative correlation with the target features are: Performance Rating and Training times last year.

7. ADVANTAGES AND DISADVANTAGES

Advantges

Not all turnovers are negative, we genrally feel that an employee leaving the organization is detrimental to the organization, but there is a flip side to it. Employees leaving an organization may lead to benefits. This type of job attrition is called 'healthly attrition' and is needed for growth and development of an organization

1. Higher manpower costs
2. Negative people effect
3. New Idea
4. Higher Performance
5. Setting the culture right

Disadvantages

When employees leave the organization it is a loss to the company, the team and the individuals.

Employees are the backbone of any organization and their departing may lead to lot of various losses to company on different aspects.

1. Decrease overall performance
2. Daily task management
3. Increased cost
4. Lack of knowledge employees
5. Create negative image
6. Employee development

8. CONCLUSION

Predictive Attrition Model helps in not only taking preventive measures but also into making better hiring decisions. Deriving trends in the candidate's performance out of past data is important in order to predict the future trends, as well as to board new employees. Moreover, HR can use the employee data to predict attrition, the possible reasons behind it and can take appropriate measures to prevent it.

9. FUTURE SCOPE

1. Transportation should be provided to employees living in the same area.
2. Plan and allocate project in such a way to avoid the overtime.
3. Employees which hit their two-year anniversary should be identified as potentially having a hgher risk of leaving.

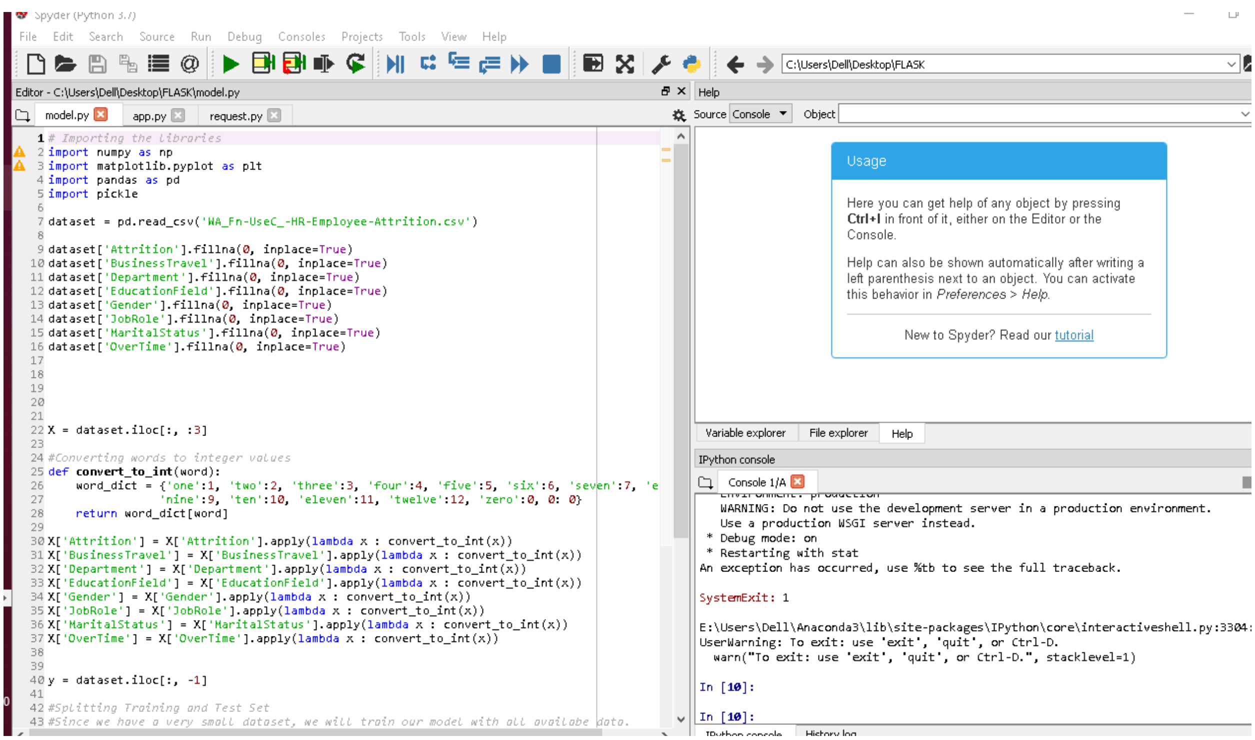
4. Gather information on industry benchmarks to determine if the company is providing competitive wages.

