**Predicting the results of NBA games by analyzing Tweets**

**By Yusu Qian**

**Abstract:**

Our team collects real-time tweets on ‘NBA’ using Twitter API, performs sentiment analysis on the tweets, and trains various machine learning models to predict the outcome of different matches using counts of tweets on the two teams involved and the percentage of positive, negative and neutral tweets respectively.

**Research Agenda and Work Allocation:**

In the week of November 13th, we studied and compared different methods of collecting tweets and analyzing sentiment contained in the tweets. I followed tutorials to wrote two separate codes for our project; one for real-time tweet collecting and one for sentiment analysis. Leona made modifications to put the two together.

In the week of November 20th, we finished the first draft and ran a test to see how our program is functioning. We decided on the amount of tweets to gather each day to be 10000, which normally takes one to two hours to complete.

In the week of November 27th, we collected tweets and performed sentiment analysis on them through Monday to Saturday. On Sunday Leona (Xiaoning He) and Po Yang created a csv file to put in all the numeric data, such as counts of tweets and percentages of positive tweets of Warriors. I built several machine learning models to train on the data.

In the week of December 3rd, we collected tweets for a few more days to see whether the size of data effects the accuracy of prediction significantly. I finalized the project by comparing the performance of models I built.

**Data Gathered and Curated:**

First we used Twitter API to collect 10000 tweets containing the key word ‘NBA’ approximately two hours before the first match on that day takes place. The data collecting process should end before the first match takes place, otherwise the ongoing match itself will influence people’s sentiment in tweets they send. While it was not necessary, we wrote a function to download the text file containing tweets we collected.

Next we preprocessed the data. We’d like to know what percentage of the tweets were in English. On different days, the percentage ranged from 75% to 80%. This was acceptable so we decided to neglect tweets in other languages.

We then wanted to separate the tweets into groups by setting a second filter. For example, on December 1st, there were 14 teams taking part in the games. The names of these teams ('Bucks','Knicks','Nets','Wizards','Warriors','Pistons','Celtics', 'Timberwolves', 'Bulls','Rockets','Raptors','Cavaliers','Pacers','Kings') formed the second filter. I drew a bar chart to visualize the counts of tweets about these teams respectively (figure 1 in appendix).

To add more fun, we drew word clouds to see what people talk about when they talk about different teams (figure 2 in appendix)

Then we analyzed the sentiment of tweets for each team using Textblob. Here is a description of Textblob from its official website*: TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more.*

We wrote a function that puts sentiment analysis and tweet counts of two teams in the same match together in order to compare the statistics and predict which team is more likely to win. Here is an example of what we got for a match between Bulks and Knicks: *Bucks: total tweets: 237*

*Positive Tweets: 121, percentage: 51.054852%*

*Negative Tweets: 30, percentage: 12.658228%*

*Neutral Tweets: 86, percentage: 36.286920%*

*Knicks: total tweets: 349*

*Positive Tweets: 162, percentage: 46.418338%*

*Negative Tweets: 43, percentage: 12.320917%*

*Neutral Tweets: 144, percentage: 41.260745%*

The percentage of positive tweets about Bucks is slightly higher than that of Knicks. The percentage of negative tweets about Bucks is also slightly lower than that of Knicks. However, the total number of tweets of Bucks is one third smaller than that of Knicks. In this circumstance, it is hard to determine which team is more likely to win. It turned out that Knicks won, suggesting that a larger amount of tweets might mean that a team is more popular than its opponent, a factor we should take into consideration when making predictions.

Let’s look at the match between Pacers and Kings on the same day as another example:

*Pacers: total tweets: 22*

*Positive Tweets: 8, percentage: 36.363636%*

*Negative Tweets: 4, percentage: 18.181818%*

*Neutral Tweets: 10, percentage: 45.454545%*

*Kings: total tweets: 42*

*Positive Tweets: 20, percentage: 47.619048%*

*Negative Tweets: 5, percentage: 11.904762%*

*Neutral Tweets: 17, percentage: 40.476190%*

In this case, Kings not only has a higher percentage of positive tweets and a lower percentage of negative tweets, but also a larger amount of tweets. It turned out than Kings did win, just like we predicted.

**Applied Data Science Methodologies:**

From the last part we figured out that different weights should be given to percentage of positive and negative tweets as well as the count of tweets. After careful consideration, we decided to use these as our input variables: percentage of counts of team 1 (meaning counts of team 1 divided by the total counts of both teams in a match), percentage of counts of team 2, percentages of positive and negative tweets of both teams. These six categories make the input variables while the results of matches, represented using 1 for won and 0 for lost, make the labels (figure 3).

