CASE STUDY 3

Eric Schuman, Jackson Lombardi, Amos Roche, Mark Buono

DS 3010

1 Introduction:

This case study provided a focus around working with "Text Data" and more specifically the sentiment analysis that can be conducted from text data. For this study, our group looked at movie reviews from a polarity dataset from cornell. In this dataset existed positive and negative movie reviews. The Python library that was most essential for this assignment was scikit-learn. The data was uploaded to Google Colab and sentiment analysis was coded using the solution provided from scikit-learn. The next step was to gain understanding of the term frequency-inverse document frequency (TF-IDF) that was used in the sentiment analysis. The next problem focused on gaining understanding of two different machine learning algorithms (Linear Classifier and K Nearest Neighbors Classifier). These algorithms were run and their accuracies were compared further. The Multi-Layer Perceptron (MLP) algorithm was also run to classify the reviews, and its accuracies and runtimes were then compared using different inputs. Lastly as a business problem, our group has looked into the Stem Player that Kanye is releasing his newest album on. Using our algorithm we wanted to gauge public sentiment on the idea of purchasing a Stem Player device to listen to the album exclusively. After this we then went on to modify a solution to Kanye's idea and tap into this untouched market.

2 Sentiment Analysis on Movie Reviews:

After getting the movie review data into Colab, we split the data into a training set and testing set (70% training and 30% testing). We then created a pipeline with a vectorizer and classification model that we would use to run on the data. From here, we performed a grid search using training data to see how the model worked with just unigrams, or unigrams and bigrams. We were then able to fit the model with the best parameters. Using panda's dataframe, we were able to store the results of our grid search using the best parameters (see Figure 1). Using this model we obtained an accuracy score of 86% (see Figure 2), and terms that appeared in less than 3 of the documents and more than 90% of the documents were ignored. We then wanted to see how using the default parameters on the vectorizer would affect our model. We then gave the

gridsearch a range of parameters for the C and kernel values and it resulted in a linear kernel with a C = 100 as the optimal parameter configuration. After training the model, we obtained a testing accuracy score of 83%. So using the default parameters resulted in a three point decrease in our model accuracy.

mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_C	param_kernel	params	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score	mean_test_score	std_test_score	rank_test_score
3.808352	0.031918	0.904448	0.007968	100	rbf	{'C': 100, 'kernel': 'rbf'}	0.842857	0.850000	0.828571	0.825000	0.810714	0.831429	0.013814	3
3.788221	0.040701	0.842395	0.014444	100	linear	{'C': 100, 'kernel': 'linear'}	0.864286	0.835714	0.839286	0.839286	0.832143	0.842143	0.011384	1
3.849466	0.047682	0.906489	0.006651	1000	rbf	{'C': 1000, 'kernel': 'rbf'}	0.842857	0.850000	0.828571	0.825000	0.810714	0.831429	0.013814	3
3.782237	0.017419	0.848062	0.014482	1000	linear	{'C': 1000, 'kernel': 'linear'}	0.864286	0.835714	0.839286	0.839286	0.832143	0.842143	0.011384	1

Figure 1: Results of the Linear SVC grid search. The linear kernel model performed best, while the C value did not affect model accuracy.

	precision	recall	f1-score	support
neg	0.88	0.85	0.87	319
pos	0.84	0.87	0.85	281
accuracy			0.86	600
macro avg	0.86	0.86	0.86	600
weighted avg	0.86	0.86	0.86	600

```
[[272  47]
 [ 37 244]]
```

Figure 2: Classification Report and Confusion Matrix for the best Linear SVC model from the grid search.

3 Scikit-Learn TF-IDF Vectorizer Class:

The term frequency-inverse document frequency (TF-IDF) returns a statistic to convey how important a word is to a document in a collection. The statistic increases as word frequency in the document increases and is offset by the document frequency containing the word. It must be offset due to the fact that some words appear frequently in general. The next thing done was to compare the different inputs. The min_df input refers to the minimum document frequency that the word must appear in (it can be a percentage or specified number). The max_df input refers to the maximum document frequency that the word can appear in (it can be a percentage or specified number). The ngram_range refers to the types of words selected, examples being unigrams or bigrams. When putting tighter constraints on min_df and max_df compared to the default it was seen

that the feature count would decrease. When changing ngram_ranges, the one that accepted both unigrams and bigrams returned the most words and the one that only returned bigrams returned the least. This made sense considering that there are more combinations of unigrams and bigrams than just bigrams.

