DETAILING THE ANALAYSIS OF MENTAL HEALTH IN THE IT WORKPLACE BASED ON A SURVEY OF EMPLOYEE ATTITUDES

Leroy Maswaure 1710240005

Akwolu Mbamalu Festus 1712150010

Aung Kyaw Moe 180110007

ITE351 PROGRAMMING FOR DATA SCIENCE AJ. Chaky

Introduction

The data was from Kaggle but the main source was from Open Sourcing Mental Illness LTD, (OSMI Mental Health) in tech survey in 2016.

Why we used this data

[We were interested in examining how mental health is viewed within the technology or IT workplace, and the prevalence of mental health disorders within the tech industry.]

To predict treatment of employees because it can affect their work and moral

1.Are Males, females or trans (which gender is) more likely to seek mental health treatment?

2.Family history cause participants to seek mental treatment?

3.Can work interferences cause participants to seek mental treatment?

4. Predicting if employees should get proper treatment based on mental treatment history and work performance , age and gender.

I. List of Variables

- Timestamp
- Age
- Gender
- Country
- state: If you live in the United States, which state or territory do you live in?
- self_employed: Are you self-employed?
- family_history: Do you have a family history of mental illness?
- treatment: Have you sought treatment for a mental health condition?
- work_interfere: If you have a mental health condition, do you feel that it interferes with your work?
- no_employees: How many employees does your company or organization have?
- remote_work: Do you work remotely (outside of an office) at least 50% of the time?
- tech_company: Is your employer primarily a tech company/organization?
- benefits: Does your employer provide mental health benefits?
- care_options: Do you know the options for mental health care your employer provides?
- wellness_program: Has your employer ever discussed mental health as part of an employee wellness program?

- seek_help: Does your employer provide resources to learn more about mental health issues and how to seek help?
- anonymity: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?
- leave: How easy is it for you to take medical leave for a mental health condition?
- mental_health_consequence: Do you think that discussing a mental health issue with your employer would have negative consequences?
- phys_health_consequence: Do you think that discussing a physical health issue with your employer would have negative consequences?
- coworkers: Would you be willing to discuss a mental health issue with your coworkers?
- supervisor: Would you be willing to discuss a mental health issue with your direct supervisor(s)?
- mental_health_interview: Would you bring up a mental health issue with a potential employer in an interview?
- phys_health_interview: Would you bring up a physical health issue with a potential employer in an interview?
- mental_vs_physical: Do you feel that your employer takes mental health as seriously as physical health?
- obs_consequence: Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?
- comments: Any additional notes or comments

II. What we observed in the data

We observed that the variables are questions and comments from the sample group that was used from the compilers of the data. Variables such as age had a number of values that were skew because we noticed age groups such as -1 and 99.9999 which would be impossible because the age groups who participated in the survey where from the age of 18-65. It is also important to note that the participants of the survey are from different countries, this would help us get varied responses.

III. Methods of analysis

We cleaned the data in variables such as gender because we found out that participants listed their gender differently, so we categorized them into 3 genders which are Male, Female and

Trans or Other. Through the use of visualizations of a histogram we managed to find out that out of all 3 genders Male attitudes towards mental health are positive because more males would (seek help) for mental health issues other than females and trans.

We also looked at the participants and (work interference), and we found out that most participants stated that sometimes mental health issues can interfere with work. So, which gives us an assumption that participants who answered sometimes face some form of mental issues occasionally.

Also, by the value count we found out that out of all the 1242 participants 628 have sought for treatment for mental health issues and 614 have not. We also analyzed the correlation of our variables using the correlation matrix for our models.