Before building machine learning models, we’d like to have a better idea of the distribution of the variables. We drew box plot, histograms and scatter matrixes for each input variable (figure 4).

The last part of this project is building machine learning models used for predicting the results of matches. I chose Logistic Regression, Decision Tree, SVM and KNN. The accuracy of these on the training set and the test set is shown in figure 5.

The performance is far from satisfying. To further improve the accuracy, I built a Neural Network model. I used 3 layers and set epoch to 100. The first layer had 6 units, the second layer had 3 units and the last layer had 1 unit. The accuracy is now 69%, slightly higher than other models (figure 7).

**Improvement:**

In this project, when setting the second filter we used the official names of teams. However, when people tweets about NBA teams they use nicknames more often than not. To achieve a better result, nicknames should be put into the second filter as well.

Also, the result of NBA matches and tweets on the matches are correlated, but there is no causal relationship. We believe that if the project is run on presidential election it might provide much more useful information.

**Conclusion:**

In recent years Twitter has gained popularity and many tweets contain real-time opinions on ongoing events. Throughout this project, we collected, cleaned and visualized data, and developed various models implementing Logistic Regression, SVM, Decision Tree and Neural Network to predict results of NBA games using information contained in Tweets.

**References:**

1. Pushkar Mandot  <https://medium.com/@pushkarmandot/build-your-first-deep-learning-neural-network-model-using-keras-in-python-a90b5864116d>