4 Machine Learning Algorithms:

In this section, we wanted to compare the results of the support vector classifier using a linear classifier and the KNN classifier. We wanted to explore a vectorizer ignoring terms that appear in less than 10% of the documents and more than 90% of the documents. We then passed the KNN and Linear SVC into the randomized search cross validation, and the Linear SVM accuracy score for a $C = 10$ was 77%. The cross validation showed that $k = 11$ was the ideal k value, but since randomized search is not as reliable as grid search since it isn't exhaustive, we plotted accuracy scores for different k values manually. If we look at Figure 3, we can see that k=8 gives the best accuracy. Looking at the results, it looks like the data fits more accurately to the linear classifier. When a linear classifier model performs well, this means the data possesses a linear relationship and stronger models may not be better because of the simplicity of the data. In Figure 5 , we can see there are 137/600 observations that are misclassified for the linear classifier. This may be because the linear classifier is not optimized all the way, causing more hyper parameter tuning. Also, there is always noise in the data, so the data can never fit 100% accurately.

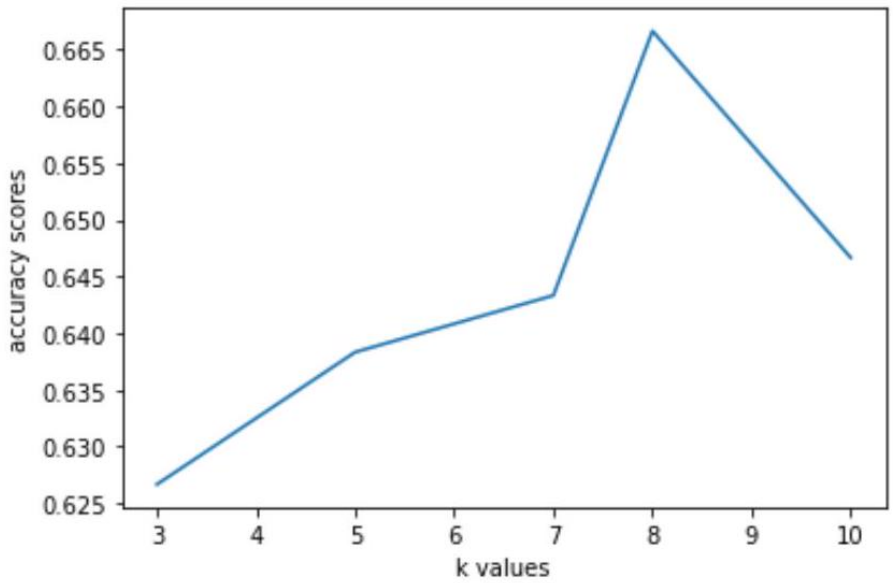
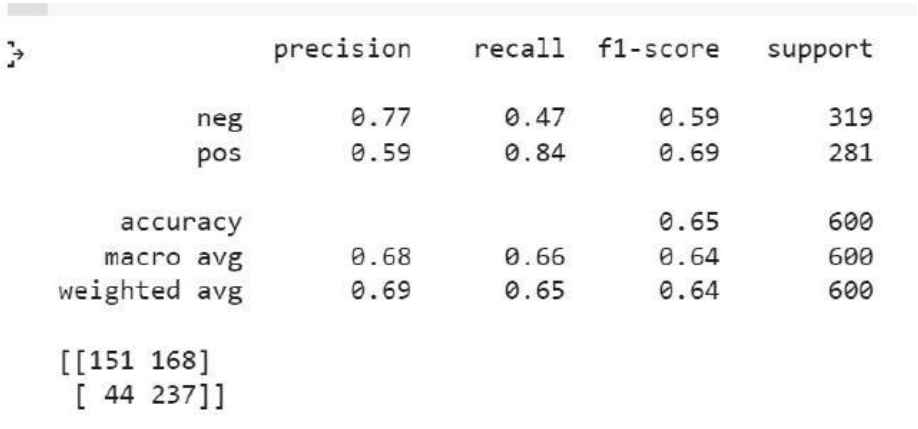


Figure 3 - Relationship between k value and accuracy score



	precision	recall	f1-score	support
neg	0.77	0.47	0.59	319
pos	0.59	0.84	0.69	281
accuracy			0.65	600
macro avg	0.68	0.66	0.64	600
weighted avg	0.69	0.65	0.64	600

```
[[151 168]
 [ 44 237]]
```

Figure 4- Classification Report and Confusion Matrix for the KNN model with $k = 11$, min_df = .10, max_df = .90.

	precision	recall	f1-score	support
neg	0.79	0.78	0.78	319
pos	0.75	0.77	0.76	281
accuracy			0.77	600
macro avg	0.77	0.77	0.77	600
weighted avg	0.77	0.77	0.77	600

```
[[248  71]
 [ 66 215]]
```

Figure 5- Classification Report and Confusion Matrix for the Linear SVC model with a $C = 10$, min_df = .10, max_df = .90.