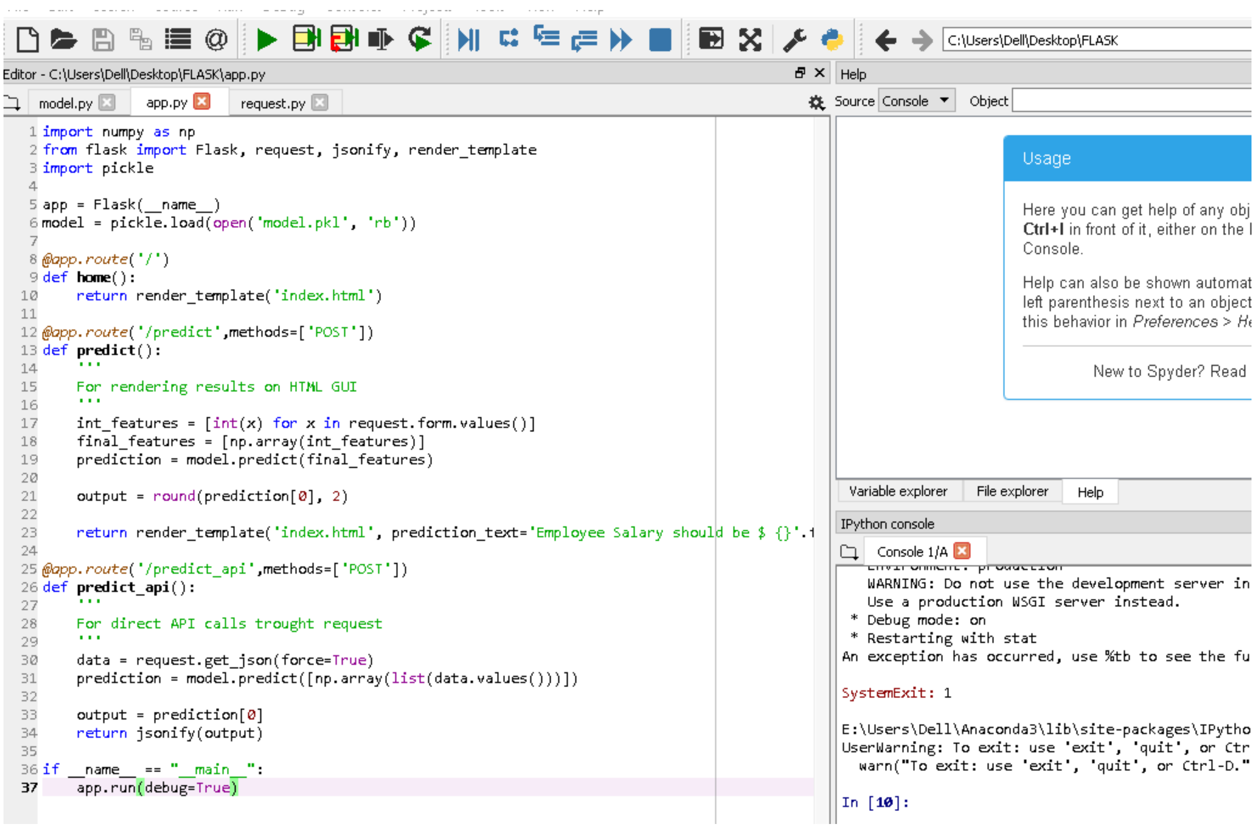
10. BIBLOGRAPHY

The content for this project report has taken from following links

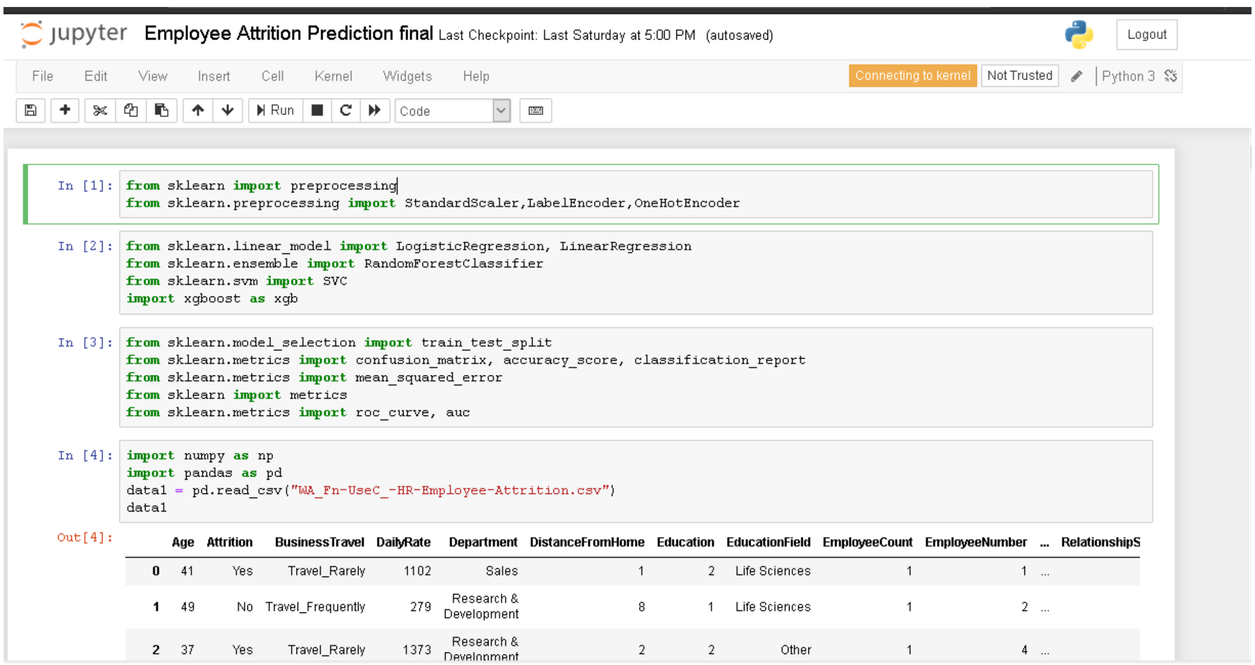
1. https://www.slideshare.net/ShrutiMohan5/predicting-employee-attrition-149113703
2. https://medium.com/analytics-vidhya/predict-employee-attrition-a34e2c5a972d
3. https://www.kaggle.com/c/1056lab-employee-attrition-prediction/

SCREENSHOTS





C:\Users\Dell\Desktop\FLASK
Editor - C:\Users\Dell\Desktop\FLASK\app.py
model.py
app.py
request.py
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
@app.route('/')
def home():
return render_template('index.html')
@app.route('/predict',methods=['POST'])
def predict():
'''
For rendering results on HTML GUI
'''
int_features = [int(x) for x in request.form.values()]
final_features = [np.array(int_features)]
prediction = model.predict(final_features)
output = round(prediction[0], 2)
return render_template('index.html', prediction_text='Employee Salary should be $ {}'.f
@app.route('/predict_api',methods=['POST'])
def predict_api():
'''
For direct API calls trought request
'''
data = request.get_json(force=True)
prediction = model.predict([np.array(list(data.values()))])
output = prediction[0]
return jsonify(output)
if __name__ == "__main__":
app.run(debug=True)
Help
Source Console
Object
Usage
Here you can get help of any obj
Ctrl+I in front of it, either on the
Console.
Help can also be shown automat
left parenthesis next to an object
this behavior in Preferences > He
New to Spyder? Read
Variable explorer
File explorer
Help
IPython console
Console 1/A
WARNING: Do not use the development server in
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
An exception has occurred, use %tb to see the fu
SystemExit: 1
E:\Users\Dell\Anaconda3\lib\site-packages\IPytho
UserWarning: To exit: use 'exit', 'quit', or Ctr
warn("To exit: use 'exit', 'quit', or Ctrl-D."
In [10]:



