The Resistance

CE811: Assignment 1

Keerthan Boraiah BORAI11904
Email:- kb22068@essex.ac.uk

The project aim is to create a model that learns, analyze and creates a probability to identify spy and resistance from game called The Resistance by Neural Network which learns from other player's behaviour. This projects explains why I choose Neural Network over Decision Tree by comparing the results of overall prediction.
The Neural Network showed significant accuracy compared to decision tree which persuaded me to choose this AI model to predict The Resistance game.

1. Introduction

The Main motivation of the project is to build a automated AI model which will be more than capable to play the game on its own and run The resistance.
The report gives detailed brief of three models, one is predefined to play which helps us to gather data of players behaviour and use it to other models to learn
The other two is neural network and decision tree which uses predefined data to learn and then play the game on its own.
Lastly, The best model is selected based on accuracy and used to play against the bot created for assignment explanation.

2. The Resistance

2.1 Game Introduction

The Resistance game is a friction Board-Game designed by **Don Eskridge** in 2010. The game is played between 5-10 players where the party is split into two seprate group (The resistance and The Spies) One the resistance is trying to take down the government reign by completing the mission and the spies who know the resistance's plan is trying to sabotage the mission and keep the government reign as it is.

2.2 Game Logic

At the Start of the game, The playes are randomly selected as spy and resistance and one player is made the Mission Leader. All the players are made to close the eyes and only spies are made to open the eyes so they would know who are the spies and remaining are the resistance group. only spies know each other but resistance dont know who is spy and that is the game who needs to identify who is spy and succes the mission to win for resistance and fail the mission for spies to win. Players must never reveal the identity card in the entire game.

2.3 Game Rules

The game is played for total number of 5 missions no matter the amount of players from 5-10. The objective of the game is for the resistance to complete 3 successfull mission to win for their team and bring down the government and the objective of spy in game is to sabotage the mission 3 times to win for their team and keep the government safe. The spy will win the game also win automatically if there is 5 missions rejection consecutively.

Each Mission there will be a leader assigned for the team to prepare for the mission and select players for the mission. The resistance must always vote for success of the mission while the spy can vote for failure or success based on the suitation they are in getting caught or not.
Each game a new leader is assigned and each game there will be vote counts of success or failure to determine wheater the mission is completed or sabatoged.
The players is split into two seperate parties as given in the below figure.

Number of Resistance Members & Government Spies

Number of players:	5	6	7	8	9	10
Resistance	3	4	4	5	6	6
Spies	2	2	3	3	3	4

Figure 2.2

3. Background:

3.1 Plain Agent

This agent doesn't use any machine learning or deep learning models to predict the probablity of being a spy or resistance and overall score. A Dictonary is used to store the count of all failed mission by players and consequently when that player is a leader he will assign and select players with less sabotage mission counts

3.2 Neural Network Agent

This agent is a machine learning model which uses weights, neurons, hidden layer(tanh, sigmoid) to predict the output based on the previous inputs. the previous data is collected by playing the bots many times and logging it then feeding the data into the model as all attribute as independent variable and output which is spy or not is

feeded as dependent variable.

Below Figure shows a simple brief explaining of how neural network works

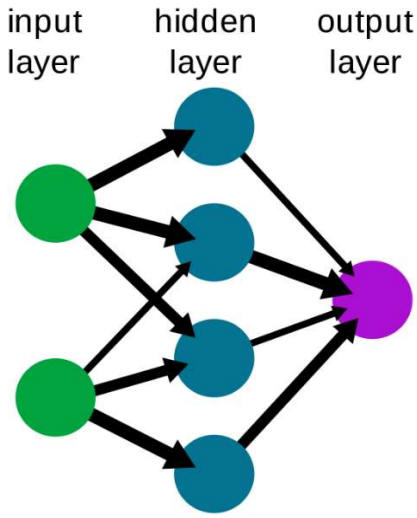


Figure 3.1

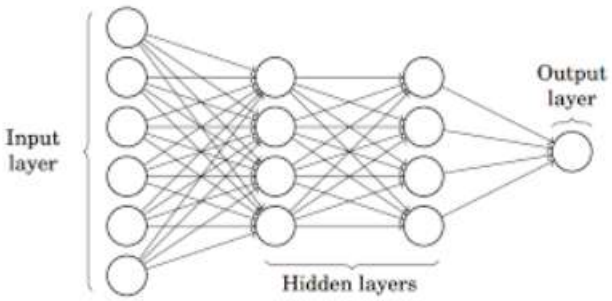


Figure 3.2

3.3 Decision Tree Agent

Decision tree is a supervised learning algorithm used for classification, which used technique of information gain and entropy (gini, chi-square) to get either yes,no or 1,0 output. the output is decide creating tree like representation by parents, nodes etc.