5 Multi-Layer Perceptron Model:

We wanted to see how the linear and KNN classifiers compare to the MLP model. More specifically, we wanted to see how different activation functions and hidden layer sizes correspond to model accuracies. To find the optimal parameters for the tf-idf vectorizer and the MLP model, a randomized search cross validation was conducted using the training data. From the results of the cross validation, the optimal parameters were using a logistic regression activation function, 30 hidden layers, 0.01 for min_df, 0.99 max_df, and $(1, 2)$ for the ngram_range.

The min_df and max_df parameters represent that we ignore terms that are in less than 1% of the documents and terms that occur in more than 99% of the documents. If we look at Figure 6, the accuracy score for this configuration of the MLP model was 84.667%. The accuracy for the MLP with a logistic activation function with 60 hidden layers and 80 hidden layers was 84.5% and 84.3% respectively. In Figure 7 , it shows that the relu activation function MLP was a little worse than the logistic MLP with 30 hidden layers. The accuracy for the relu MLP was 84.5% for hidden layer sizes of $30, 60, 80$. The results show that for the vectorized movie reviews, the different hidden layer sizes and the activation functions did not affect the accuracy significantly. In Figure 8, we can see the results of the best models in terms of accuracy for the vectorized movie reviews. We can see that the logistic MLP with 30 hidden layers was the best model, and the linear classifier with a C of 100 was the second best model.

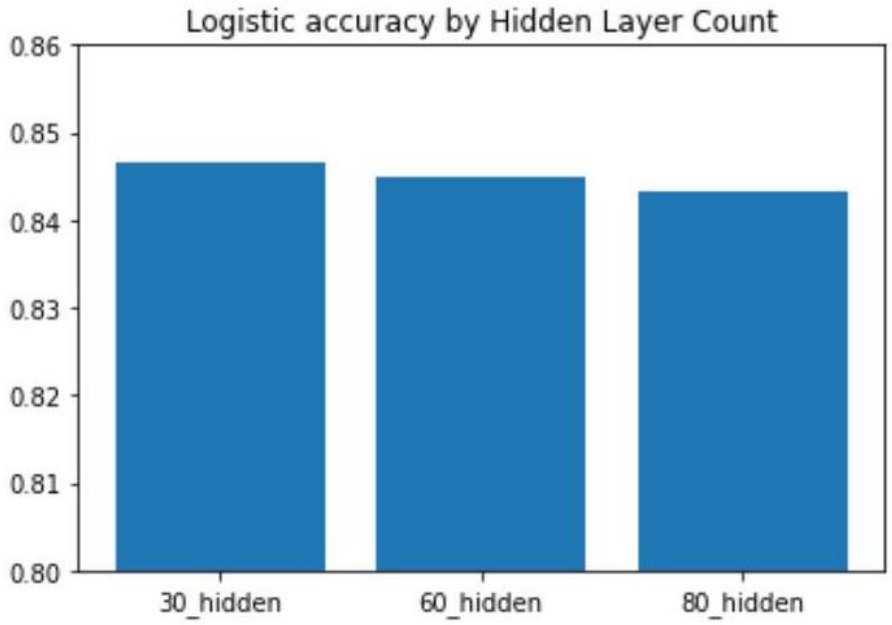


Figure 6- Relationship between number of hidden layers and accuracy using the logistic activation type.

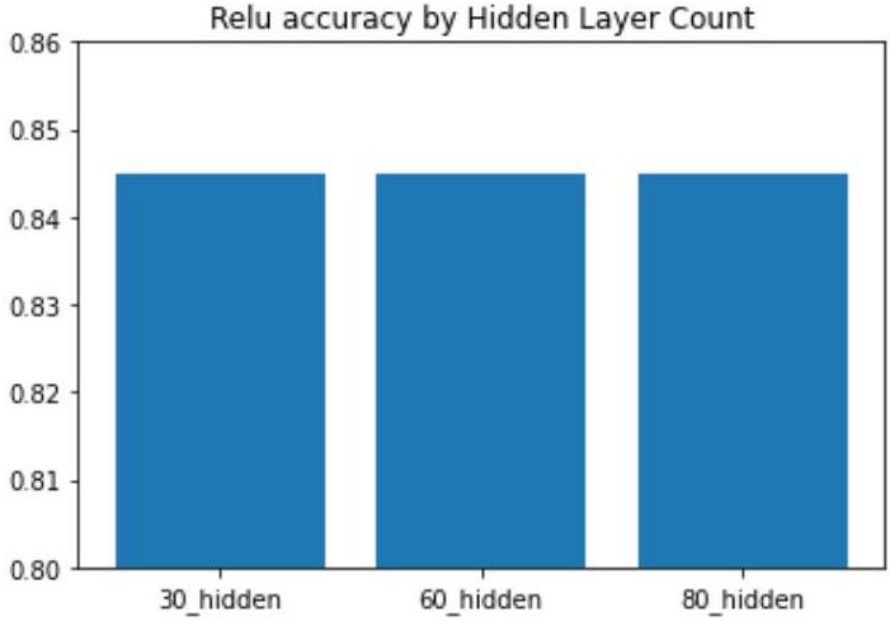


Figure 7- Relationship between number of hidden layers and accuracy using the relu activation type.