jupyter
Employee Attrition Prediction final
Last Checkpoint: Last Saturday at 5:00 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Connecting to kernel
Not Trusted
Python 3
Run
Code
In [1]:
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler,LabelEncoder,OneHotEncoder
In [2]:
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import xgboost as xgb
In [3]:
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
from sklearn.metrics import mean_squared_error
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
In [4]:
import numpy as np
import pandas as pd
data1 = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
data1
Out[4]:
Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education EducationField EmployeeCount EmployeeNumber ... RelationshipS
0 41 Yes Travel_Rarely 1102 Sales 1 2 Life Sciences 1 1 ...
1 49 No Travel_Frequently 279 Research & Development 8 1 Life Sciences 1 2 ...
2 37 Yes Travel_Rarely 1373 Research & 2 2 Other 1 4 ...

jupyter **Employee Attrition Prediction final** Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3

4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...
...	...	...	...	...	...	...	...	...	...	...	...
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...

1470 rows × 35 columns

In [5]:
```
data1.head()
```

Out[5]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatis
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3

5 rows × 35 columns

In [6]:
```
data1.shape
```

Out[6]:
```
(1470, 35)
```

In [7]:
```
data1.isnull().sum()
```

Out[7]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
```

File Edit View Insert Cell Kernel Widgets Help Not Trusted Pytho

```
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [8]:
```
data1.isnull().any()
```

Out[8]:
```
Age                         False
Attrition                   False
BusinessTravel              False
DailyRate                   False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount               False
EmployeeNumber              False
EnvironmentSatisfaction     False
Gender                      False
HourlyRate                  False
JobInvolvement              False
JobLevel                    False
```

jupyter Employee Attrition Prediction final Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted

```
MonthlyIncome               False
MonthlyRate                 False
NumCompaniesWorked          False
Over18                      False
OverTime                    False
PercentSalaryHike           False
PerformanceRating           False
RelationshipSatisfaction    False
StandardHours               False
StockOptionLevel            False
TotalWorkingYears           False
TrainingTimesLastYear       False
WorkLifeBalance             False
YearsAtCompany              False
YearsInCurrentRole          False
YearsSinceLastPromotion     False
YearsWithCurrManager        False
dtype: bool
```

In [9]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder()
data1['Attrition'] = labelencoder_y.fit_transform(data1['Attrition'])
data1['BusinessTravel'] = labelencoder_y.fit_transform(data1['BusinessTravel'])
data1['Department'] = labelencoder_y.fit_transform(data1['Department'])
data1['EducationField'] = labelencoder_y.fit_transform(data1['EducationField'])
data1['Gender'] = labelencoder_y.fit_transform(data1['Gender'])
data1['JobRole'] = labelencoder_y.fit_transform(data1['JobRole'])
data1['MaritalStatus'] = labelencoder_y.fit_transform(data1['MaritalStatus'])
print(data1)
```

File Edit View Insert Cell Kernel Widgets Help

Run Code

```
      Age  Attrition  BusinessTravel  DailyRate  Department  DistanceFromHome  \
0      41          1               2       1102           2                 1
1      49          0               1        279           1                 8
2      37          1               2       1373           1                 2
3      33          0               1       1392           1                 3
4      27          0               2        591           1                 2
...   ...        ...             ...        ...         ...               ...
1465   36          0               1        884           1                23
1466   39          0               2        613           1                 6
1467   27          0               2        155           1                 4
1468   49          0               1       1023           2                 2
1469   34          0               2        628           1                 8

      Education  EducationField  EmployeeCount  EmployeeNumber  ...  \
0             2               1              1               1  ...
1             1               1              1               2  ...
2             2               4              1               4  ...
3             4               1              1               5  ...
4             1               3              1               7  ...
...         ...             ...            ...             ...  ...
1465          2               3              1            2061  ...
1466          1               3              1            2062  ...
1467          3               1              1            2064  ...
1468          3               3              1            2065  ...
1469          3               3              1            2068  ...

