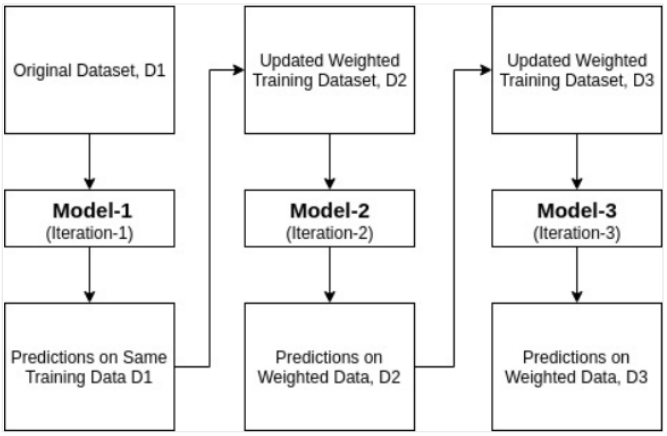
Prediction whether a HR Employee will leavee or not Using Adaboost method.

Briyana Rana – 50442498
brana@buffalo.edu

I. ADABOOST CLASSIFIER

An ensemble classifier is Ada-boost, which is similar to Random Forest Classifier. (Multiple classifier algorithms are merged to form an ensemble classifier, whose output is the sum of the results from all of the individual classifier algorithms.) The Ada-boost classifier combines weak classifier techniques to produce a strong classifier. The things might not be sufficiently classified by one algorithm. But if we combine numerous classifiers with the appropriate weighting during the final voting and the selection of a training set at each iteration, we might have a classifier with a respectable accuracy rating. It can be utilized in various domains such as credit, insurance, marketing, and sales.



The three most important variables are base estimator, n estimators, and learning rate.

base estimator: The model was trained using this unreliable learner. DecisionTreeClassifier is employed by default as a weak learner for training purposes. You may also choose from a variety of machine learning algorithms.
number of poor pupils to receive iterative training (**n estimators**).
Learning rate: It has an impact on the weighting of poor learners. It utilizes 1 as its default setting.

AdaBoost is simple to use. By integrating weak learners, it repeatedly corrects the flaws in the weak classifier and increases accuracy. With AdaBoost, several base classifiers are available. Overfitting is not a problem with AdaBoost. The findings of experiments can reveal this, but there is no clear cause.
Each weak classifier is trained using a random subset of overall training set.
Ada-boost provides weight to every training item after a classifier has been trained at any level. The misclassified item is given a heavier weight so that it has a larger likelihood of appearing in the training subset of the following classifier.

Each classifier is trained before having a weight assigned to it depending on accuracy. Higher weights are given to classifiers that are more accurate so that they can influence outcomes more.

A. Mathematical Understanding of Adaboost:

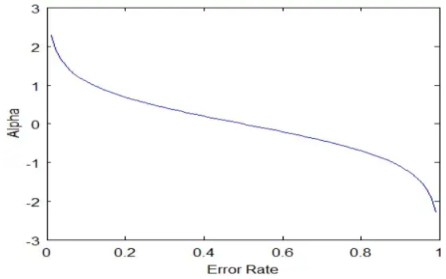
$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

h_t(x) is the output of weak classifier t for input x

alpha_t is weight assigned to classifier.

alpha_t = 0.5 * ln((1 — E)/E) : weight of classifier is straight forward, it is based on the error rate E.

A plot of alpha_t v/s error rate



$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Here, D_t is weight at previous level. D_{t+1} is the updated weight.

We normalize the weights by dividing each of them by the sum of all the weights, Z_t.

II. Experiment Motivation

This Dataset is an employee retention related dataset that can be used for Classification and Boosting Algorithms to solve the problem of which employee will stay back or leave a company. The Dataset contains various attributes related to an employee that will help in determining whether he will leave or stay back.

III. METHODOLOGY

The current paper focuses on using a special machine learning model called Adaboost Classifier in addition to extracting significant non-correlated features from the dataset. The model will proceed according to the standard procedures, which include data extraction followed by exploratory data analysis, data preprocessing, model construction, and model parameter evaluation.