Below Figure shows basic working of Decision Tree

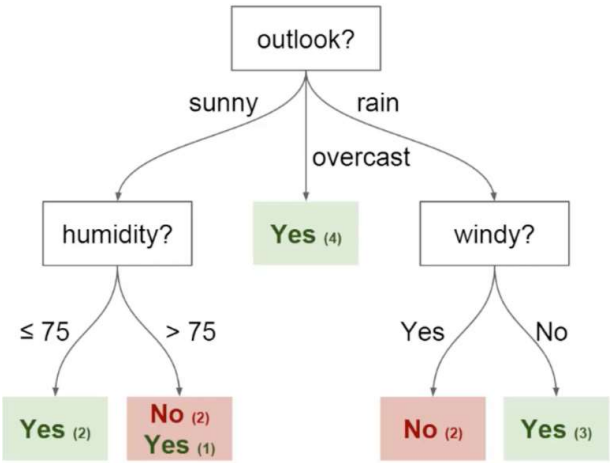


Figure 3.3

Below Formula shows how Decison Tree decides the prediction

Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)
Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)
Gini Index= 1- ∑jPj2

4. Techniques Implemented:

4.1 Tensor Flow/Keras Neural Network

Neural Network also known as artificial neural networks (ANNs) is a reflection of how human brain behaves, it allows computer program to recognize patterns and solve easy to complex problems. It is a open source machine learning and deep learning model in the AI field.
Below expression represents the NN in simple brief

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Figure 4.1

4.2 Sklearn Decision Tree

Decison Tree is also a open source machine learning model, it is a flowchart-like structure in which each of the internal node represents "test" and each branch represents the outcome of the test, and each of the leaf node represents a class label.
Below Figure explains brief of DT

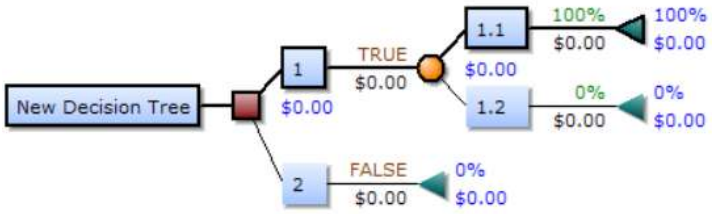


Figure 4.2

4.3 Model Building Technique

The data collected is splited into 70/30 ratio of Training/Testing using sklearn.model_selection package. The training model is used for building the model and the chunk of data which is testind data is used to check the prediction if trained model and get the accuracy score. This process is called as model evaluation either in accuracy, precision, recall.
The below figure represents the model building process

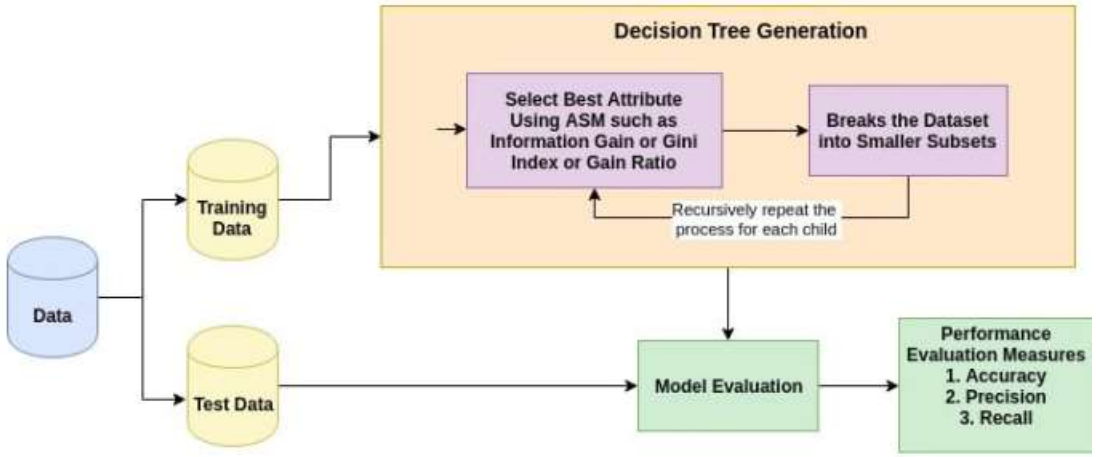


Figure 4.3

4.4 Data Collection

The data is collected by the running 10000 games of Beginners bot and using loggerbot to log the entire data in a ordered column and row format and saved as .log format. The data consists of 19 attributes as listed below.
Game.Turn:- (Attribute not used for building model)
Game.Tries:- (Attribute not used for building model)
Index:- (Attribute not used for building model)
Name:- (Attribute not used for building model)
Mission Been on:- Total mission been on in game
Failed Mission Been on:- Total Failed mission been on in game
Missions_voted_up_with_total_suspect_count:- count of voted for success of mission along with suspect count