      RelationshipSatisfaction  StandardHours  StockOptionLevel  \
0                            1             80                 0
1                            4             80                 1
```

jupyter Employee Attrition Prediction final Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

```
y = data1.iloc[:,3].values
```

In [11]:
```
print(x[0],y[0])
```

```
[41  1  2] 1102
```

In [12]:
```
from sklearn.compose import ColumnTransformer
```

In [13]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [14]:
```
ct = ColumnTransformer(transformers=[("oh",OneHotEncoder(),[0])],remainder="passthrough")
```

In [15]:
```
x=ct.fit_transform(x)
```

In [16]:
```
print(x)
```

```
  (0, 23)	1.0
  (0, 43)	1.0
  (0, 44)	2.0
  (1, 31)	1.0
  (1, 44)	1.0
  (2, 19)	1.0
  (2, 43)	1.0
  (2, 44)	2.0
  (3, 15)	1.0
  (3, 44)	1.0
  (4, 9)	1.0
```

jupyter Employee Attrition Prediction final Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
(1469, 44)    3.0064854641274565
```

In [18]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=10)
```

In [19]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1029, 45)
(441, 45)
(1029,)
(441,)
```

In [20]:
```
'''
classifier_reg = LogisticRegression()
classifier_reg.fit(x_train,y_train)

Commented out as LogReg is used for Classification and your task is Regression
'''

reg = LinearRegression()
reg.fit(x_train, y_train)
```

Out[20]:
```
LinearRegression()
```

In [21]:
```
y_pred=reg.predict(x_test)
```

In [22]:
```
print(y_pred)#.astype(int))  #Here rating are predicted as float(As obvious), if you want int values you can use .astype(int)
```

localhost:8888/notebooks/Desktop/remote internship 2020/data sets/Employee Attrition Prediction final.ip

jupyter Employee Attrition Prediction final Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted

```
 873.55506943  844.9958983   700.26543995  729.52807371  947.77601121
 799.25805454  740.14765209  902.26362546  844.9958983   666.8746855
 874.98454982  880.66167847  798.98837332  873.55506943  834.80817992
 819.88217514  838.66326789  740.14765209  782.84367849  797.15441215
 740.14765209  772.1383769   962.70201599  791.45638511  815.40399893
 844.9958983   731.53494547  753.05903372  746.0874228   872.04897185
 858.4322226   966.65401446  878.12492948  864.42603518  740.14765209
 834.80817992  797.15441215  790.64534792  796.68896486  676.04107515
 821.43785465  744.45407848  857.75913658  774.43778536 1185.10476486
 731.16141802]
```

In [23]:
```
cols = ['Model','max_error','mean_squared_error','mean_squared_log_error',
        'r2_score', 'mean_absolute_error','explained_variance_score']
models_report = pd.DataFrame(columns=cols)

print(models_report)
```

```
Empty DataFrame
Columns: [Model, max_error, mean_squared_error, mean_squared_log_error, r2_score, mean_absolute_error, explained_vari
core]
Index: []
```

In [24]:
```
rows = np.array(["Linear Regression",
                 metrics.max_error(y_test, y_pred),
                 metrics.mean_squared_error(y_test, y_pred),
                 metrics.mean_squared_log_error(y_test, y_pred),
                 metrics.r2_score(y_test, y_pred),
                 metrics.mean_absolute_error(y_test, y_pred),
                 metrics.explained_variance_score(y_test, y_pred)])
```

localhost:8888/notebooks/Desktop/remote internship 2020/data sets/Employee Attrition Prediction final.ip

jupyter **Employee Attrition Prediction final** Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted

Run Code

```
 770.28444696  878.12492948  791.45638511  805.09594514  698.34923104
 873.55506943  844.9958983   700.26543995  729.52807371  947.77601121
 799.25805454  740.14765209  902.26362546  844.9958983   666.8746855
 874.98454982  880.66167847  798.98837332  873.55506943  834.80817992
 819.88217514  838.66326789  740.14765209  782.84367849  797.15441215
 740.14765209  772.1383769   962.70201599  791.45638511  815.40399893
 844.9958983   731.53494547  753.05903372  746.0874228   872.04897185
 858.4322226   966.65401446  878.12492948  864.42603518  740.14765209
 834.80817992  797.15441215  790.64534792  796.68896486  676.04107515
 821.43785465  744.45407848  857.75913658  774.43778536 1185.10476486
 731.16141802]
```