1. Susan Li  <https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2>

**Appendix:**

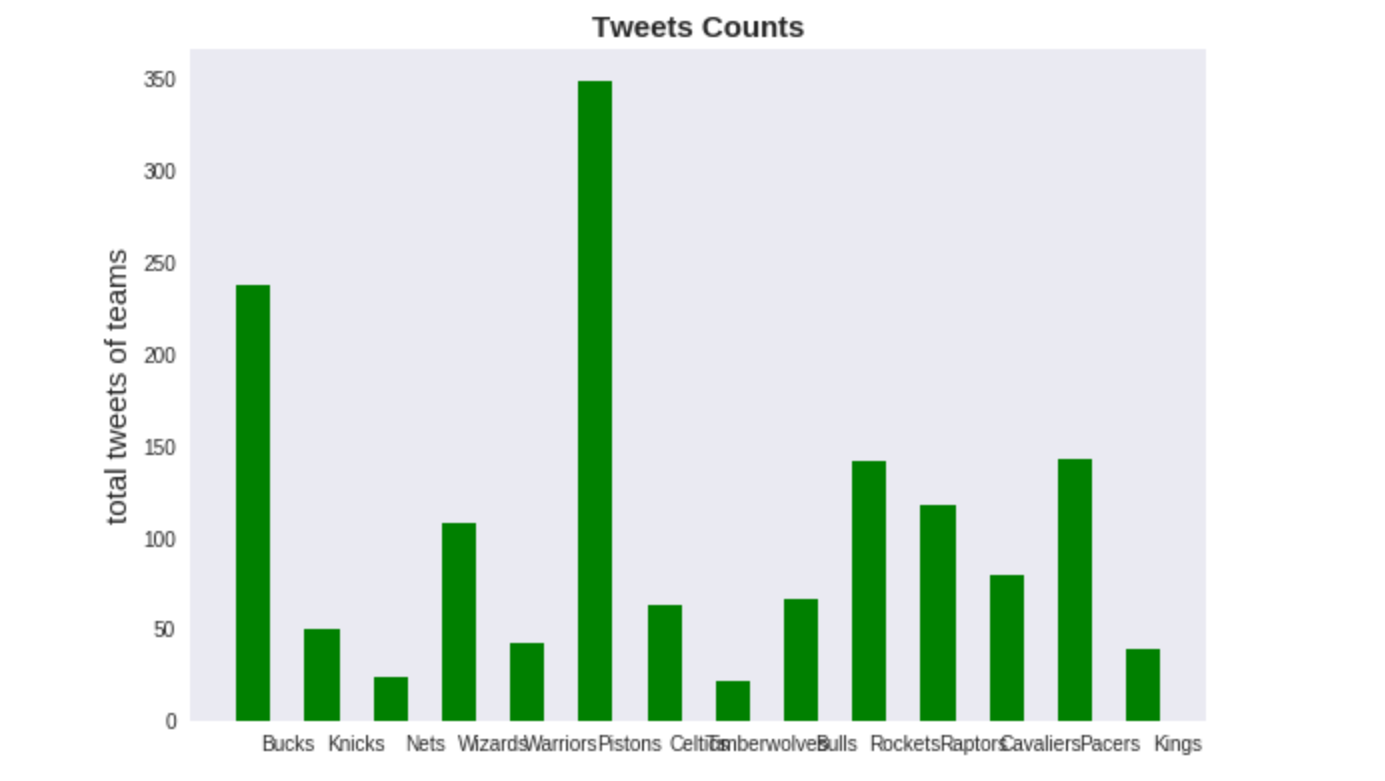


Figure 1: counts of tweets