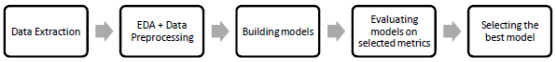


Fig 1. Flowchart describing Methodology

A. Dataset:

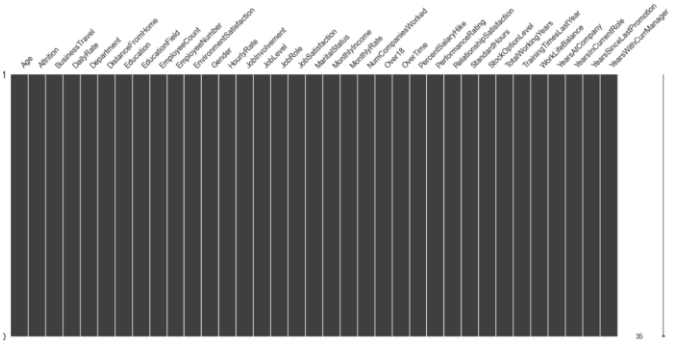
I used a dataset from Kaggle which contains 1470 observations and 35 columns. The columns are:

```
0   Age                       1470 non-null   int64
1   Attrition                 1470 non-null   object
2   BusinessTravel            1470 non-null   object
3   DailyRate                 1470 non-null   int64
4   Department                1470 non-null   object
5   DistanceFromHome          1470 non-null   int64
6   Education                 1470 non-null   int64
7   EducationField            1470 non-null   object
8   EmployeeCount             1470 non-null   int64
9   EmployeeNumber            1470 non-null   int64
10  EnvironmentSatisfaction   1470 non-null   int64
11  Gender                    1470 non-null   object
12  HourlyRate                1470 non-null   int64
13  JobInvolvement            1470 non-null   int64
14  JobLevel                  1470 non-null   int64
15  JobRole                   1470 non-null   object
16  JobSatisfaction           1470 non-null   int64
17  MaritalStatus             1470 non-null   object
18  MonthlyIncome             1470 non-null   int64
19  MonthlyRate               1470 non-null   int64
20  NumCompaniesWorked        1470 non-null   int64
21  Over18                    1470 non-null   object
22  OverTime                  1470 non-null   object
23  PercentSalaryHike         1470 non-null   int64
24  PerformanceRating         1470 non-null   int64
25  RelationshipSatisfaction  1470 non-null   int64
26  StandardHours             1470 non-null   int64
27  StockOptionLevel          1470 non-null   int64
28  TotalWorkingYears         1470 non-null   int64
29  TrainingTimesLastYear     1470 non-null   int64
30  WorkLifeBalance           1470 non-null   int64
31  YearsAtCompany            1470 non-null   int64
32  YearsInCurrentRole        1470 non-null   int64
33  YearsSinceLastPromotion   1470 non-null   int64
34  YearsWithCurrManager      1470 non-null   int64
```

B. Exploratory Data Analysis and data preprocessing:

The data has around 1470 rows and 35 columns. The view of the subset of the dataset is given below:

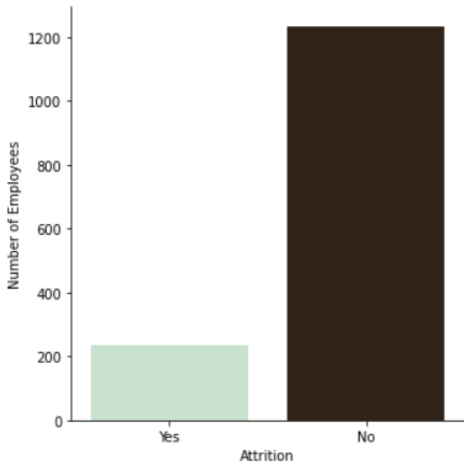
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relations
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	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	...	



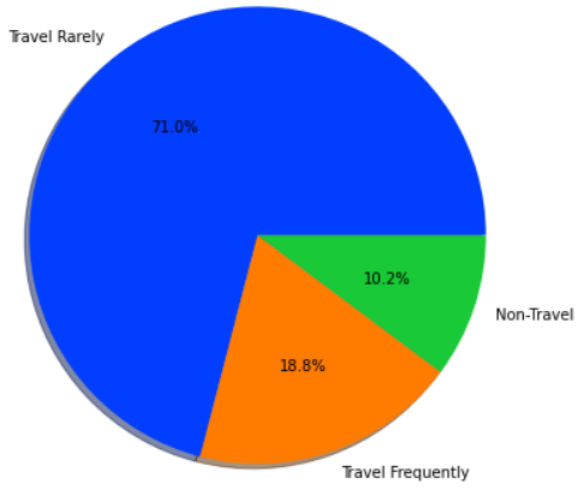
The shows a distribution of null values in the data frame. As you can see there are no null values.