In [23]:

```
cols = ['Model','max_error','mean_squared_error','mean_squared_log_error',
        'r2_score', 'mean_absolute_error','explained_variance_score']
models_report = pd.DataFrame(columns=cols)

print(models_report)
```

```
Empty DataFrame
Columns: [Model, max_error, mean_squared_error, mean_squared_log_error, r2_score, mean_absolute_error, explained_varia
core]
Index: []
```

In [24]:

```
rows = np.array(["Linear Regression",
                 metrics.max_error(y_test, y_pred),
                 metrics.mean_squared_error(y_test, y_pred),
                 metrics.mean_squared_log_error(y_test, y_pred),
                 metrics.r2_score(y_test, y_pred),
                 metrics.mean_absolute_error(y_test, y_pred),
                 metrics.explained_variance_score(y_test, y_pred)])
```

localhost:8888/notebooks/Desktop/remote internship 2020/data sets/Employee Attrition Prediction final.ip

jupyter **Employee Attrition Prediction final** Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

```
Reg_report = models_report.append(temp1, ignore_index = True)
Reg_report
```

Out[24]:

	Model	max_error	mean_squared_error	mean_squared_log_error	r2_score	mean_absolute_error	explained_variance_score
0	Linear Regression	1043.1047648598567	172301.8190985129	0.4940112052587489	-0.07772779874596614	349.3459593777875	-0.07523144662864523

In [25]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.linear_model import ElasticNet
pipe = Pipeline([
    ('rescale', StandardScaler(with_mean=False)),
    ('enet', ElasticNet())
])
```

In [26]:

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

Out[26]:

```
Pipeline(steps=[('rescale', StandardScaler(with_mean=False)),
                ('enet', ElasticNet())])
```

In [27]:

```
y_predict=pipe.predict(x_test)
y_predict
```

Out[27]:

```
array([ 886.16702856,  843.85302785,  883.58942605,  865.10801632,
        753.27684107,  787.29736364,  787.20798184,  814.44553394,
        813.05398632,  825.26424697,  784.5879239 ,  798.89068945,
        813.05398632,  750.38048678,  825.26424697,  727.20301047,
        861.61750506,  775.17280644,  768.27369459,  850.93244611,
        806.93285558,  804.23720742,  813.05398632,  802.20327245,
```

jupyter **Employee Attrition Prediction final** Last Checkpoint: Last Saturday at 5:00 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run | Code

```
826.89114903,  721.08187972,  855.49637432,  875.34831553,
707.14896884,  839.35453362,  825.26424697,  869.22718479,
```

In [28]:

```
rows = np.array(["ELastic Net",
                 metrics.max_error(y_test, y_predict),
                 metrics.mean_squared_error(y_test, y_predict),
                 metrics.mean_squared_log_error(y_test, y_predict),
                 metrics.r2_score(y_test, y_pred),
                 metrics.mean_absolute_error(y_test, y_predict),
                 metrics.explained_variance_score(y_test, y_predict)])
temp2 = pd.Series(rows, index = cols)

Reg_report = Reg_report.append(temp2, ignore_index = True)
```

```
               Model           max_error  mean_squared_error  \
0  Linear Regression  1043.1047648598567   172301.8190985129
1        ELastic Net   915.3322479071323  166560.77872274717

  mean_squared_log_error              r2_score mean_absolute_error  \
0     0.4940112052587489  -0.07772779874596614   349.3459593777875
1     0.4862341686303916  -0.07772779874596614   344.8101705846798

  explained_variance_score
0     -0.07523144662864523
1    -0.039109234859640685
```

In [29]:

```
Reg_report
```

Out[29]:

	Model	max_error	mean_squared_error	mean_squared_log_error	r2_score	mean_absolute_error	explained_variance_score