IV. Algorithms which we used and why and their results

- Decision Tree
- Random Forest
- Naive Bayes

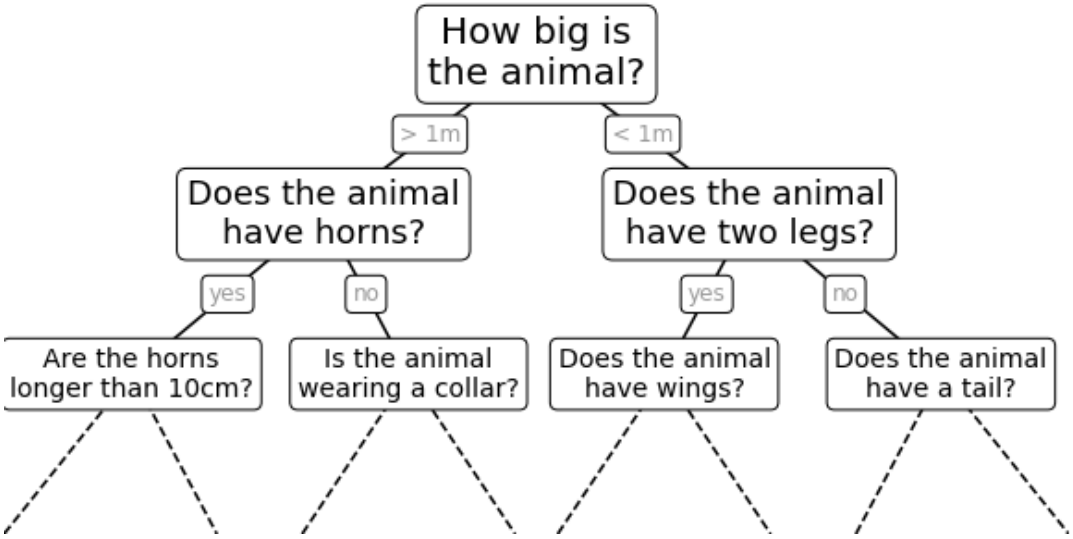
Machine Learning Algorithm We use

1.Decision Tree

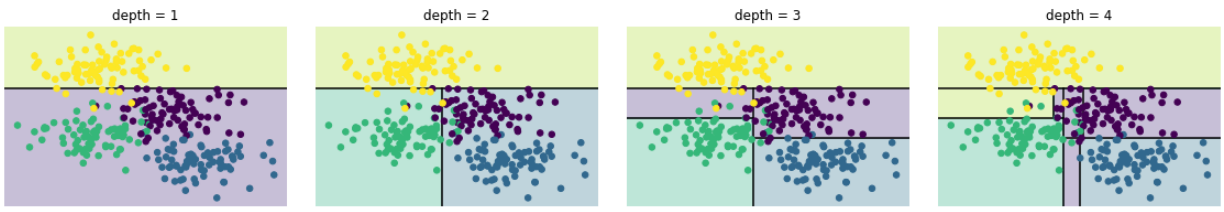
Decision Tree is one of the most common algorithm in classification. Basically it solving a machine learning problem by building a tree from data-set attributes. Each attribute becomes a question node which evaluate the attributes value of data-set and divide the data-set into two data-subsets. These two data-subsets are true and false answers of the question that is asked by the question node. These two data-subsets become the input data-sets for another two question nodes. The first most node is called the root node. This action repeats and more nodes are involved until we get the final pure answer. The binary splitting makes this extremely efficient: in a well-constructed tree, each question will cut the number of options by approximately half, very quickly narrowing the options even among a large number of classes.

Efficiency is import that we have to find the most suitable question for each node to divide the data-set into two child data-subsets. The uncertainty of the data-set is that probability of classes (our final answer). That uncertainty is called Gini impurity. To find the most suitable question for that certain node, we need to find the question with reduce most uncertainty and use that as a node filter.

First, Algorithm starts to find the first question for the node by iterating all different rows' values as question and find the most information gain(information gain = input data-set uncertainty – output uncertainty). If there is no question to ask, the output data-subset will become the answer called leaf. Then it gets to the next column of the data-set and performs the same task. And to steps together build a complete model. Since we are just building one tree as a model for our algorithm there is a lot of over fitting in some cases. This can be solved by using a method called Bagging in which it makes use of an ensemble (a grab bag, perhaps) of parallel estimators, each of which over-fits the data, and averages the results to find a better classification. An ensemble of randomized decision tree is known as a Random Forest.




```
model_1 = DecisionTreeClassifier(criterion='entropy', max_depth= 3, max_features=6, min_samples_leaf= 7, min_samples_split=8)
model_1.fit(X_train,y_train)
y_predict = model_1.predict(X_test)
evalClassModel(model_1,y_test,y_predict)
```

