PROJECT NAME: CLASSIFICATION OF TREATMENT OF MENTAL HEALTH IN THE TECH INDUSTRY

STUDENT: ROBIN KHAOYA WAFULA

SUPERVISOR: SAMUEL WAIYAKI

ABSTRACT

Mental health is a very important aspect of the overall health of any individual. There are different mental illnesses e.g. depression, anxiety and bipolar disorder just to list but a few. These conditions, like any other illness, have the ability to affect all parts of an individual's life, from social to economic and even physical health.

The aim of this project was to use classification models to determine wether or not people in the tech industry should seek treatment for mental illness based on different parameters.

From the study, the AdaBoost Classifier proved to be the most robust model as will be shown in later sections of this report.

ACKNOWLEDGEMENT

I would like to acknowledge every one whose contributions made this project a success.

I would also like to appreciate the online community of developers and data analysts whose tutorials and forum posts helped me tackle the various problems I encountered.

My heartfelt gratitude to my supervisor, Mr. Waiyaki, for his guidance and constructive feedback that enabled me to complete this project in a timely manner, meeting all the requirements of the project scope.

Table of Contents

1.0 CHAPTER ONE

1.1 Introduction

About 14% of the global burden of disease has been attributed to neuropsychiatric disorders, mostly due to the chronically disabling nature of depression and other common mental disorders (Prince, Martin, 2007). With the ever increasing complexity of work, and the pressure to keep up with new technologies, people in the tech industry often suffer from mental health issues. Cases of mental illnesses have been steadily increasing over the years and it is only getting worse as time goes on.

Mental health is affected by various aspects of an individual's life, be it social, professional or personal. Just as there are many causes of mental illness, there are also a variety of treatments and coping mechanisms to improve the quality of life of the individual. Mental disorders also manifest in different ways and in varying degrees, from truancy to aggressive behaviour.

This study focused on mental health in the tech industry and the aim was to classify whether individuals should seek treatment or not based on a number of factors for example, whether their mental health interferes with their physical health or their profession. A variety of machine learning classification models were used to achieve this.

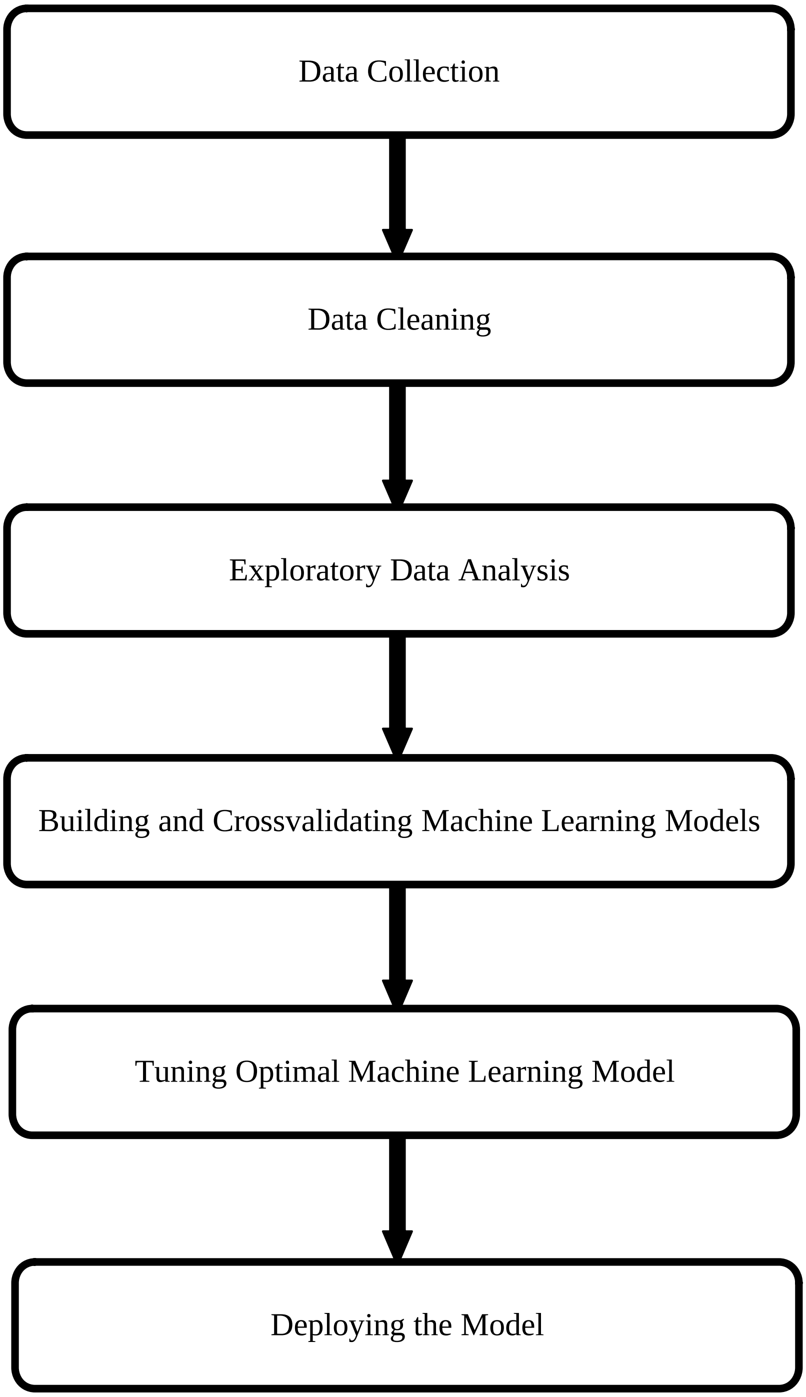
The dataset used contained a variety of columns, like the type of companies the respondents were working in, and whether or not those companies had wellness programs for mental and physical health of their employees.

1.2 Project Scope and Methodology

1.2.1 Project Scope

The aim of this study was to clean the data, perform exploratory data analysis, create machine learning models to be able to classify whether or not individuals should seek treatment.

1.2.2 Methodology


Data Collection
Data Cleaning
Exploratory Data Analysis
Building and Crossvalidating Machine Learning Models
Tuning Optimal Machine Learning Model
Deploying the Model

The above diagram shows the steps and methodology used in this study. Each step will be discused in great detail in the chapters and sections to come.

1.3 Tools, Programs and Requirements

The requirements for this study were ;

- A computer / laptop
- Internet connection

There were various programs and libraries used in this project. They include ;

- An IDE (VS Code was used in this project but any other ide eg Jupyter can be used)
- Anaconda
- Python (version 3.9.2 was used in this study but any version above 2.7 will suffice)
- Python libraries for data science

An installation guide for each of the mentioned programs will be included in the appendix of this document.

2.0 CHAPTER TWO: DATA COLLECTION

2.1 COLLECTION

The dataset used in this study was not collected by me, but was provided freely by Kaggle (www.kaggle.com). The link below leads to the exact dataset used in this study.
www.kaggle.com/osmi/mental-health-in-tech-survey

The data was compiled after a survey was carried out in several countries all over the world. The survey was targeted at people working in tech. Some of the respondent worked in tech companies eg. Google, Facebook etc, some worked in non-tech companies and others were self employed.

The survey also included the work environment of the respondents, the number of collegues, wellness programs and benefits provided by the companies they work for and how their mental health issues have affected their work and families.

3.0 CHAPTER THREE: DATA CLEANING

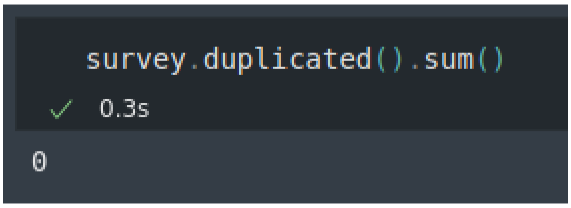
3.1 Introduction

Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data. (Ilyas, I. F., Chu, X. 2019). When data is collected, a lot of errors may be made in the process. Missing values and structural errors are the most common types of errors. A respondent may choose not to fill their age or gender for example, resulting in missing values in the collected data. Structural errors may occur for example, when one respondent fills their gender as 'Female' and another respondent fills 'F' or 'woman'. These three values are different words that all mean the same thing. This is just one of the many ways structural errors may occur.

There is no agreed upon procedure to clean data as the nature of data varies greatly for every dataset. In the following sections, the steps followed in cleaning the data used in this study will be addressed.

3.2 Handling of duplicates

When cleaning the data, no duplicated records were found as shown in the diagram below

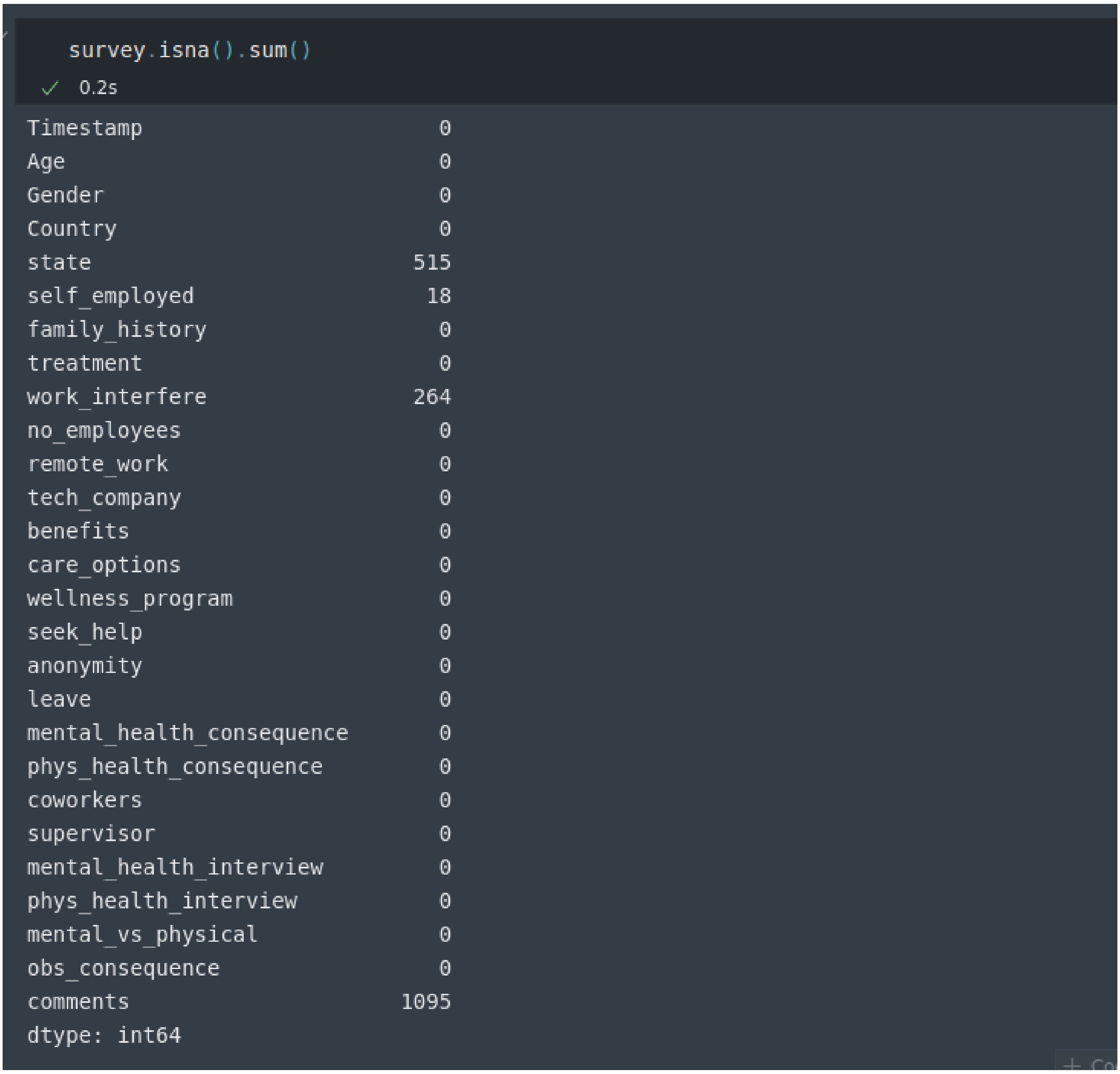


```
survey.duplicated().sum()
✓ 0.3s
0
```

The duplicated() method is a built-in method included in the pandas library.
The line of code shown above returns the total number of duplicate records found in the dataset.
The result is zero meaning no duplicates were found.