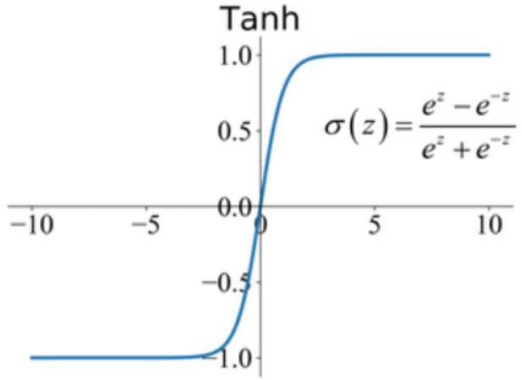
Missions_voted_down_with_total_suspect_count:- count of voted for failure of mission along with suspect count
Spy:- Showing wheather the player spy(1) or not(0)

5. Experimental Study:

5.1 Neural Network Experiments

Experiment 1:-

- I've used **2 hidden layers with 10 neurons** and activation function as **Tanh** ((e^x – e^-x) / (e^x + e^-x))
- why tanh as activation function is because tanh results in higher values of gradient during training and higher updates in the weights of the network which is in the range of -1 to +1 and wanted to experiment with this
- Optimizer as adam and **Learning rate of 0.001** with the **epoch of 120** iterations
- The **Accuracy** I've observed is **0.7258**

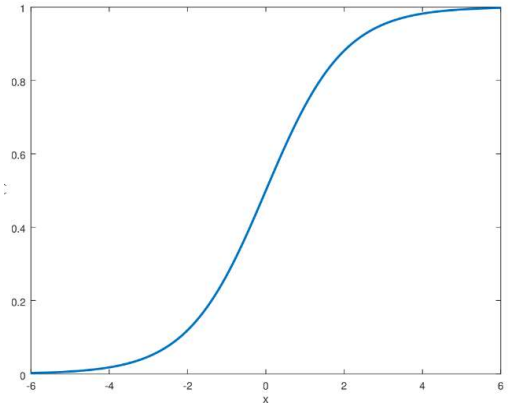


Tanh Figure 5.1

Experiment 2:-

- I've used **1 hidden layers with 10 neurons** and activation functions as **sigmoid** (1/1+e power -x)

- why sigmoid as activation function is beacuse sigmoid takes any real value as input and outputs values in the range of 0 to 1 so wanted to try this function.
- Optimizer as adam and **Learning rate of 0.001** with the **epoch of 150** iterations
- The **Accuracy** I've observed is **0.7230**



Sigmoid Figure 5.2

5.2 Decision Tree

As Neural network Accuracy, Performance and Prediction were higher than decision tree which had the following performances as stated below

- **Accuracy:- 0.7060**
- **Confusion Matrix:-**
  Precision:- **0.70**
  Recall:- **0.71**
  F1-Score (Mean of both Precision and Recall) :- **0.70**

The Experiment against Neural Network and Decision Tree Concluded with NN as a winner for the current problem solution.
Refer Below Classification Report for detailed performance

```
              precision    recall  f1-score   support

           0       0.75      0.75      0.75      2369
           1       0.64      0.65      0.65      1669

    accuracy                           0.71      4038
   macro avg       0.70      0.70      0.70      4038
weighted avg       0.71      0.71      0.71      4038

Confusion Matrix is:-  [[1768  601]
 [ 586 1083]]
Accuracy is:-  0.7060425953442299
```

Figure 5.3

6. Analysis:

6.1 Plain agent Performance

I've noticed that the simple agent which doesnt use any machine learning model to predict the spy or resistance probability is actually predicting better even for 1000 games and 10000 games played

```
TOTAL
  RuleFollower     49.9% (e=3.69 n=701)
  Hippie           49.2% (e=3.66 n=713)
  Paranoid         46.6% (e=3.66 n=710)
  Neighbor         44.6% (e=3.64 n=713)
  Deceiver         42.3% (e=3.60 n=721)
  Jammer           38.4% (e=3.57 n=709)
  RandomBot        28.1% (e=3.24 n=733)
```