Figure 2: word cloud

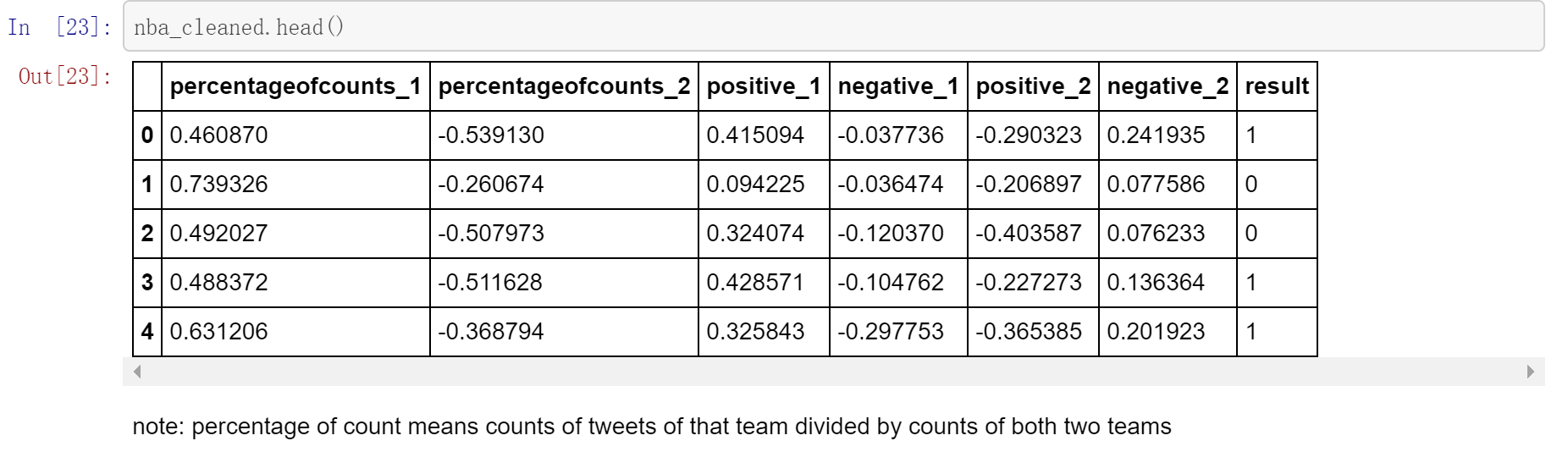


Figure 3: cleaned data

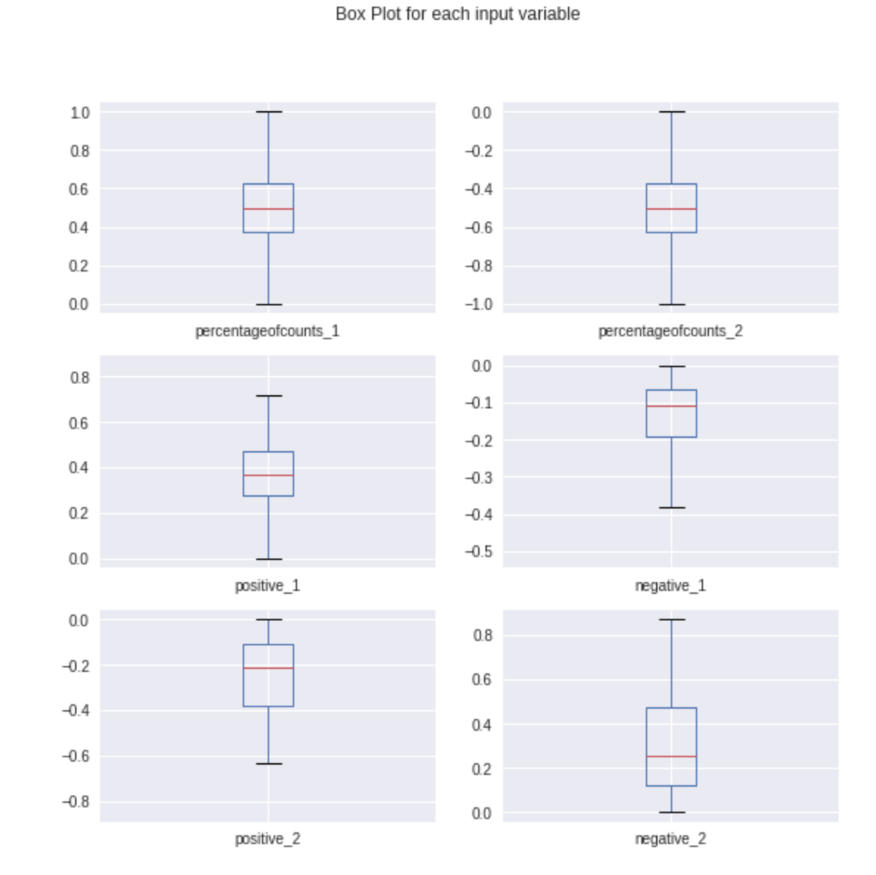
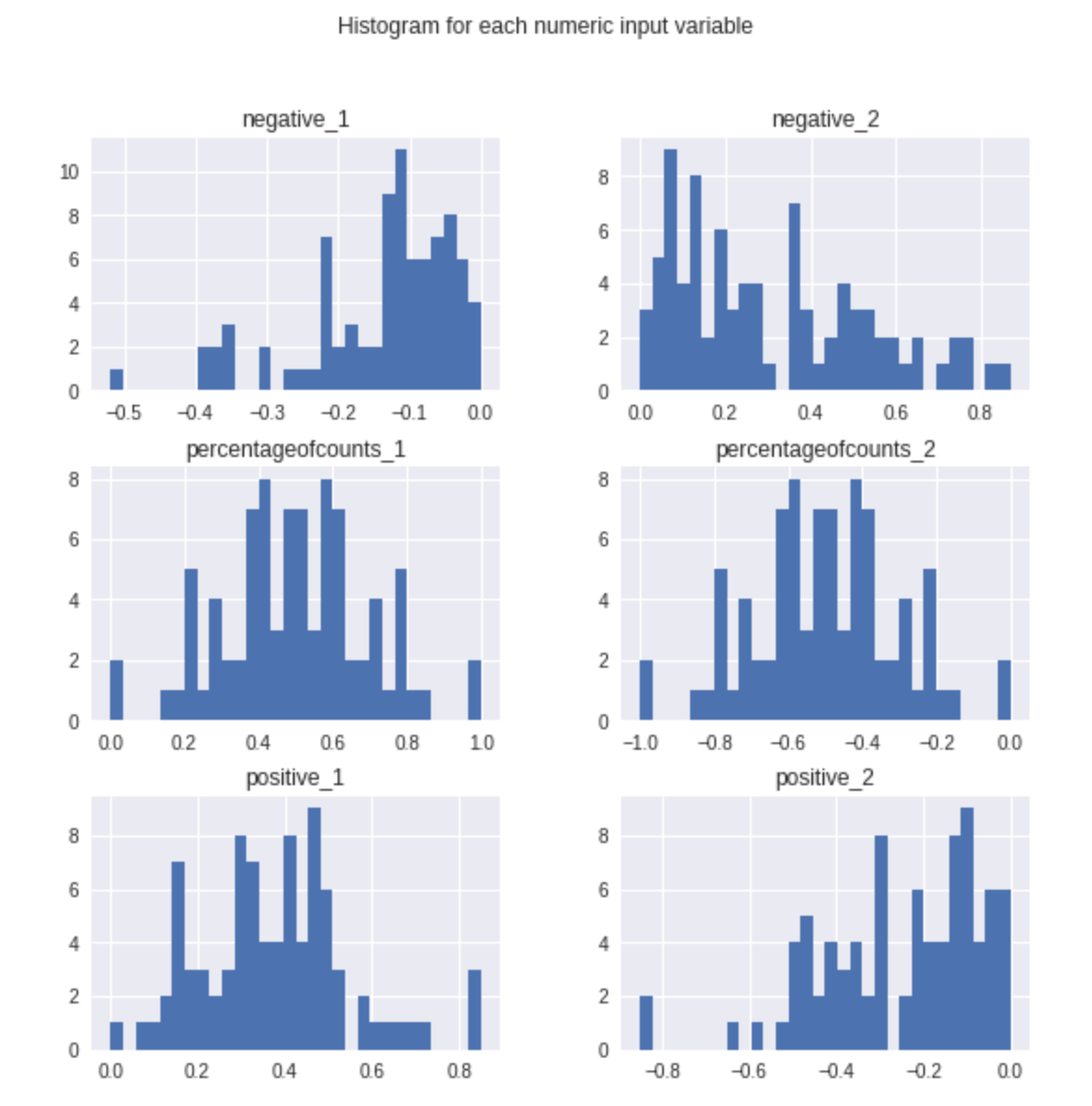