3.3 Missing values

There was a significant amount of missing values in the dataset as seen in the image below.



```
survey.isna().sum()
```

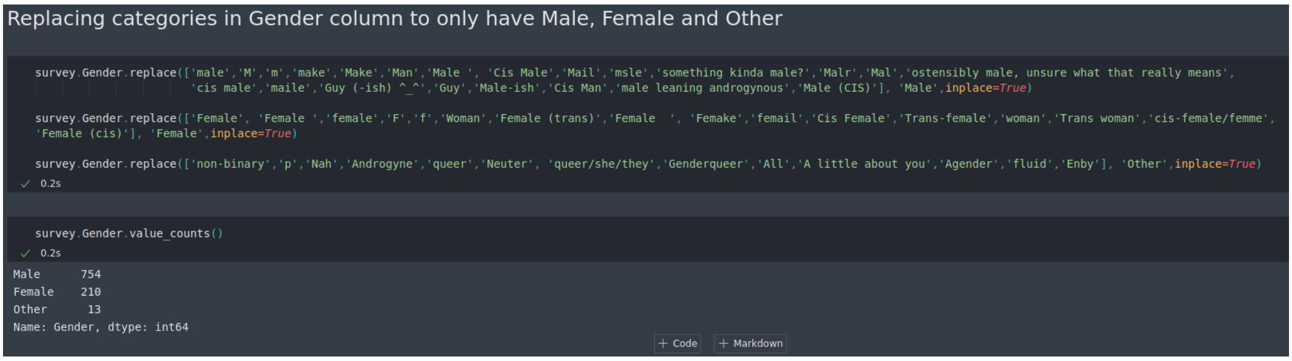
✓ 0.2s

```
Timestamp                        0
Age                              0
Gender                           0
Country                          0
state                          515
self_employed                   18
family_history                   0
treatment                        0
work_interfere                 264
no_employees                     0
remote_work                      0
tech_company                     0
benefits                         0
care_options                     0
wellness_program                 0
seek_help                        0
anonymity                        0
leave                            0
mental_health_consequence        0
phys_health_consequence          0
coworkers                        0
supervisor                       0
mental_health_interview          0
phys_health_interview            0
mental_vs_physical               0
obs_consequence                  0
comments                      1095
dtype: int64
```

Some columns, namely ‘state’ and ‘comments’ had too many missing values that using those entire columns had to be deleted.For the remaining columns, only the rows with missing values were deleted.

3.4 Structural Errors

The ‘Gender’ column was the only column with issues. Different spellings and cases were used for words like male and female. Those values were replaced with one of three categories, ‘Male’, ‘Female’ and ‘Other’, as shown below.



Replacing categories in Gender column to only have Male, Female and Other

```
survey.Gender.replace(['male','M','m','make','Make','Man','Male ', 'Cis Male','Mail','msle','something kinda male?','Malr','Mal','ostensibly male, unsure what that really means',
                        'cis male','maile','Guy (-ish) ^_^','Guy','Male-ish','Cis Man','male leaning androgynous','Male (CIS)'], 'Male',inplace=True)

survey.Gender.replace(['Female', 'Female ','female','F','f','Woman','Female (trans)','Female  ', 'Femake','femail','Cis Female','Trans-female','woman','Trans woman','cis-female/femme',
'Female (cis)'], 'Female',inplace=True)

survey.Gender.replace(['non-binary','p','Nah','Androgyne','queer','Neuter', 'queer/she/they','Genderqueer','All','A little about you','Agender','fluid','Enby'], 'Other',inplace=True)
```

✓ 0.2s

```
survey.Gender.value_counts()
```

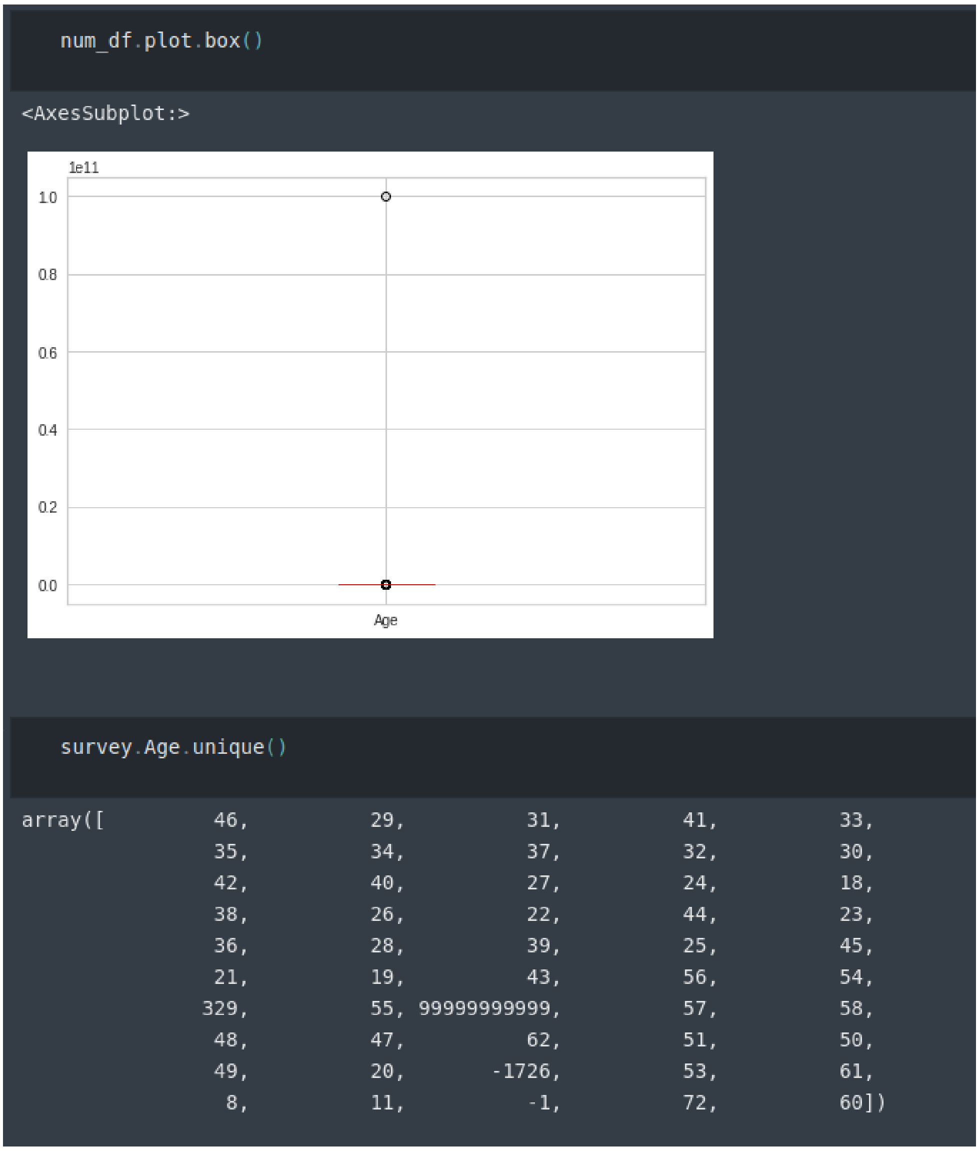
✓ 0.2s

```
Male      754
Female    210
Other      13
Name: Gender, dtype: int64
```

.

3.5 Outliers

Outliers were found in the 'Age' column only as it was the only numeric column in the dataset. A boxplot of age shows the presence of outliers and using the unique() method, we can see some extreme values in the column.



```
num_df.plot.box()
```

```
<AxesSubplot:>
```

```
survey.Age.unique()
```

```
array([         46,          29,          31,          41,          33,
                35,          34,          37,          32,          30,
                42,          40,          27,          24,          18,
                38,          26,          22,          44,          23,
                36,          28,          39,          25,          45,
                21,          19,          43,          56,          54,
               329,          55, 99999999999,          57,          58,
                48,          47,          62,          51,          50,
                49,          20,       -1726,          53,          61,
                 8,          11,          -1,          72,          60])
```

These outliers were removed using the interquartile range method.

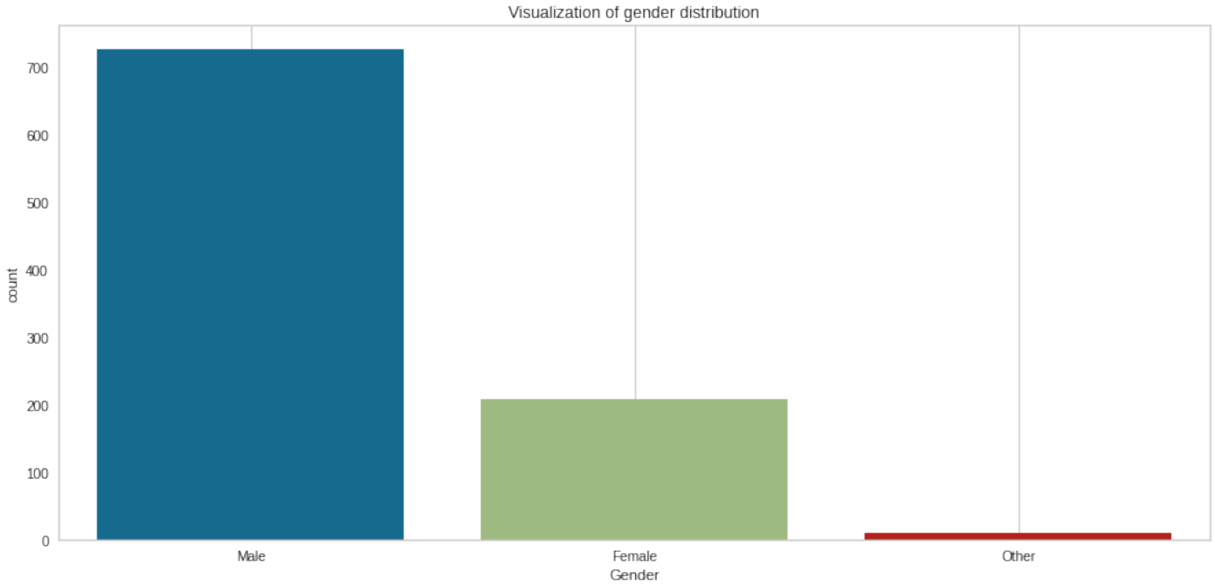
4.0 CHAPTER FOUR: EXPLORATORY DATA ANALYSIS (EDA)

4.1 Introduction

EDA is done not only to familiarize yourself with all the data you have collected, but also to reduce the workload during analysis (Cox, Victoria, 2017). In this section, graphs of different natures will be used to visualize different columns in the dataset and their relationship to each other. Python libraries such as *pandas*, *seaborn* and *matplotlib* will be used in the visualization process.