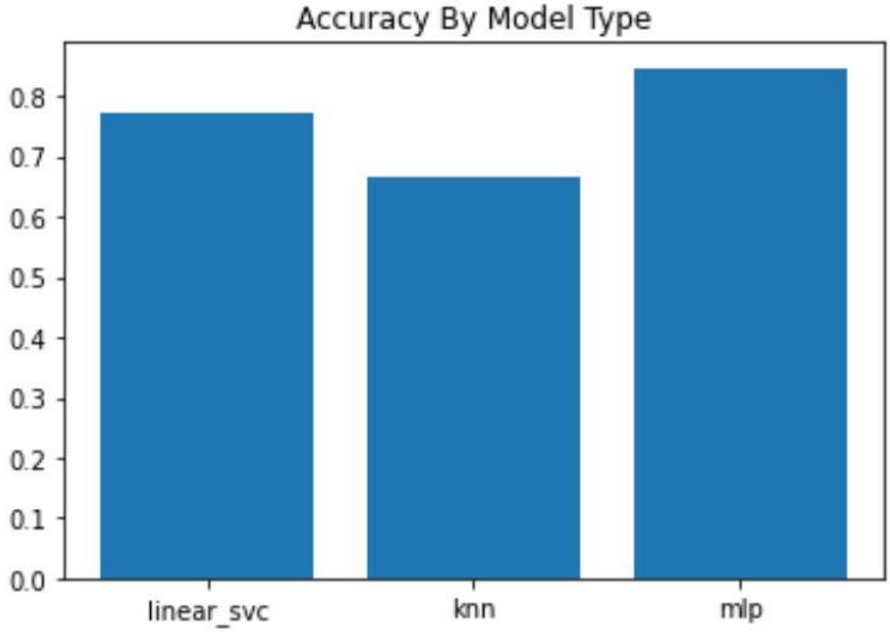


Figure 8- Comparing the accuracies of the three models (Linear SVC, KNN, MLP)

6 MLPClassifier Runtime and Accuracy Comparison:

After using the Multilayer Perceptron model to classify the reviews, we wanted to see how changing the number of hidden layers and activation types would affect the runtime and accuracy of the algorithm. We first ran the classifier using 300 iterations and the default values for hidden layer size and activation type (100 hidden layers and relu activation type). The algorithm took 31.57 seconds to run and returned an accuracy score of about .78.

To see the effect of adding hidden layers to the algorithm, we ran a for loop where the MLP Classifier changed the number of hidden layers from 50 to 200 using 50 layer increments. For each iteration of the for loop, we recorded the runtime of the algorithm and the accuracy score returned. The results can be seen in Figure 9 and Figure 10. There was a direct relationship observed between the number of layers and the runtime, which was expected as it takes more computations and data transformations as the number of hidden layers is increased. At first, the accuracy score increased as the number of layers increased. A peak accuracy score was reached at .778 (in which there were 100 hidden layers). The accuracy score then decreased and leveled out at .770 as more layers were added. We concluded that the default value of about 100 hidden layers was optimal for this specific problem.

In an attempt to validate our conclusions further, we re-ran the algorithm with 9 for loop iterations, ranging from 1 to 200 hidden layers (25 layer increments). We were surprised to find that our prior conclusions were not entirely accurate. As shown in Figure 11, the runtime still increased as the number of hidden layers increased, but the increase was more dramatic as the number of layers increased from 1 to 25 (20 second increase), and then gradual as the number of layers increased from 25 to 200 (only increased about 10 seconds). As Figure 12 shows, the accuracy score fluctuated. Having only 1 hidden layer outperformed all other models, and the accuracy score trend saw peaks at 50 and 100 hidden layers, but local minimums at 25,75 , and 125 hidden layers. Overall the accuracy did not change much in value as it still ranged from .76 to .78, so we concluded that the number of hidden layers did not have a significant impact on the model's accuracy.

To see the effect the activation type of the MLP Classifier had on runtime and accuracy score, we ran a for loop again, this time changing the activation type on each iteration. We found that the relu activation type had the fastest runtime of about 25 seconds, while the logistic function had the slowest runtime of close to 50 seconds (see Figure 13). The accuracy score did not change much between the four different types, with the relu and logistic types slightly edging the tanh and identity types. All four types returned an accuracy score close to

.78 (results in Figure 14).