The table above shows the statistical description of the entire data-frame.

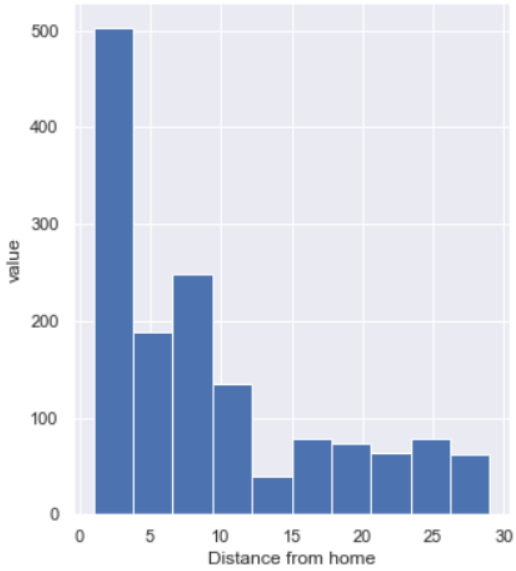
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvemer
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.00000
mean	36.923810	802.485714	9.192517	2.912925	1.000000	1024.865306	2.721769	65.891156	2.72993
std	9.135373	403.509100	8.106864	1.024165	0.000000	602.024335	1.093082	20.329428	0.71156
min	18.000000	102.000000	1.000000	1.000000	1.000000	1.000000	1.000000	30.000000	1.00000
25%	30.000000	465.000000	2.000000	2.000000	1.000000	491.250000	2.000000	48.000000	2.00000
50%	36.000000	802.000000	7.000000	3.000000	1.000000	1020.500000	3.000000	66.000000	3.00000
75%	43.000000	1157.000000	14.000000	4.000000	1.000000	1555.750000	4.000000	83.750000	3.00000
max	60.000000	1499.000000	29.000000	5.000000	1.000000	2068.000000	4.000000	100.000000	4.00000



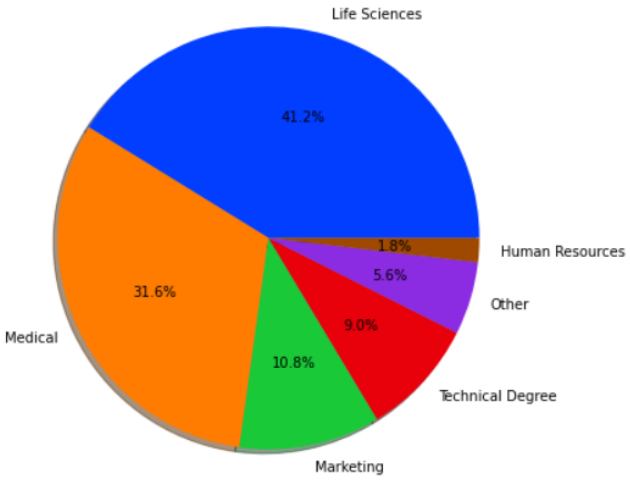
This is a pie-chart displaying different types of Business Travel done by the Employees. This clearly states that There are almost 71% employees who travels rarely.

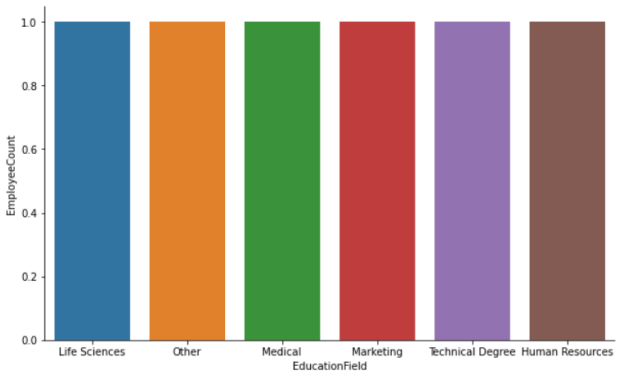


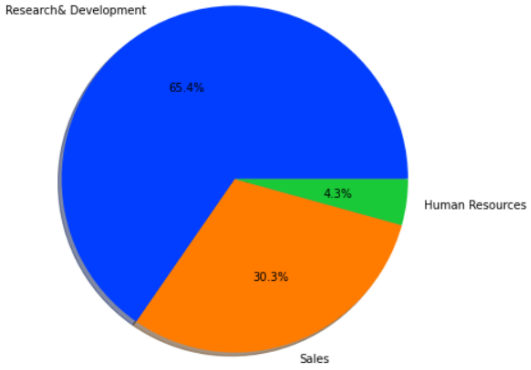
This pie-chart displays the Education fields of each and every employees by certain sections of various fields.