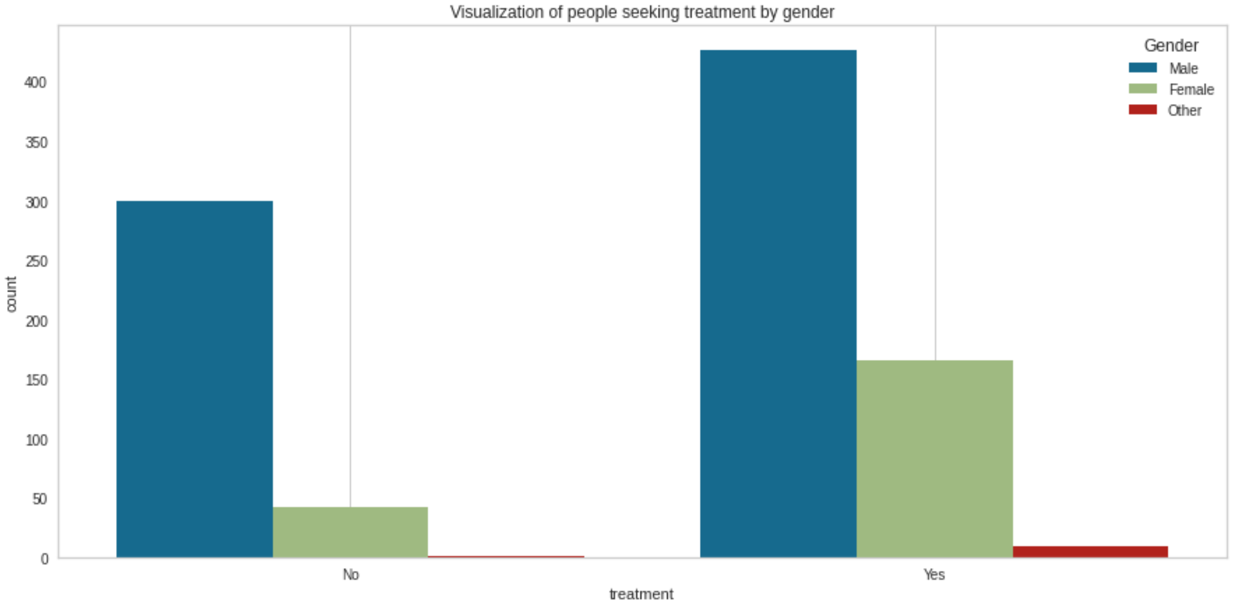
4.2 Visualization

4.2.1 Visualization of the distribution of gender



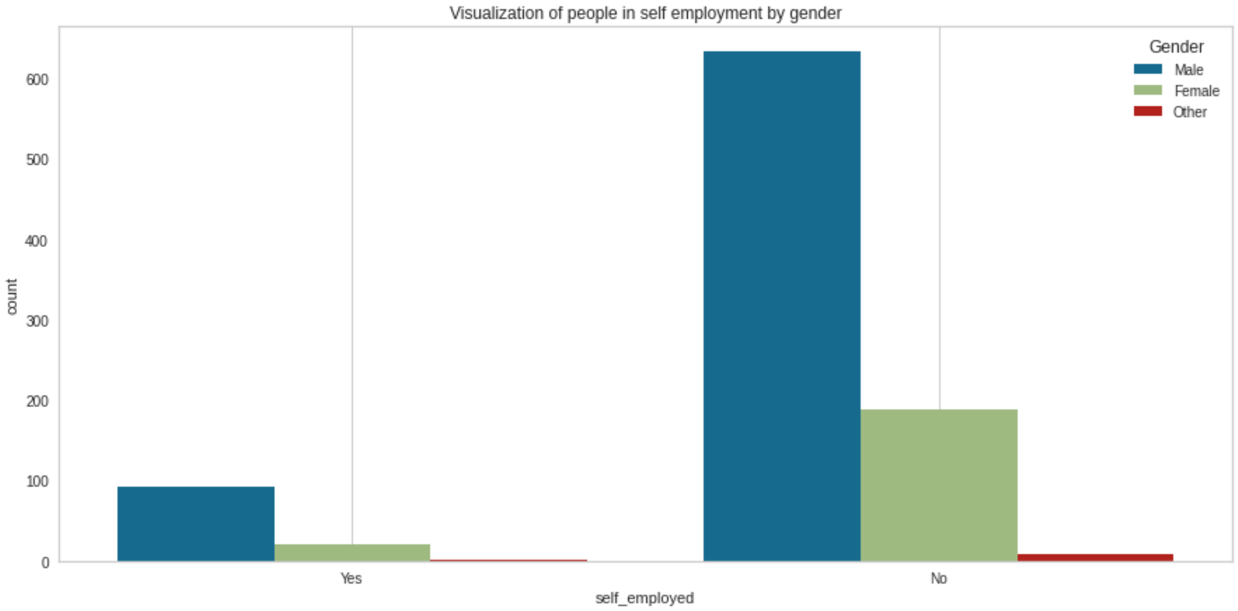
As seen in the image above, the data contained more males than females or other. This may be due to the fact that the tech industry is heavily male dominated.

4.2.2 Visualization of people seeking treatment classified by gender



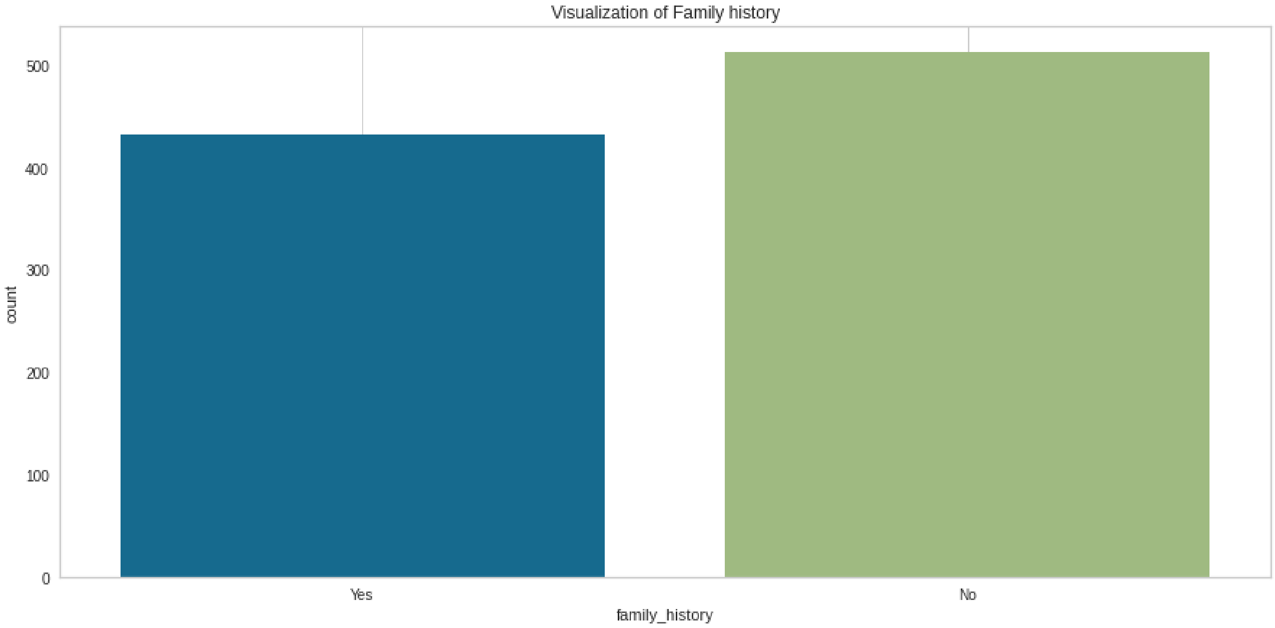
In all gender categories, there are more respondents seeking treatment as oppossed to those who do not. The number of males not seeking treatment is significantly higher than the other gender categories (Female and Other)

4.2.3 Visualization of people in self employment based on gender



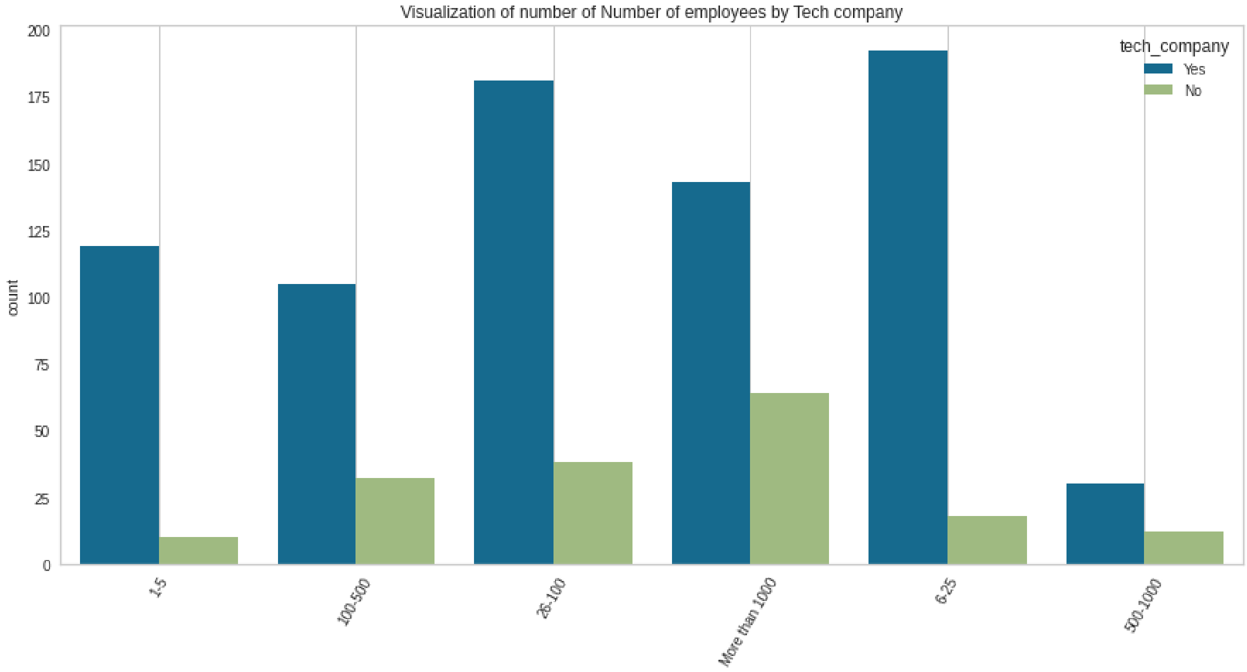
While most people are not self employed, a majority of those who are, are male.