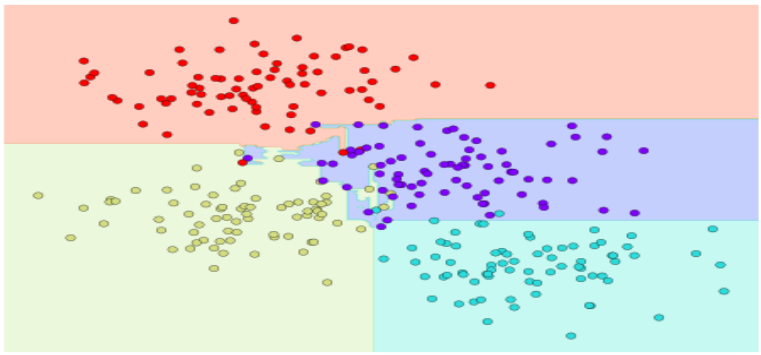


Decision tree in depth and final over fitting

```
: from sklearn.tree import DecisionTreeClassifier
  from sklearn.ensemble import BaggingClassifier

  tree = DecisionTreeClassifier()
  bag = BaggingClassifier(tree, n_estimators=100, max_samples=0.8,
                          random_state=1)

  bag.fit(X, y)
  visualize_classifier(bag, X, y)
```



2.Random Forest

Random Forest is another optimized form of Decision Tree. But in Random Forest instead of making one tree, we try to grow as many trees as possible with random features. Those become models and we take the majority accuracy as the final.
Random forests are a powerful method with several advantages:

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees. In addition, both tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities.
- The multiple trees allow for a probabilistic classification: a majority vote among estimators gives an estimate of the probability (accessed in Sci kit-Learn with the `predict_proba()` method).
- The non-parametric model is extremely flexible, and can thus perform well on tasks that are under-fit by other estimators.

A primary disadvantage of random forests is that the results are not easily interpret able: that is, if you would like to draw conclusions about the meaning of the classification model, random forests may not be the best choice.

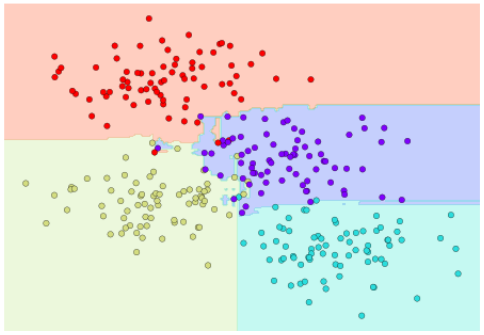
Our Random Forest and Its parameters

```
model_2 = DecisionTreeClassifier(criterion='entropy', max_depth= 3, max_features=6, min_samples_leaf= 7, min_samples_split=8)
model_2.fit(X_train,y_train)
y_predict = model_2.predict(X_test)
evalClassModel(model_2,y_test,y_predict)
```

Example of best fit Random Forest

```
|: from sklearn.ensemble import RandomForestClassifier

  model = RandomForestClassifier(n_estimators=100, random_state=0)
  visualize_classifier(model, X, y);
```



3.Naive Bayes

Naive Bayes classification is where we make use of naive bayes theorem as the algorithm. Naive bayes theorem states that we can find the probability of a given scenario by this equation-

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

If we are trying to decide between two labels

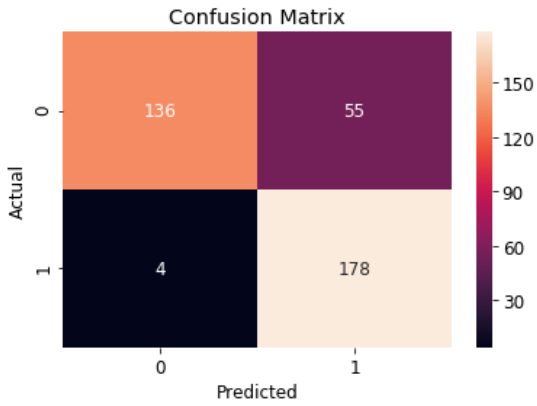
$$\frac{P(L_1 \mid \text{features})}{P(L_2 \mid \text{features})} = \frac{P(\text{features} \mid L_1)}{P(\text{features} \mid L_2)} \frac{P(L_1)}{P(L_2)}$$