We also ran a quick test to compare the runtime needed to fit and predict the model. We predicted that the fit time would take longer than making predictions as it requires more computations and data transformations. Our results validated our hypothesis as the fit time was 21.31 seconds compared to the prediction time of only .015 seconds.

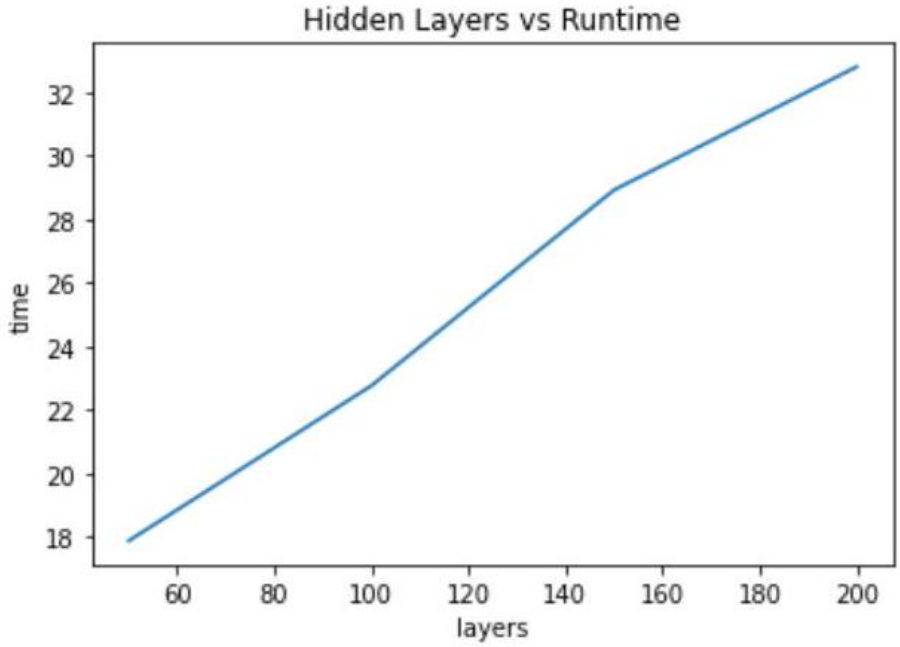


Figure 9: Effect of hidden layers on runtime $(50, 100, 150, 200$ layers tested)

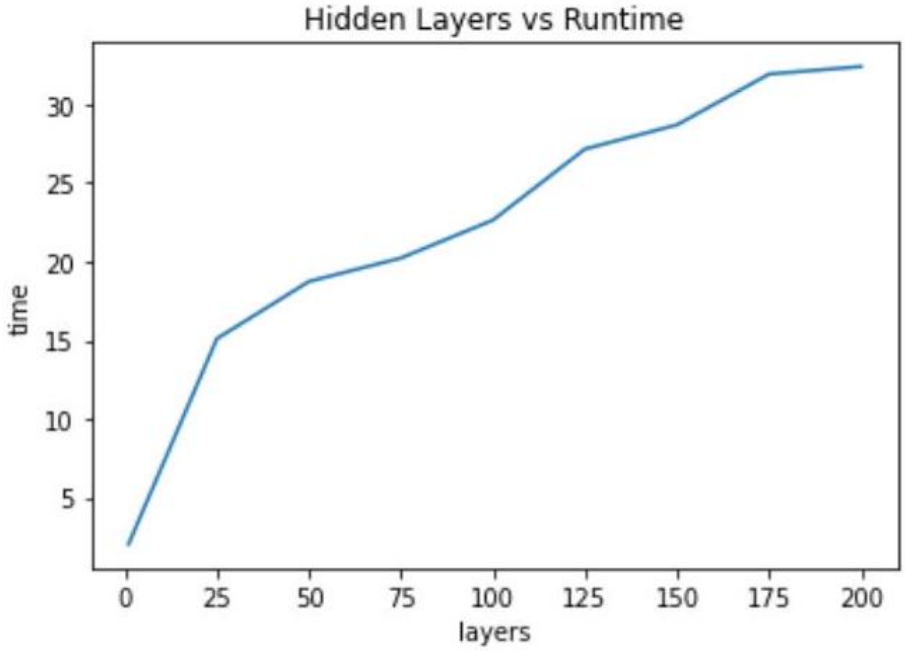


Figure 9: Effect of hidden layers on runtime ($1, 25, 50, 75, 100, 125, 150, 175, 200$ layers tested)