4.2.4 Visualization of mental health issues in family of the respondents



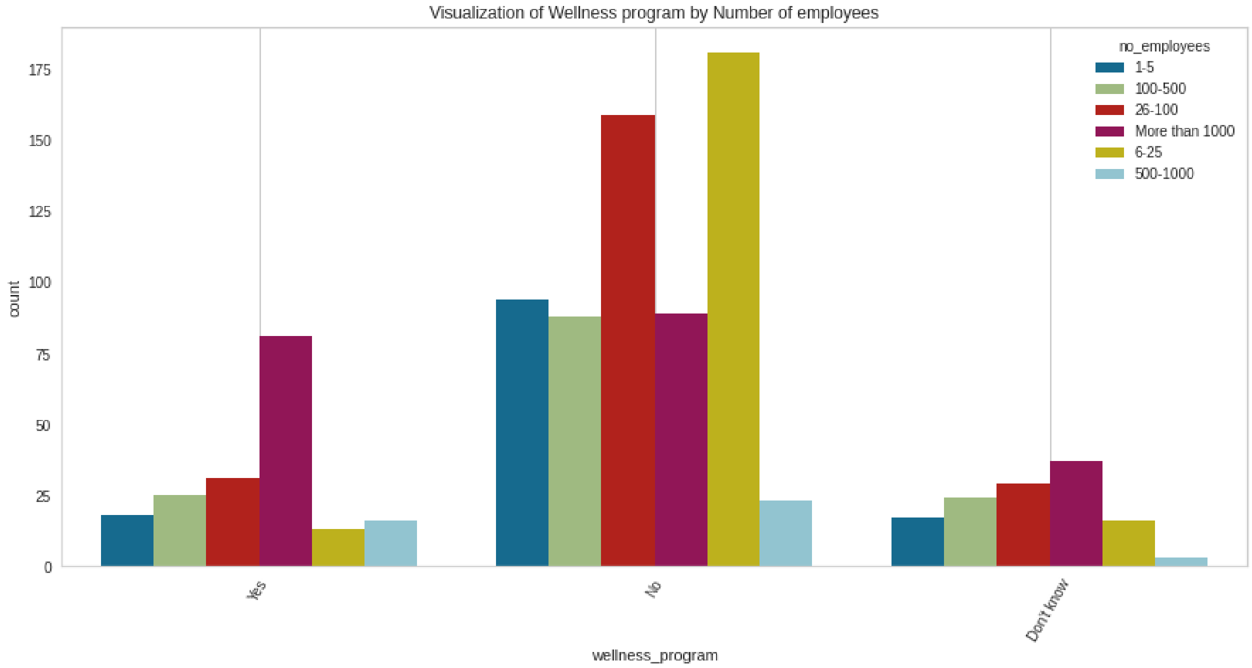
A slim majority of the respondents do not have a history of mental health issues in their families. Some mental health issues can be hereditary thus affect the respondent.

4.2.5 Visualization of number of employees based on type of company



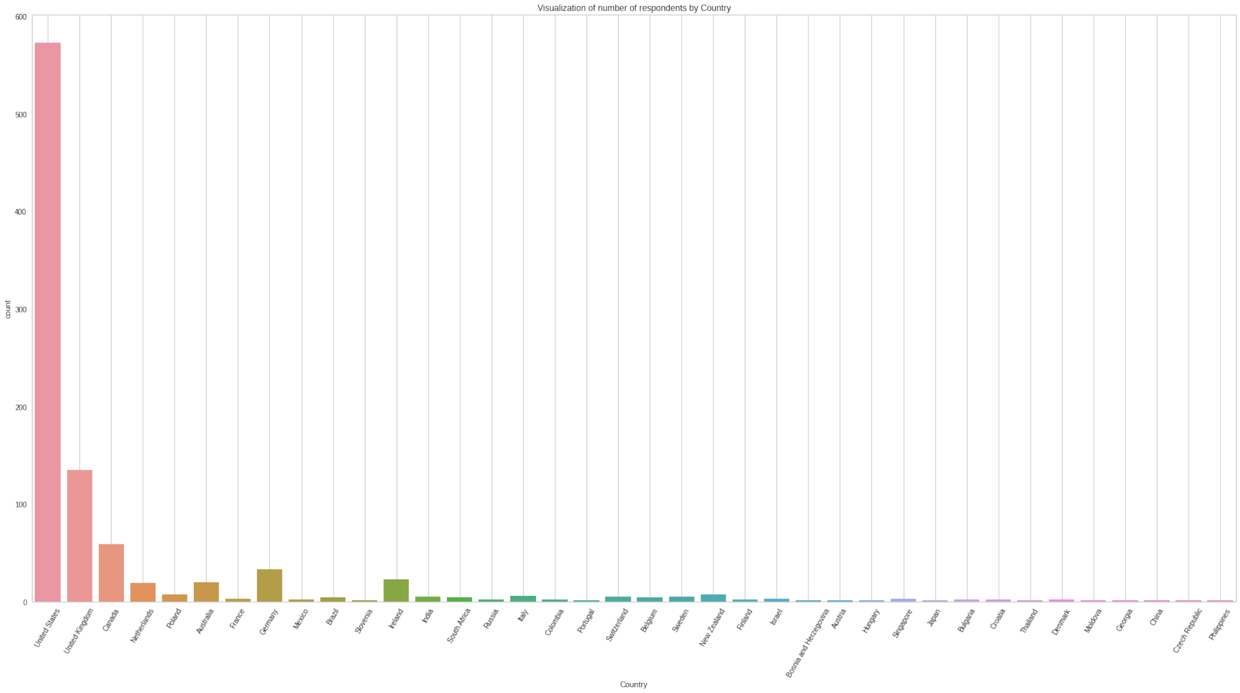
For each category of number of employees, tech companies are the biggest employers.

4.2.6 Visualization of wellness program based on number of employees



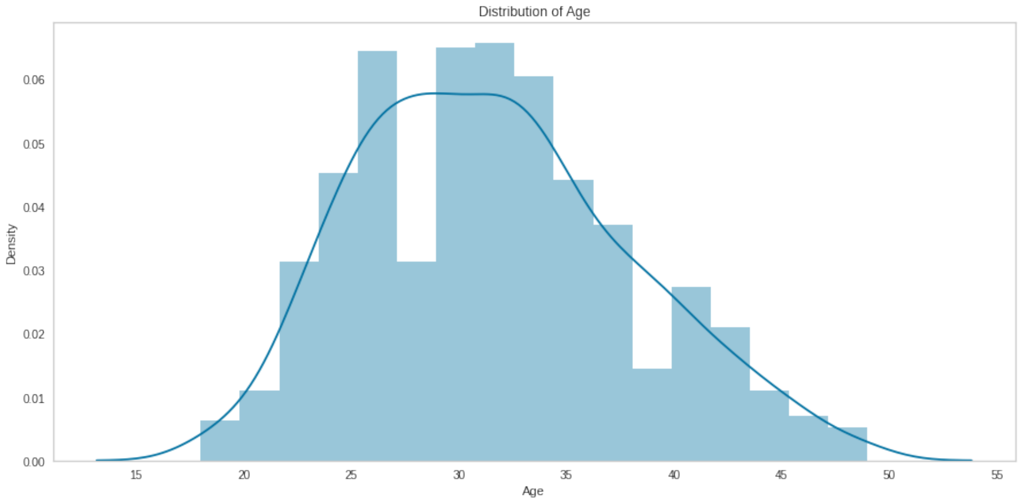
Larger companies (more than 1000 employees) are the majority in companies that offer wellness programs for their employees. This can be attributed to the fact that larger companies have more resources thus can invest more into the health of their employees.

4.2.7 Visualization of number of respondents by country

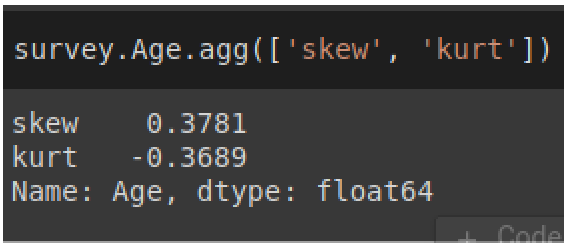


An overwhelming majority of the respondents are from the United States followed by the United Kingdom. This is because a majority of world leading tech companies are based in the US eg Amazon, Facebook, Google etc.

4.2.8 Visualization of age



Age seems to be slightly positively skewed as the right tail is slightly longer than the left tail.



```
survey.Age.agg(['skew', 'kurt'])
```

```
skew     0.3781
kurt    -0.3689
Name: Age, dtype: float64
```

It is also normally distributed because of the skewness as it falls between -0.5 and 0.5.

5.0 CHAPTER FIVE: MACHINE LEARNING (CLASSIFICATION)

5.1 Introduction

The aim of machine learning for this study was to classify whether the respondent should seek treatment for their mental health issues based on a variety of factors included in the dataset. The procedures followed are as listed below;

1. Feature engineering
2. Separation of predictor and dependent variables
3. Encoding the predictor variable
4. Splitting the data into training and testing sets
5. Building the classification models and cross validation
6. Feature selection
7. Scaling the data
8. Building the classification models and cross validation using the scaled data
9. Hyperparameter tuning

5.2 Feature engineering

The timestamp column had to be dropped as it does affect the machine learning classification that will be done and has no significance. The country column also had to be dropped as the work people in tech do is really similar all over the world. This reduced the number of columns from 25 to 23.

5.3 Separation of predictor and dependent variables

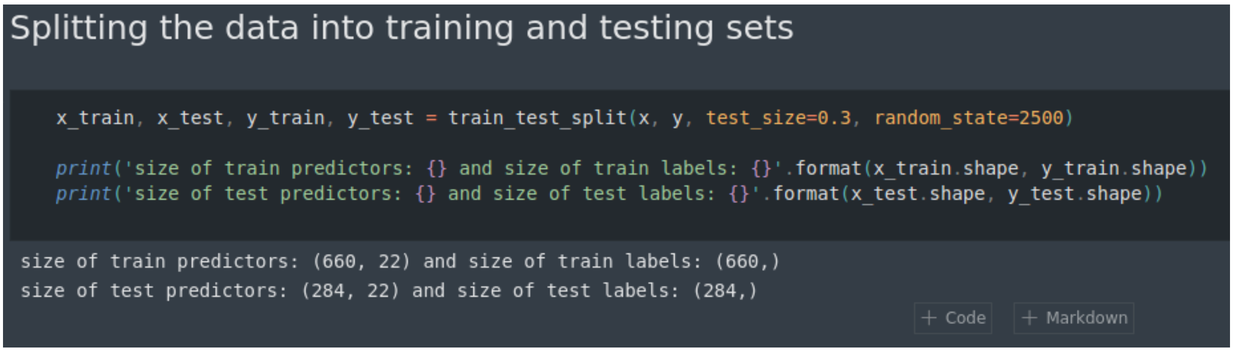
The treatment column is the dependent/response variable while the rest of the columns are the independent/predictor variable. The predictor variable is labeled x while the response variable is y. The x variable has 22 columns.

5.4 Encoding the predictor variable

Machine learning models use data in numeric form. Since the data had categorical columns, it was necessary to convert the data into a format the models could utilise. Encoding was done using LabelEncoder instead of dummies as dummies would have increased the number of columns in the predictor variable. This would have made feature selection very difficult.