Evaluating A Model

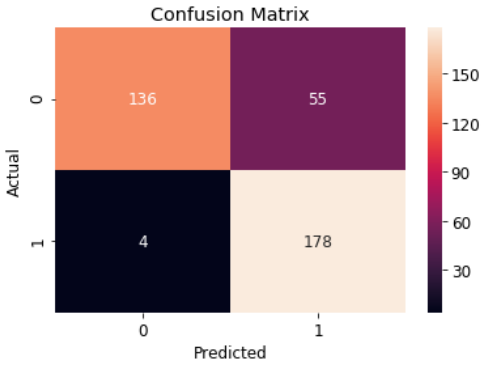
After all algorithms of find the best model for the data-set, it has not completed without measuring the accuracy. One good way of doing it is splitting the data-set into two subsets for training and testing. We use training subset to train our model and use the testing subset's features for guessing the answer, in the other way testing. The we check the match with testing subset's labels and calculate the accuracy of the model. This calculation involves confusion matrix-
Basically we draw a relation table between numbers of time that we make a right guess and number of time that we makes a bad move. There are four ways of we making bad and good decisions. First one is called true positive means that we make a right guess that is being a positive. Another true negative means that we also make a right guess that is being a negative. And for third we got false positive means that we made a bad guess that is being positive but it turns out to be negative. And finally we got true negative means that we made a bad guess again that is being negative but it turns out to be positive.

		Actual class	
		Cat	**Non-cat**
Predicted class	**Cat**	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives

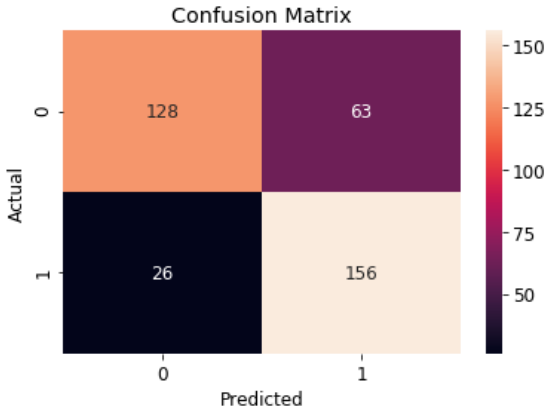
Our Decision Tree Matrix
with the accuracy of 84.2%



Our Random Forest
with the accuracy of 84.2%



Our Naive Bayes
with the accuracy of 76.1%

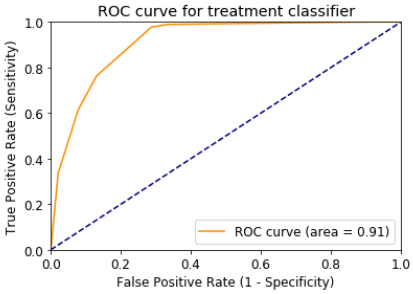


accuracy formula

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

We got AUC for threshold testing and model qualifying.
For Decision Tree

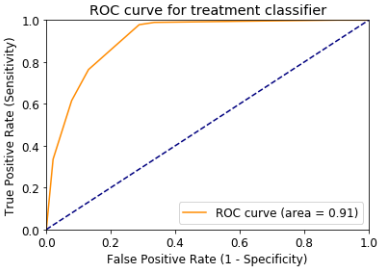
```
Classification Accuracy: 0.8418230563002681
Classification Error: 0.1581769436997319
False Positive Rate: 0.2879581151832461
Precision: 0.7639484978540773
AUC Score: 0.8450319314193661
Cross-validated AUC: 0.8874453342063073
```



For

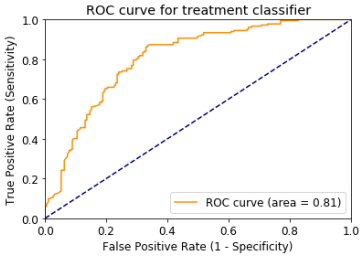
Random Forest


```
Classification Accuracy: 0.8418230563002681
Classification Error: 0.1581769436997319
False Positive Rate: 0.2879581151832461
Precision: 0.7639484978540773
AUC Score: 0.8450319314193661
Cross-validated AUC: 0.8874453342063073
```