Figure 6.1 1000 Games

```
TOTAL
  RuleFollower     48.1% (e=1.16 n=7102)
  Paranoid         45.1% (e=1.16 n=7123)
  Hippie           44.8% (e=1.15 n=7202)
  Neighbor         42.9% (e=1.15 n=7081)
  Deceiver         42.4% (e=1.14 n=7178)
  Jammer           41.4% (e=1.14 n=7190)
  RandomBot        31.4% (e=1.08 n=7124)
```

Figure 6.2 10000 Games

6.2 Multiple Algorithm Analysis

I've tried to build multiple learning agent from the data collected by playing 10000 games, and implemented Neural Network, Decision Tree and Support Vector Machine models. To my suprise, SVM had a not so good accuracy than Decision Tree but usually A support vector machine is used for each decision in the tree Then "optimal" decision tree is characterized. Finally, had to go with Neural Network and Decision Tree as my final comparision models.

6.3 Learning agent Performance

By training Neural Network and letting it predict the probability of spy and resistance I've analysed the prediction was actually better that the predifined performance.

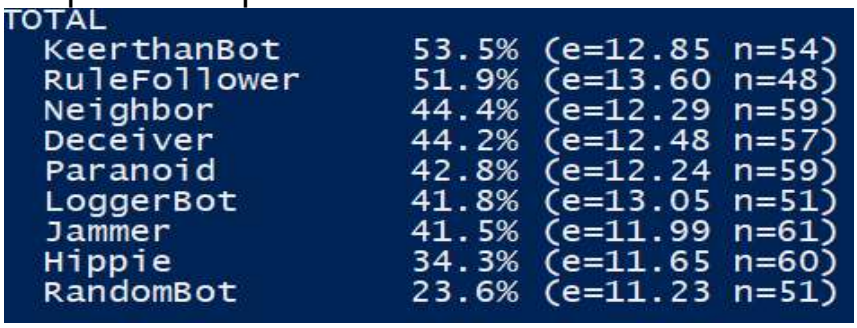


Figure 6.3 Neural Network Prediction

To Improve the model I had to change some logic in sabatoge and vote function to only return the fuction if the condition is satisfied. I've later analysed and compared that by tweaking the logic i was able to improve the probability prediction by a getting better performance.
The difference varies from original to changed prediction because the spy probability was increased accurate while resistance probabilty was varied a little causing overall changes to varies from high to low from original neural network

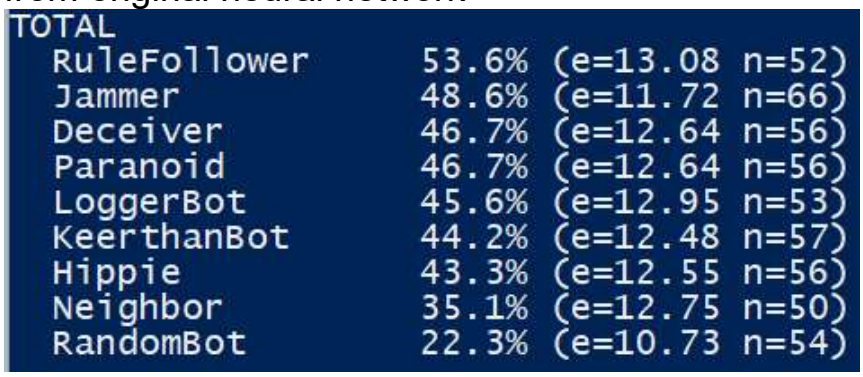


Figure 6.4 Neural Network Prediction

7. Overall conclusions:

The overall project showed me that whatever models we build by implementing machine learning models, that can be imporved either

by changing the logic of voting, sabotage, etc. Moreover, from the analysis and experimental study I've got to understand that the more data tuning and better data will lead to better accuracy of model predicition. Firstly, the game focuses on playing as a resistance and identify the spy as the main goal. so by improving the spy cunningness and technique of spy playing, the model and the game can be improved much further. Finally, the voting is a basic logic and could be implemented a complex structure to make it more intriguing.

8. References

1. https://en.wikipedia.org/wiki/The_Resistance_(game)- Reference for the rule players dividing table and chart
2. https://www.analyticsvidhya.com/blog/2021/05/beginners-guide-to-artificial-neural-network/- Reference for neural network working and chart
3. https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-decision-tree-classification-using-python/- Reference for Decision Tree working and chart
4. https://www.ibm.com/cloud/learn/neural-networks- Neural Network explanation
5. https://en.wikipedia.org/wiki/Decision_tree- Decision Tree explanation and model building technique chart