5.5 Splitting the data into training and testing sets

The predictor variable was split into two uneven portions, a train set and a test set. The test set was 30% of the entire predictor variable. The code is as shown below



Splitting the data into training and testing sets

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=2500)

print('size of train predictors: {} and size of train labels: {}'.format(x_train.shape, y_train.shape))
print('size of test predictors: {} and size of test labels: {}'.format(x_test.shape, y_test.shape))
```

```
size of train predictors: (660, 22) and size of train labels: (660,)
size of test predictors: (284, 22) and size of test labels: (284,)
```

5.6 Building the classification models and cross validation

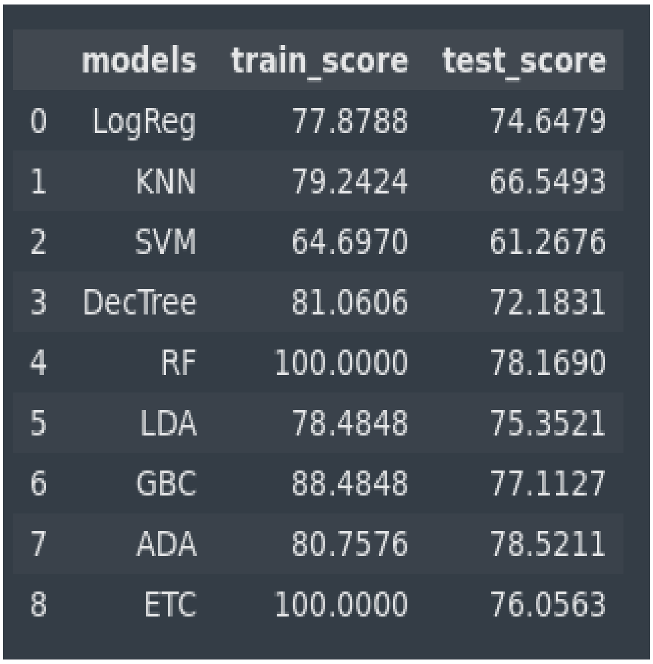
5.6.1 Building the models

The classification models that were used in this study were as listed below;

- Logistic regression classifier
- KNeighbors classifier
- Support vector machine classifier
- Decision tree classifier
- Random forest classifier
- Linear discriminant analysis
- Gradient boost classifier
- Ada boost classifier
- Extra trees classifier

5.6.2 Performance of the models

The accuracy of the models built are highlighted in the image below.

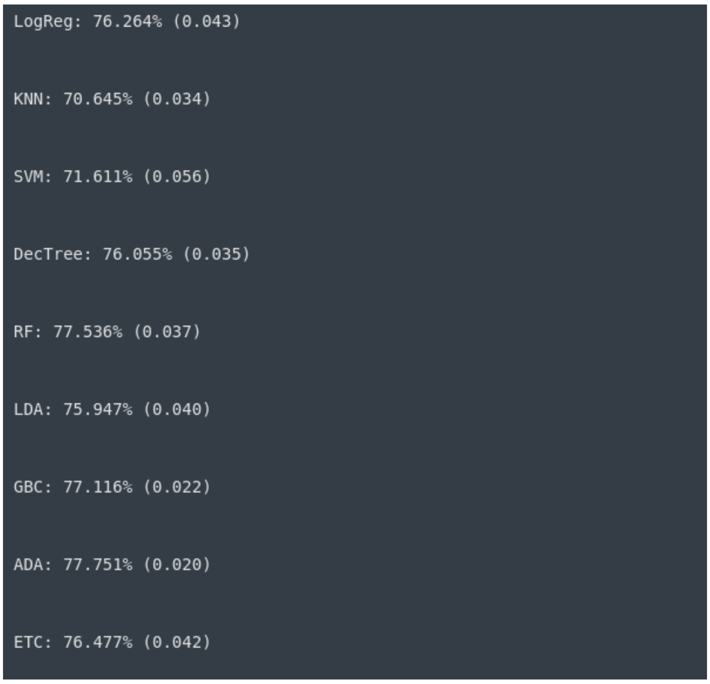


	models	train_score	test_score
0	LogReg	77.8788	74.6479
1	KNN	79.2424	66.5493
2	SVM	64.6970	61.2676
3	DecTree	81.0606	72.1831
4	RF	100.0000	78.1690
5	LDA	78.4848	75.3521
6	GBC	88.4848	77.1127
7	ADA	80.7576	78.5211
8	ETC	100.0000	76.0563

Logistic regression and Ada boost models were the least affected by overfitting. This can be seen as the difference in the accuracy of the train score and the test score was not as huge as in the other models.

5.6.3 Cross Validation

Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models and to tune model parameters (Berrar, D, 2019). The accuracy scores of the models on cross validation are shown in the image below.



```
LogReg: 76.264% (0.043)

KNN: 70.645% (0.034)

SVM: 71.611% (0.056)

DecTree: 76.055% (0.035)

RF: 77.536% (0.037)

LDA: 75.947% (0.040)

GBC: 77.116% (0.022)

ADA: 77.751% (0.020)