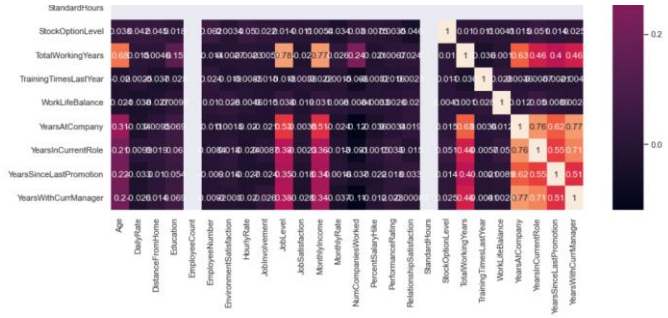
The histograms shows the distribution of the distance between the company and the house of each employee.







This pie chart shows the department in which each employees fall under. For example, 65.4 % of the employees fall under Research & Development.



This is the correlation matrix in the form of heatmap displaying all the non-categorical features relations/correlations with the target variable and among each other.

IV. Model Building

I discovered the model's parameters and trained it for Adaboost using the sklearn general library:

```
stump = tree.DecisionTreeClassifier(max_depth=1)

clf = ensemble.AdaBoostClassifier(base_estimator = stump, algorithm="SAMME", n_estimators=6, random_state=0)
clf = clf.fit(X_train, y_train)
```

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.5)
    loss=tf.keras.losses.MeanSquaredError()
)
```

I used Decision tree model as my base estimator(model).

A. Evaluation Metrics:

i. Accuracy Score:

Accuracy is a metric for classification models that measures the number of predictions that are correct as a percentage of the total number of predictions that are made.

$$Accuracy = \frac{\#\ of\ correct\ predictions}{\#\ of\ total\ predictions}$$

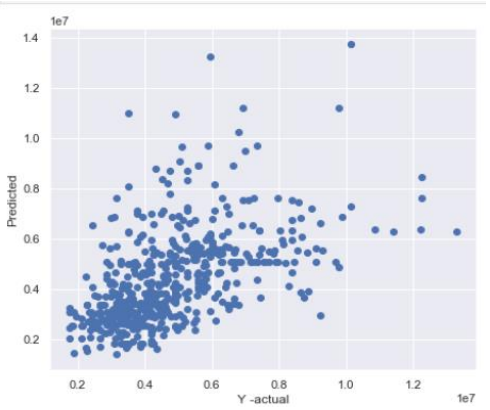
ii. AUC_CURVE:

Basically, ROC curve is a graph that shows the performance of a classification model at all possible thresholds (threshold is a particular value beyond which you say a point belongs to a particular class). The curve is plotted between two parameters

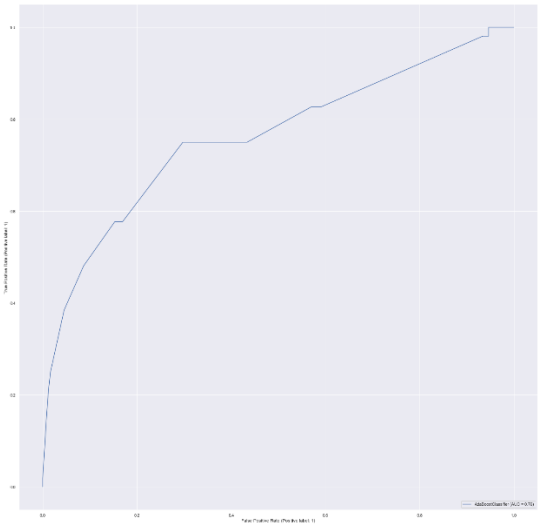
- TRUE POSITIVE RATE
- FALSE POSITIVE RATE

V. Evaluation Results

The below diagram shows how the predicted and the actual values are related:



As you can see the loss function, there is an abrupt change in loss.



Here the auc_curve score is 0.79 hence the model is good.

VI. Conclusion

In this kernel, we have discussed AdaBoost classifier.We have discussed how the base-learners are classified.Then, we move on to discuss the intuition behind AdaBoost classifier.

VII. References

i. Changyou Chen, CSE 574 Lecture Slides(2022).
ii. https://www.kaggle.com/datasets/itssuru/hr-employee-attrition