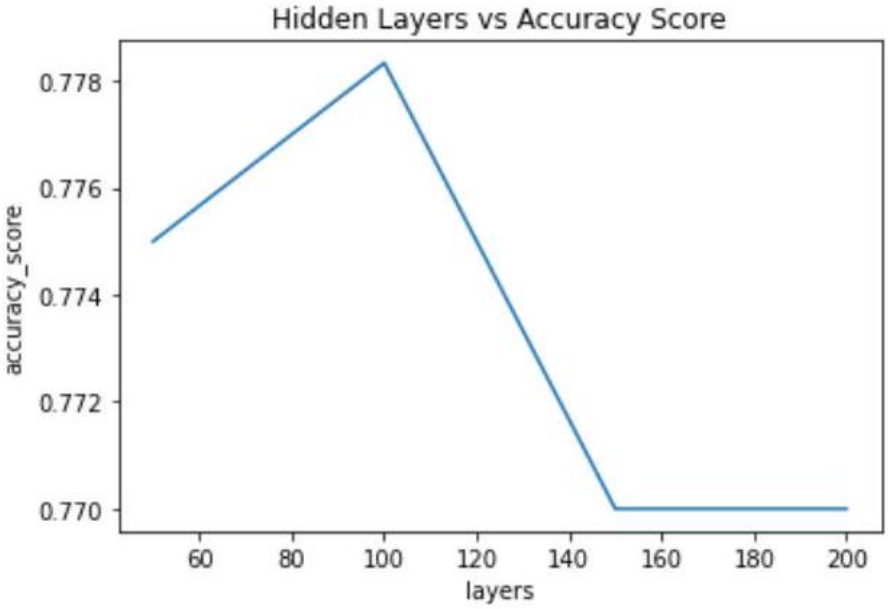


Figure 10: Effect of hidden layers on accuracy (50, 100, 150, 200 layers tested)

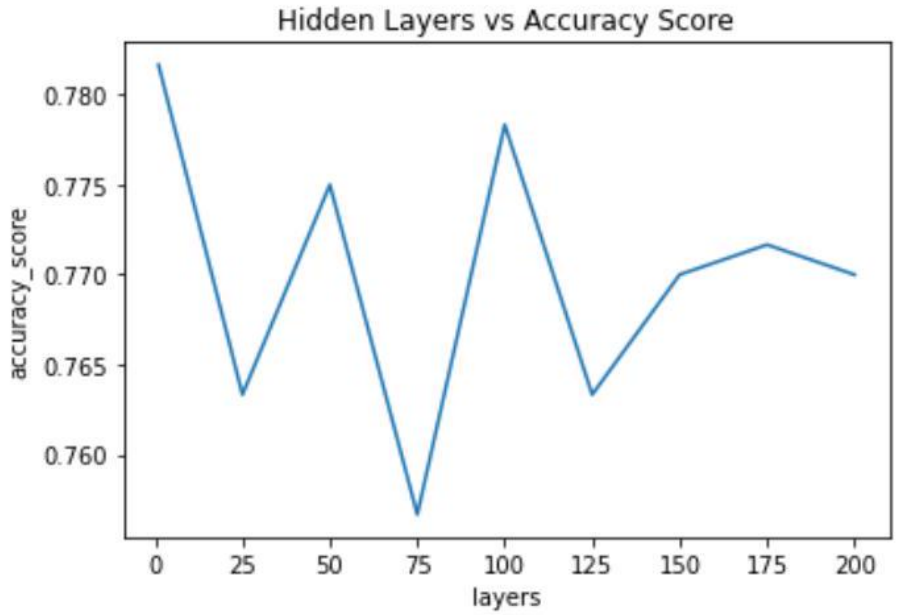


Figure 10: Effect of hidden layers on accuracy $(1, 25, 50, 75, 100, 125, 150, 175, 200$ layers tested)