For Naive Bayes

```
Classification Accuracy: 0.7613941018766756
Classification Error: 0.23860589812332444
False Positive Rate: 0.3298429319371728
Precision: 0.7123287671232876
AUC Score: 0.7636499626028422
Cross-validated AUC: 0.795378904249872
```



Precision

Precision is one way of measuring the accuracy where is calculate true positive out of all assuming positive (which is true positive + false positive)

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall

Recall is one way of measuring the accuracy where is calculated true positive out of all actual positive (which is true positive + false neagative)

$$\text{Recall} = \frac{tp}{tp + fn}$$

F1-score

F1-score is one way of measuring the accuracy where is calculated by this equation.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Our Decision tree score

	precision	recall	f1-score	support
0	0.97	0.71	0.82	191
1	0.76	0.98	0.86	182
micro avg	0.84	0.84	0.84	373
macro avg	0.87	0.85	0.84	373
weighted avg	0.87	0.84	0.84	373

Our random forest score

	precision	recall	f1-score	support
0	0.97	0.71	0.82	191
1	0.76	0.98	0.86	182
micro avg	0.84	0.84	0.84	373
macro avg	0.87	0.85	0.84	373
weighted avg	0.87	0.84	0.84	373

Our Naïve Bayes score

	precision	recall	f1-score	support
0	0.83	0.67	0.74	191
1	0.71	0.86	0.78	182
micro avg	0.76	0.76	0.76	373
macro avg	0.77	0.76	0.76	373
weighted avg	0.77	0.76	0.76	373

Tuning Hyper-Parameters

We use RandomziedSearchCV to tune that best fit hyper parameter which is basically iterate through all the input parameter and find the best parameter combination with highest accuracy.

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
from sklearn.tree import DecisionTreeClassifier


def finding_the_best_fit(model,X,y, param_dist):
    rand = RandomizedSearchCV(model, param_dist, cv=10, scoring='accuracy', n_it
    rand.fit(X, y)
    print('Rand. Best Score: ', rand.best_score_)
    print('Rand. Best Params: ', rand.best_params_)
    best_scores = []
    for _ in range(20):
        rand = RandomizedSearchCV(model, param_dist, cv=10, scoring='accuracy',
        rand.fit(X, y)
        best_scores.append(round(rand.best_score_, 3))
    print(best_scores)
featuresSize = feature_cols.__len__()
```

For Decision Tree Hyper parameter

```
param_dist_decision = {"max_depth": [3, None],
              "max_features": randint(1, featuresSize),
              "min_samples_split": randint(2, 9),
              "min_samples_leaf": randint(1, 9),
              "criterion": ["gini", "entropy"]}
model_decision = DecisionTreeClassifier()
finding_the_best_fit(model_decision,X,y, param_dist_decision)
```

```
Rand. Best Score:  0.8301127214170693
Rand. Best Params:  {'criterion': 'entropy', 'max_depth': 3, 'max_features': 6,
'min_samples_leaf': 7, 'min_samples_split': 8}
```

For Random Forest Hyper parameter

```
from sklearn.ensemble import RandomForestClassifier

param_dist_forest = {"max_depth": [3, None],
              "max_features": randint(1, featuresSize),
              "min_samples_split": randint(2, 9),
              "min_samples_leaf": randint(1, 9),
              "criterion": ["gini", "entropy"]}
model_forest = RandomForestClassifier(n_estimators = 20)
finding_the_best_fit(model_forest,X,y, param_dist_forest)
```

```
Rand. Best Score:  0.8301127214170693
Rand. Best Params:  {'criterion': 'entropy', 'max_depth': 3, 'max_features': 6,
'min_samples_leaf': 7, 'min_samples_split': 8}
```

Conclusion

The mental health of employees in the workplace is important because it can affect their moral and productivity especially in the IT industry where innovation and new ideas are desperately in need.

Recommendation

The workplace should provide employees with monthly mental health checkups.