ETC: 76.477% (0.042)
```

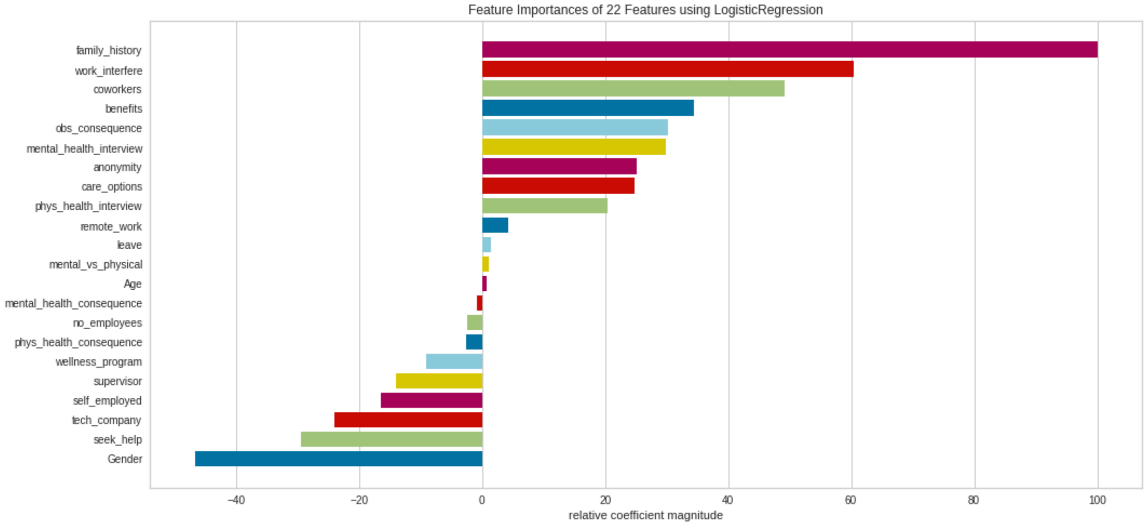
Ada boost classifier had one of the highest scores with the smallest variance of 0.02, proving to be the most robust model.

5.7 Feature selection

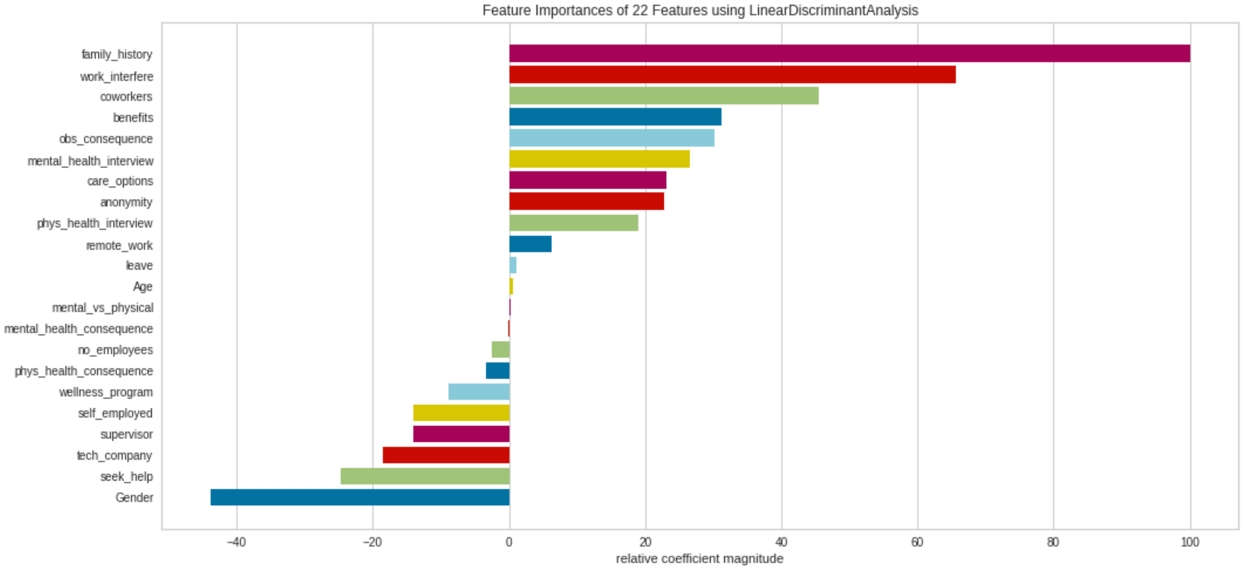
Feature selection was done using a python library called yellowbrick. Each model had measures for feature importance. They will be highlighted below.

5.7.1 Logistic Regression Classifier and Linear Discriminant Analysis Classifier

These two models had the same metrics for feature importance. They both had family history, work interfere and coworkers as the three most important features respectively.



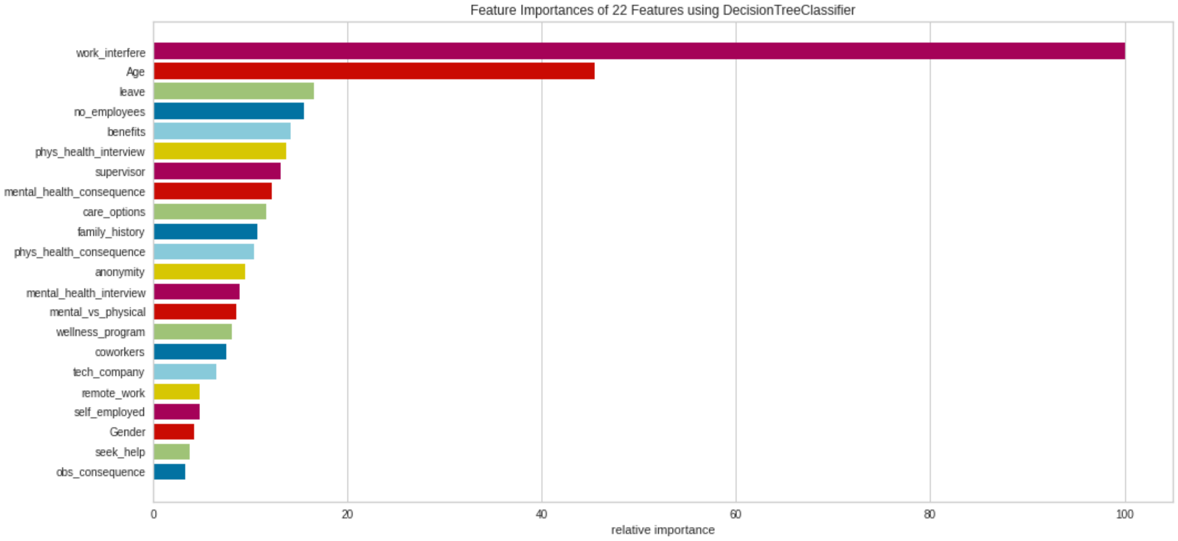
The image above is a plot of feature importance for *Logistic Regression Classifier*.



The image above is a plot of feature importance for *Linear Discriminant Analysis Classifier.* The only difference between the feature importance of these two models was on the placement of *care option, anonymity, mental vs physical* and *age*. These differing feature importances were, however, low ranking in the chart.

5.7.2 Decision Tree Classifier

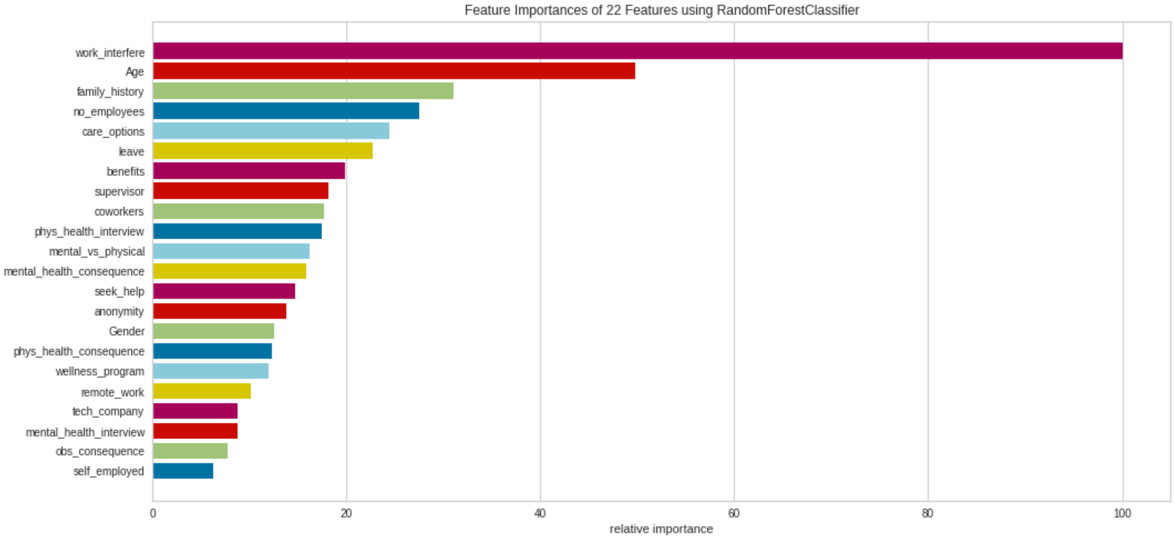
The *Decision Tree Classifier* had the features work interfere, age and leave as its most important features as shown in the image below.



It had completely different metrics for feature importance compared to the previous models.

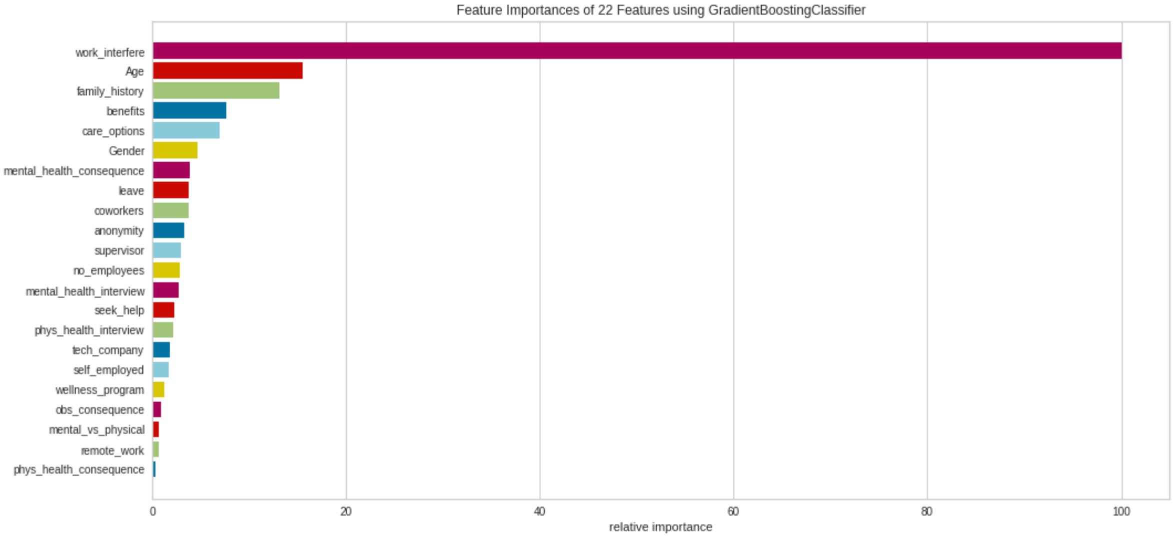
5.7.3 Random Forest Classifier

The Random Forest Classifier had *work interfere, age* and *family history* as its most important features as highlighted below.



5.7.4 Gradient Boosting Classifier

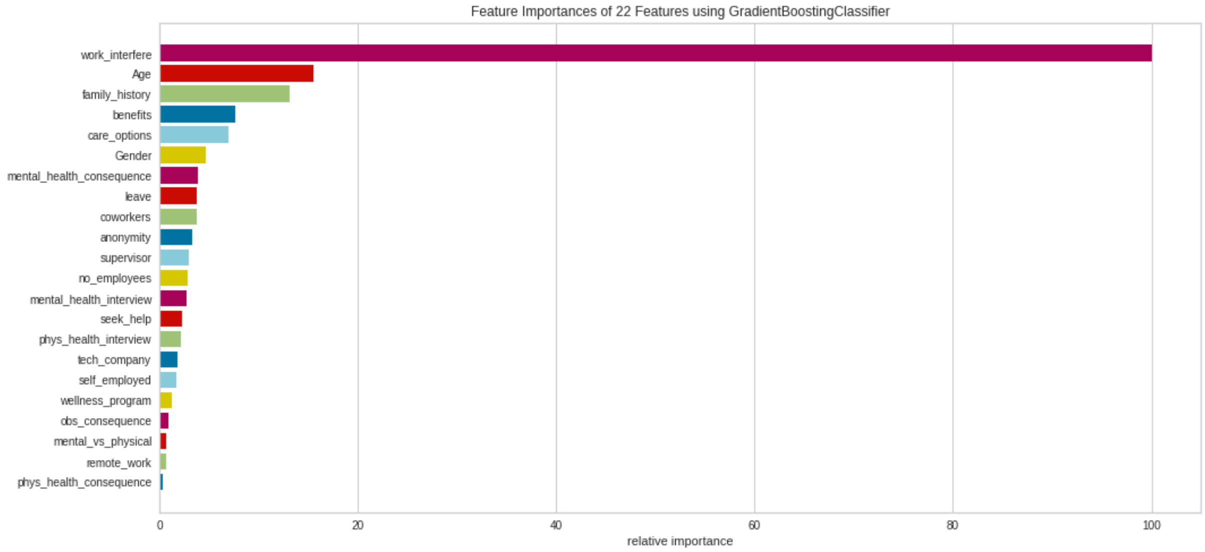
The Gradient Boosting Classifier had the same top three feature importances as Random Forest Classifier. The difference came in after the first three features as shown in the image below.



5.7.5 Extra Trees Classifier

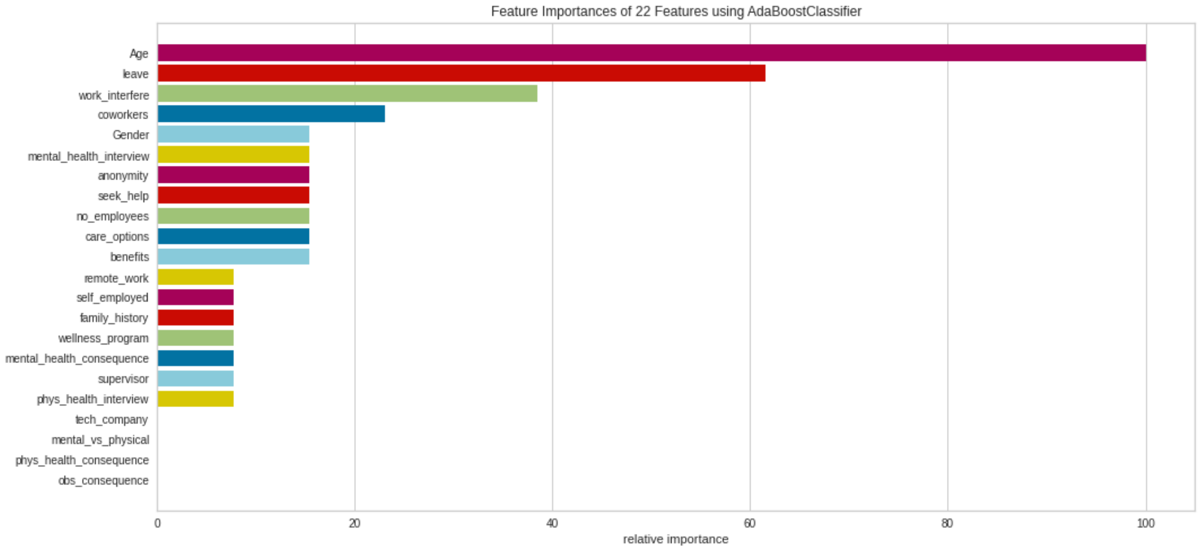
The Extra Trees Classifier also had *work interfere* as its most important feature. However, the features to follow differed from the previous models. *Family history* and *age* came in second and third respectively.

The image below shows a chart for the feature importance for the model.



5.7.6 AdaBoost Classifier

The AdaBoost Classifier was the most unique model when it came to feature selection. It placed more emphasis on *age* as its most important feature. This massive difference from the other models might explain the robust performance of the model.
The image below shows a chart of the feature importance of the model.



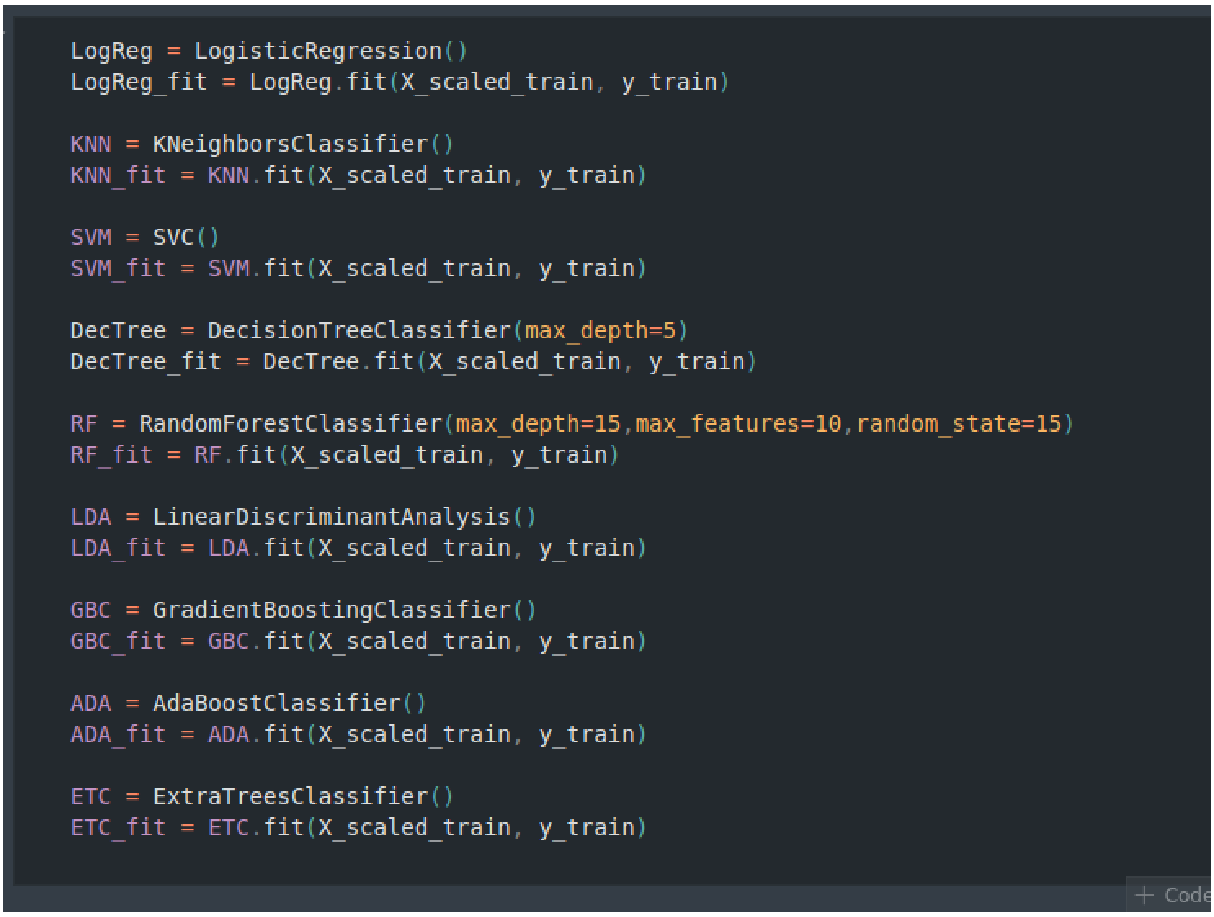
5.8 Scaling the data

Feature scaling is a method used to normalize the range of independent variables or features of data. There are several methods used to scale data. The *MinMaxScaler()* was used it this project. MinMaxScaler() is a simple scaling technique that rescales the data to a range of 0 – 1 or -1 – 1.

5.9 Building the classification models and cross validation using the scaled data

5.9.1 Building the models using the scaled data

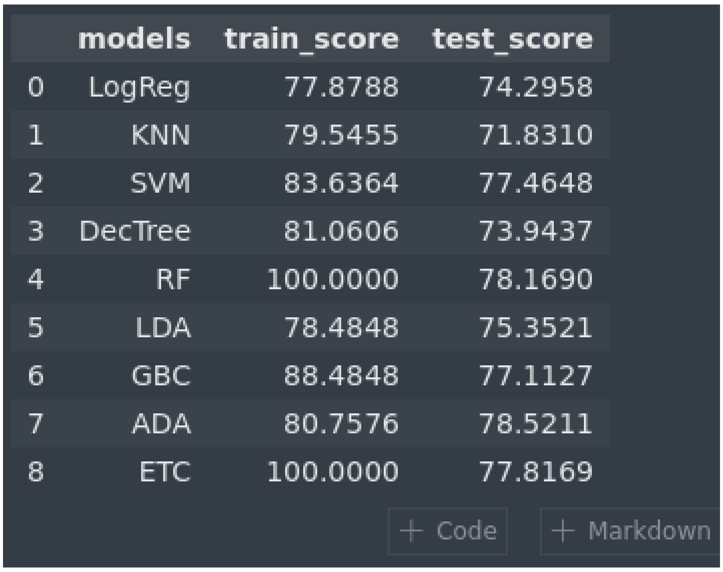
The same models that were built earlier were rebuilt using the scaled data. The scaled predictor variable was named *X_scaled.* The building and fitting is shown in the image below.