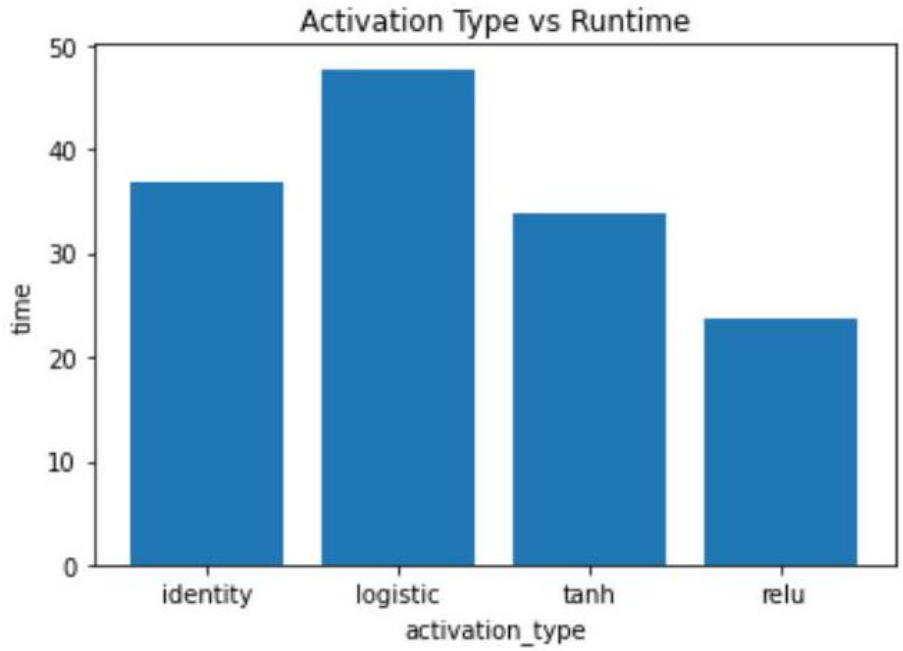


Figure 13: Effect of Activation Type on Runtime

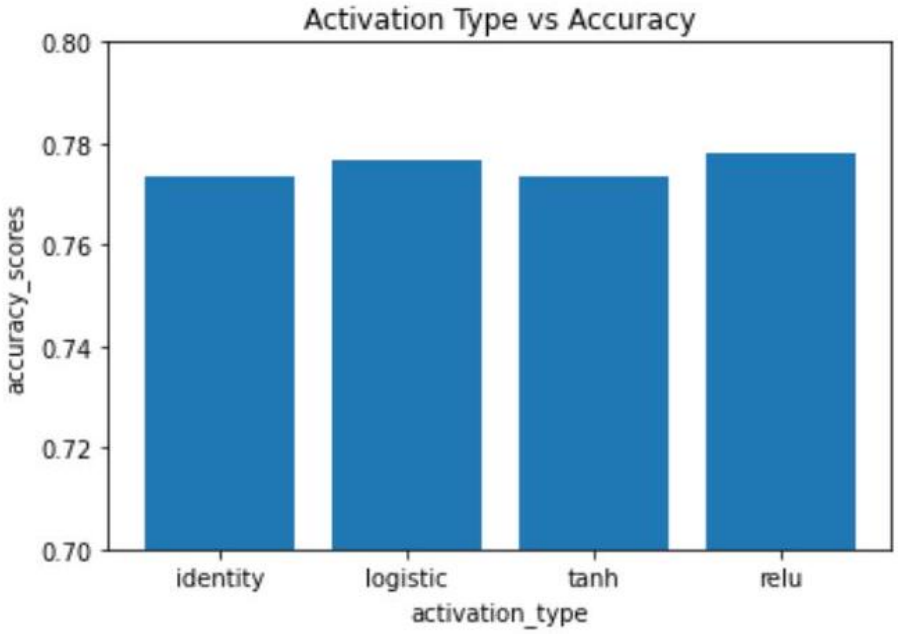


Figure 14: Effect of Activation Type on Accuracy

7 Stem Player Business Problem:

Kanye West will be taking his Donda 2 album off of all streaming platforms and selling it exclusively to be played off his product the Stem Player. Stem Player is both a streaming device as well as a physical product that can be purchased. This product is named after a stem which in the world of music is used for audio production. The device can be used for any song as it has the bluetooth feature. The device is being sold for $200, however if people want to listen to Donda 2 without that steep price it is likely to be for sale on the Stem Player platform as well. The reason that Kanye has decided to take action and release his music on Stem Player is an aim at breaking free from the "oppressive system" that the music industry is in. The point that Kanye is making is that artists are making small sums for the music that they are making as the music industry is taking more money. To quote Kanye, "We set our own price for our art. Tech companies made music practically free so if you don't do merch sneakers and tours you don't eat" (West Speaking on Donda 2). Even though what Kanye says is a good point, what he must have is the public's support for his idea. Our group decided to run sentiment analysis on tweets to gauge how the public felt about Stem Player.

8 Stem Player Data Analysis:

We predicted that Kanye West's Stem Player was receiving negative dissent, as consumers have to spend $200 just to listen to an album, rather than just downloading it on Apple Music or Spotify. To test our prediction, we collected recent Twitter data on the topic. To collect the data, we ran searches using four keywords: Stem Player, #Stemplayer, Donda2,, and #Donda2. All of the data was inserted into the cloud through MongoDB. This ensured that the data could be easily stored and queried. Then, using the MLP Classifier that we trained on movie reviews, we conducted a sentiment analysis on the Stem Player tweets. We compiled the results of 578 unique tweets and took the average of them, returning a value of .204. This indicated that the overall sentiment for Stem Player was quite low as the classifiers scale ranges from 0 to 1. The breakdown of tweets given a value of 0 vs 1 is shown in Figure 15.

Our results would have been more accurate if we had trained the model using tweets rather than movie review data. As we had no training data regarding twitter data, we decided to still use this movie review model as a baseline. Given more time, we would create a MLP Classifier model trained on twitter data and collect more data to use for training and cross validation testing.