Figure 4: histograms and box plots of each variable

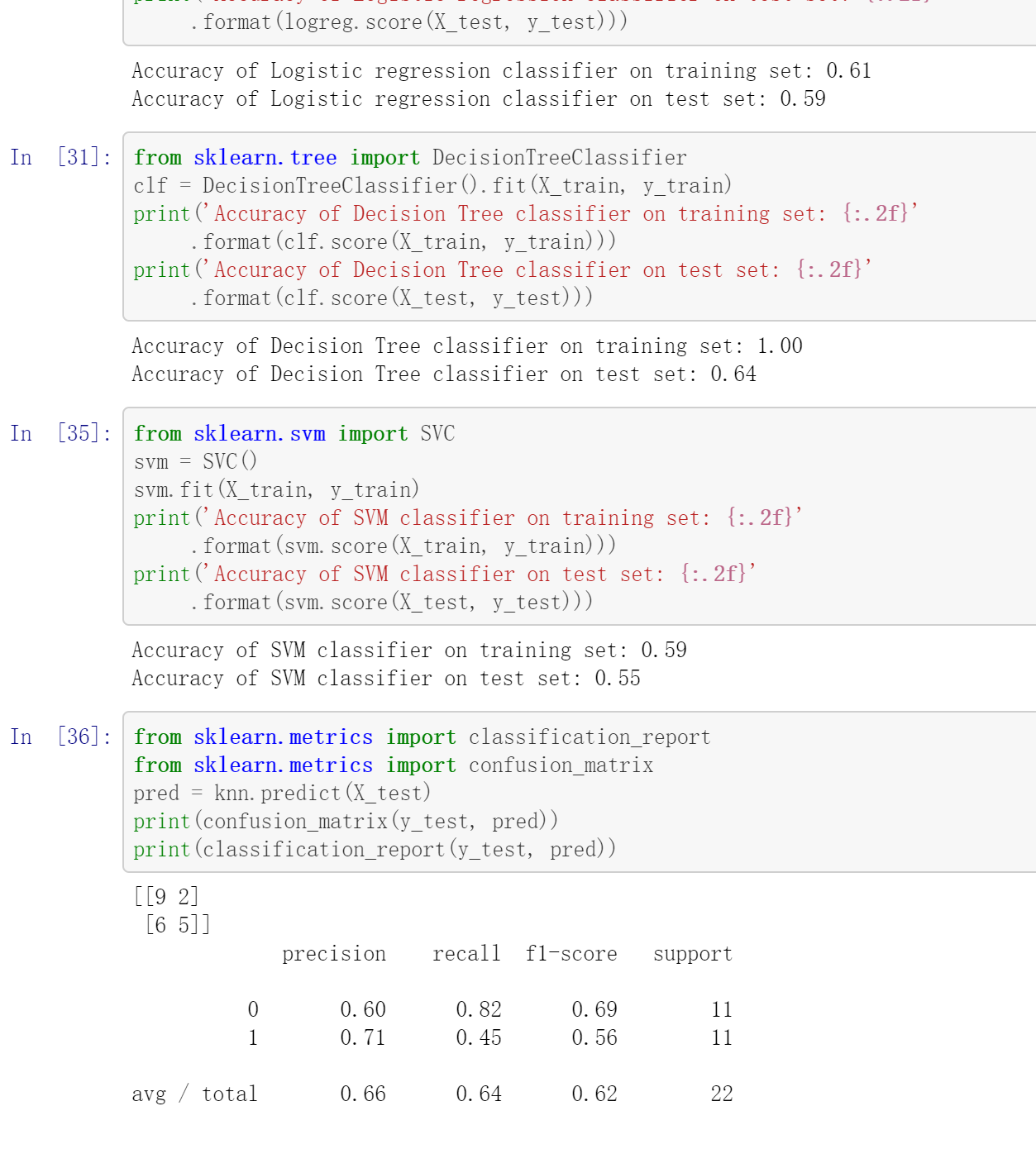


Figure 6: accuracy of logistic regression, decision tree and SVM models



Figure 7: accuracy of neural network model