```
LogReg = LogisticRegression()
LogReg_fit = LogReg.fit(X_scaled_train, y_train)

KNN = KNeighborsClassifier()
KNN_fit = KNN.fit(X_scaled_train, y_train)

SVM = SVC()
SVM_fit = SVM.fit(X_scaled_train, y_train)

DecTree = DecisionTreeClassifier(max_depth=5)
DecTree_fit = DecTree.fit(X_scaled_train, y_train)

RF = RandomForestClassifier(max_depth=15,max_features=10,random_state=15)
RF_fit = RF.fit(X_scaled_train, y_train)

LDA = LinearDiscriminantAnalysis()
LDA_fit = LDA.fit(X_scaled_train, y_train)

GBC = GradientBoostingClassifier()
GBC_fit = GBC.fit(X_scaled_train, y_train)

ADA = AdaBoostClassifier()
ADA_fit = ADA.fit(X_scaled_train, y_train)

ETC = ExtraTreesClassifier()
ETC_fit = ETC.fit(X_scaled_train, y_train)
```

5.9.2 Performance of the rebuilt data

The performance of the models after scaling the data is as shown in the image below.

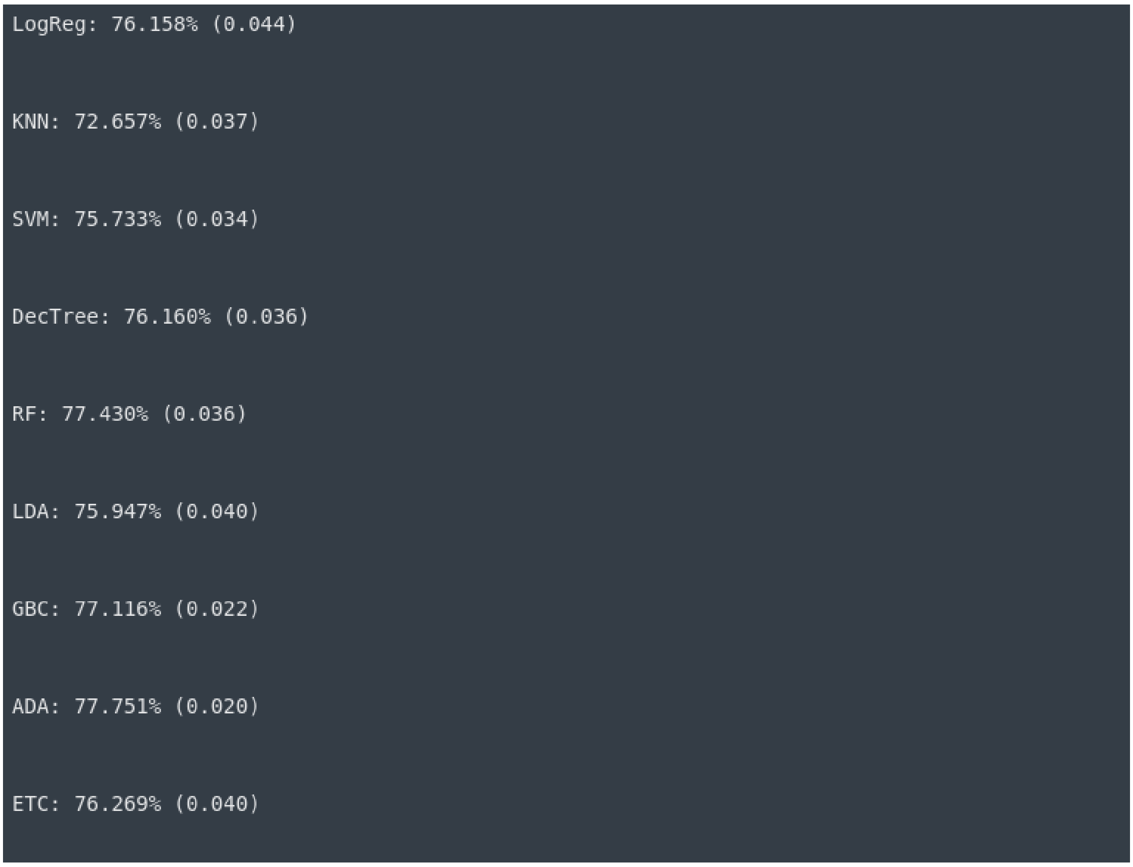


	models	train_score	test_score
0	LogReg	77.8788	74.2958
1	KNN	79.5455	71.8310
2	SVM	83.6364	77.4648
3	DecTree	81.0606	73.9437
4	RF	100.0000	78.1690
5	LDA	78.4848	75.3521
6	GBC	88.4848	77.1127
7	ADA	80.7576	78.5211
8	ETC	100.0000	77.8169

While the performance of most models remain unchanged, the SVM model improved in performance from 64.697% to 83.6364% on the train score and from 61.2676% to 77.4648% on the test score.
The performance of the *KNN, DecTree* and *ETC* models on the test score also improved.

<u>5.9.3 Cross validation</u>

From the results of cross validation, it can be noted that the *KNN* and *SVM* models got a significant improvement after scaling. The *ADA* model is still the most robust model. The results of the cross validation are as shown in the image below.



```
LogReg: 76.158% (0.044)


KNN: 72.657% (0.037)


SVM: 75.733% (0.034)


DecTree: 76.160% (0.036)


RF: 77.430% (0.036)


LDA: 75.947% (0.040)


GBC: 77.116% (0.022)


ADA: 77.751% (0.020)