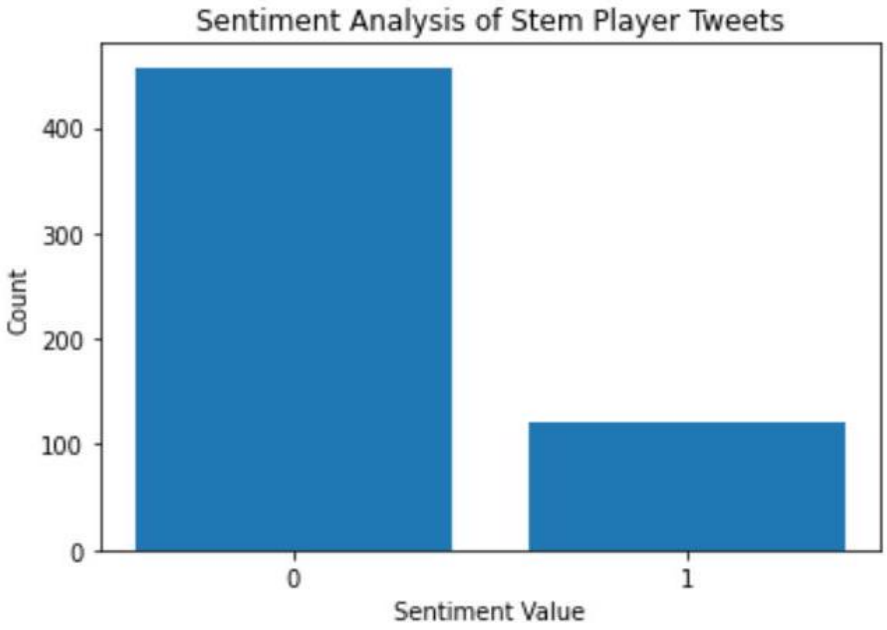


Figure 15: Sentiment Analysis of Stem Player Tweets. A 0 represents negative sentiment, while a value of 1 represents positive sentiment.

9 Stem Player Proposed Business:

After looking at our results from the Twitter data that was gathered, it was clear to see that the public was not fully supportive of Kanye's Stem Player platform. Even though what Kanye is doing is not backed by the public, our group feels that there is value to be gained here and ways that this can be modified. For our business we are aiming to open an NFT Marketplace that will specifically trade artists music on the blockchain. Recalling what Kanye had said, "we set our own price for our art". He had a great point as the music that an artist makes is a form of "art" that attracts large fanbases. The question then becomes how can artists set a price for their music and appeal to the public. The answer is an NFT Marketplace. NFT stands for non-fungible token which signifies originality and allows for ownership of items virtually. This applies to music too, what artists can do is sell their albums with unique serials to signify originality. This will allow them to sell as much at the price that they want to. Hence, achieving the idea of setting their own price that Kanye envisioned. Our company will also be profitable as we will run like standard NFT Marketplaces. This means that our business will take 2% of every transaction. This is exactly what OpenSea's marketplace does and we decided to even drop the percentage by 0.5% to make it more enticing.

10 Proposed Business Appeal:

With NFT being a big buzz term at the moment, now is the perfect time to release something like this. Our team believes that the public will have more appeal to an idea like this compared to Kanye's Stem Player. This will allow fans of these artists to invest in them. The big idea here is that this will be a stock market for music. Imagine buying Apple at $5 a share. The same idea is relevant here. Imagine buying Songs About Jane by Maroon 5 as a $5 NFT. This would have a similar effect as that album is a classic and would gain value to other collectors if it was sold in the future. In addition to this if artists wish to pull their catalogs off of other streaming platforms (like Kanye with Donda 2) this would allow an easy way to do this. Artists would then be able to set their own prices and our NFT Marketplace would take a mere fraction of the transactions. This is an untouched market as there is no way to effectively invest in artists currently. With our NFT Marketplace, you can become the first to discover the next upcoming artists and support their careers. Rather than an oppressive music system, the public will become the agents. In the future we plan to run Twitter sentiment analysis on the public's opinion of our business in order to make sure we solved the issues and gained support that Kanye didn't have. Cloud databases such as MongoDB Atlas will allow us to read tweets in and conduct searches to gather the data in an efficient fashion.

11 Limitations:

The biggest limitation that our group faced when discussing our business problem is that there is no true set of data that could be used for sentiment analysis on this topic yet. To get a proper set, there would need to be human classification to determine the sentiment of the tweets and because the topic is so new, that dataset has yet to be developed. We would also need to gain the interest of other artists to buy into the idea of selling their music as NFTs. If other artists do not get on board, the idea is unlikely to gain traction in the mainstream media.