ETC: 76.269% (0.040)
```

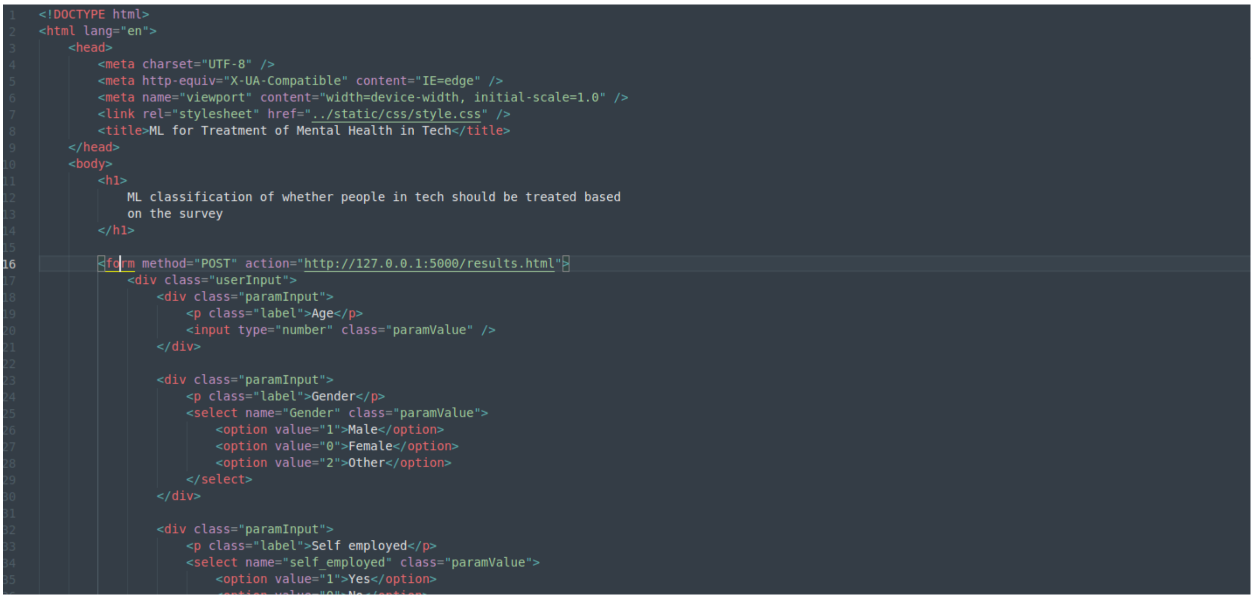
5.10 Hyperparameter Tuning

Hyperparameter tuning is the process of finding the configuration of hyperparameters that results in the best performance. The tuning was done using GridSearchCV() from the sklearn library. It is important to note that the tuning was only applied to the *ADA* model as it was the most robust model built.

The aim was to try and get a little more performance from the optimal model. After carrying out the hyperparameter tuning, the score of the *ADA* model improved from 77% to 79%.

6.0 CHAPTER SIX: DEPLOYING THE MACHINE LEARNING MODEL

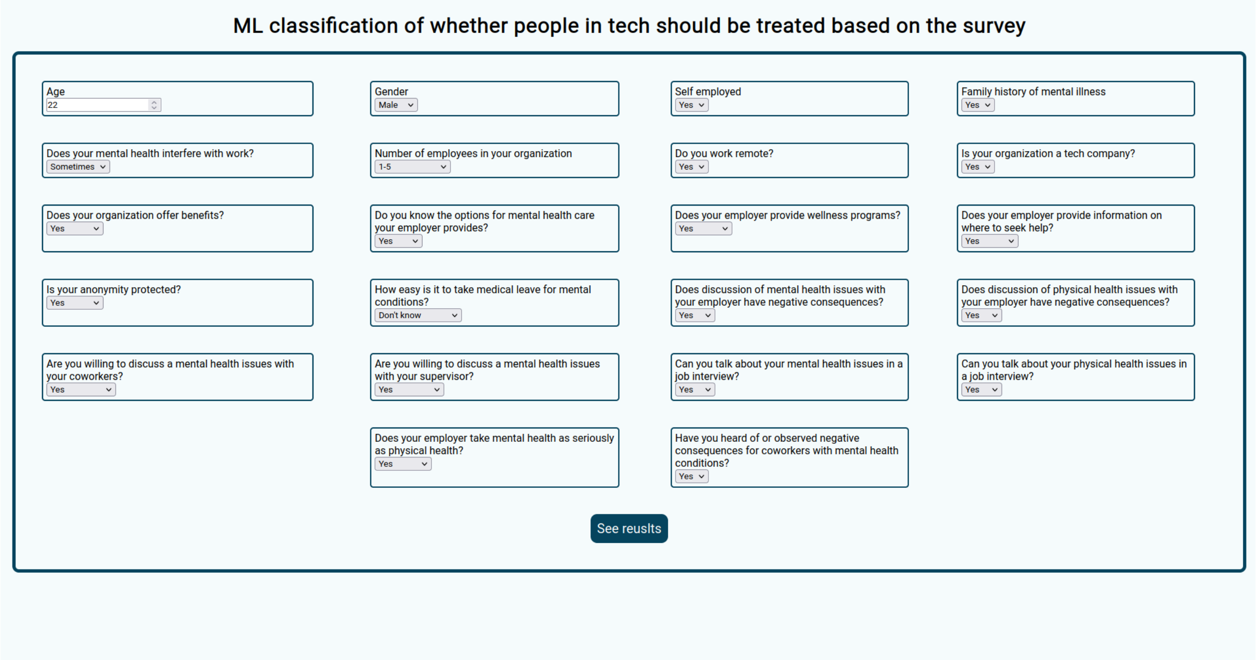
To deploy the model, the flask library was used. Flask was installed using the following command *pip install Flask*
A html page was created to be the front-end of the model. CSS was used to give the pages some basic styling.



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <link rel="stylesheet" href="../static/css/style.css" />
        <title>ML for Treatment of Mental Health in Tech</title>
    </head>
    <body>
        <h1>
            ML classification of whether people in tech should be treated based
            on the survey
        </h1>

        <form method="POST" action="http://127.0.0.1:5000/results.html">
            <div class="userInput">
                <div class="paramInput">
                    <p class="label">Age</p>
                    <input type="number" class="paramValue" />
                </div>

                <div class="paramInput">
                    <p class="label">Gender</p>
                    <select name="Gender" class="paramValue">
                        <option value="1">Male</option>
                        <option value="0">Female</option>
                        <option value="2">Other</option>
                    </select>
                </div>

                <div class="paramInput">
                    <p class="label">Self employed</p>
                    <select name="self_employed" class="paramValue">
                        <option value="1">Yes</option>
```

A form was created within the html page and the values of the encoded categorical data was used for the *value* attribute of html dropdown inputs.

A picture of the completed front-end is as seen below.



ML classification of whether people in tech should be treated based on the survey

Age 22

Gender Male

Self employed Yes

Family history of mental illness Yes

Does your mental health interfere with work? Sometimes

Number of employees in your organization 1-5

Do you work remote? Yes

Is your organization a tech company? Yes

Does your organization offer benefits? Yes

Do you know the options for mental health care your employer provides? Yes

Does your employer provide wellness programs? Yes

Does your employer provide information on where to seek help? Yes

Is your anonymity protected? Yes

How easy is it to take medical leave for mental conditions? Don't know

Does discussion of mental health issues with your employer have negative consequences? Yes

Does discussion of physical health issues with your employer have negative consequences? Yes

Are you willing to discuss a mental health issues with your coworkers? Yes

Are you willing to discuss a mental health issues with your supervisor? Yes

Can you talk about your mental health issues in a job interview? Yes

Can you talk about your physical health issues in a job interview? Yes

Does your employer take mental health as seriously as physical health? Yes

Have you heard of or observed negative consequences for coworkers with mental health conditions? Yes

7.0 CHAPTER SEVEN: CRITICAL APPRAISAL

7.1 Conclusion

The project was an overall success. It established that machine learning algorithms can be used to classify whether individuals require treatment for their mental health conditions. This can help improve the lives of not only people in tech, but everyone at large.

7.2 Challenges Encountered

1. The results of the study were lower than expected.
2. The size of the dataset was small.
3. Cleaning the data further decreased its size.
4. Working with technologies I never used before eg. flask, proved to be challenging.

7.3 Lessons Learned

Though machine learning algorithms can be used to classify whether people need treatment, it does not, however, represent all the aspects that affect the mental health of an individual. The mental health of different individuals are affected by varying conditions and to differing degrees. The machine learning algorithms can only be used in specific use cases and not everywhere.

7.4 Recommendations

Though the project was a success, the performance of the models was lower than what would have been prefered. A significantly larger dataset would have led to much better results.

REFERENCES

1. Prince, M., Patel, V., Saxena, S., Maj, M., Maselko, J., Phillips, M. R., & Rahman, A. (2007). *No health without mental health.* The lancet, 370(9590), 859-877.

2. Ilyas, I. F., Chu, X. (2019). *Data Cleaning.* United States: Association for Computing Machinery and Morgan & Claypool Publishers.

3. Cox, V. (2017). *Exploratory data analysis. In Translating Statistics to Make Decisions* (pp. 47-74). Apress, Berkeley, CA.

4. Berrar, D. (2019). *Cross-Validation.*

APPENDICES

Appendix A: Installation of software and tools

The tools that were highlighted in chapter one section three will have their installation guides showcased here to make it easier to follow along with the project.

Installation of python

For Windows

Visit the official python website (www.python.org) and download the latest python installer for Windows and run it.

For Mac

Visit the official python website (www.python.org) and download the latest python installer for Mac and run it.

For Linux

Because there are a variety of Linux systems, each with different installation processes, a link will be included containing installation guides for the most common Linux systems. (https://www.csestack.org/install-python-on-linux/)

Installation of anaconda

For Windows

Visit the official python website (https://www.anaconda.com/products/individual) and download the anaconda installer for Windows and run it.

For Mac

Visit the official python website (https://www.anaconda.com/products/individual) and download the anaconda installer for Mac and run it.

For Linux

Visit the official python website (https://www.anaconda.com/products/individual) and download the anaconda installer for Linux and run it.

Installation of Visual Studio Code (VS Code)

For Windows

Visit the official python website (https://code.visualstudio.com/Download) and download the VS Code installer for Windows and run it.

For Mac

Visit the official python website (https://code.visualstudio.com/Download) and download the VS Code installer for Mac and run it.

For Linux

Visit the official python website (https://code.visualstudio.com/Download) and download the VS Code installer for Linux and run it.

Installation of python libraries

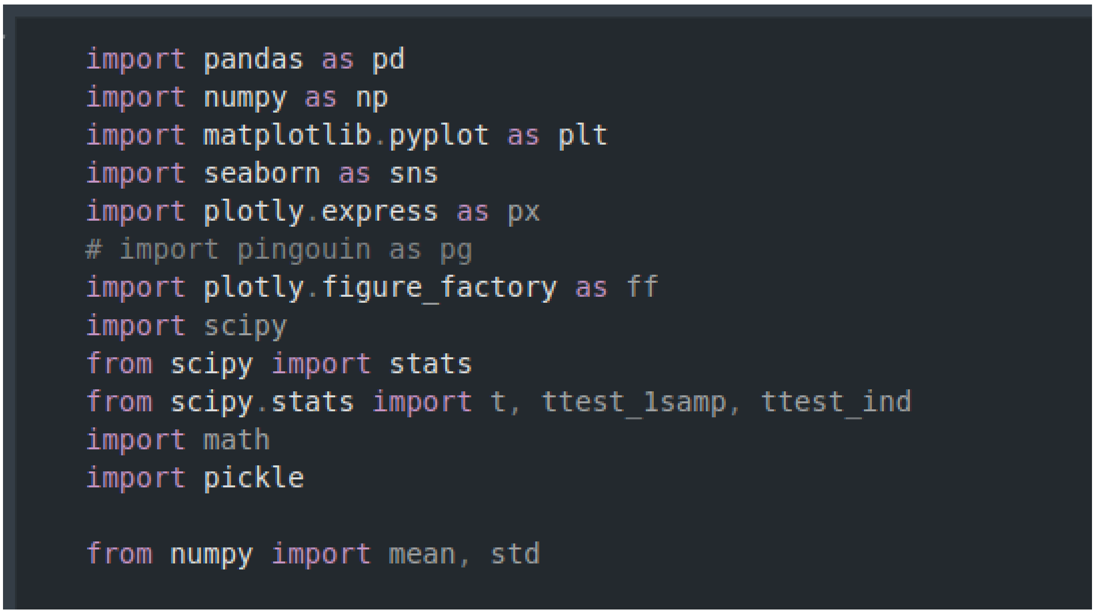
Once python is installed, the *pip* package manager can be used to install the libraries you need. You can run it in your terminal/cmd in the format of

```
pip install (library-name)
```

For example,

```
pip install pandas
```

After installation of the libraries, they can be imported into your python notebook as shown in the image below.



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
# import pingouin as pg
import plotly.figure_factory as ff
import scipy
from scipy import stats
from scipy.stats import t, ttest_1samp, ttest_ind
import math
import pickle

from numpy import mean, std
```