CIS 4190/5190: Applied Machine Learning | Fall 2024

News Title Classification Model

Team Members: Kevin Dong, Richard Zhang | *Project: News Source*

1 Summary

Text Classification is an important Natural Language Processing task that has an extremely broad application. When it comes to the intersection between Media Studies and Data Science, existing literature often focuses on using text classification to classify news into different categories, [4] [5] or detecting fake news. [3]. News classification, however, can also help answer some persistent questions, such as whether the media only focuses on reporting certain types of news, or whether they intentionally create biased news titles to mislead its readers and attract attention. If the media is impartial, a text classification model that intends to classify titles to each media should not provide a significant result. The existence of a text classification model that can accurately distinguish between the titles of the two media will support the hypothesis that the media presents biased news titles. Our goal is to examine the correctness of this hypothesis by exploring various text classification models.

This project uses a dataset that contains news titles from Fox News and NBC News. In addition to the provided small dataset, we further expanded it with more data spanning from 2022 to 2024 to make our data more representative. We assume that news titles from Fox and NBC may expose a difference since the two media are often thought to have a strong, opposite political inclination. We aim to see if it is possible to train a title classification model that could accurately distinguish the titles from the two media.

To validate our hypothesis, we have trained a handful of classification models, from the traditional Machine Learning approaches such as Logistic Regression and Random Forest, to some advanced Deep Learning models such as BERT, DistilBERT, and RoBERTa. To make our model more robust, we have also conducted hyperparameter tuning and K-fold cross-validation to help combat potential overfitting issues. Among our models, DistilBERT and BERT achieve the highest prediction accuracy with 85.01% and 85.94%, respectively. In addition, we have discussed the effect of increasing dataset size on model performance, which provides some meaningful insights into our research question, while exploring possibilities of combining BERT models to form a stacking ensemble to further boost the model accuracy.

2 Core Components

2.1 Data Collection

For this project, we have three different datasets: the first one is provided and includes 3,804 titles from both news websites, the second one is a medium-large dataset that includes a total of 29,284 titles, and the third one is the large dataset with 91,063 titles. The data collection and normalization process is as follows, where the first 2 steps are utilized to create our additional dataset:

1. **URL Fetching**: The first step is to gather URLs from the News Website. For Both news media, we retrieved URLs from their Sitemap Webpage. Here are the two APIs we mainly used:

   ```
   https://www.foxnews.com/sitemap.xml?type=articles&page=<pageNum>
   https://www.nbcnews.com/archive/articles/<year>/<month>
   ```

   We used the above URLs to access the news archive and acquire the URLs for fetching.

2. **Title Fetching**: After acquiring the URLs, we attempted to fetch each page's title. There are two methods to do so: we could directly get the title by parsing the URL itself, or we could fetch the page and capture the h1 tag page title. When performing title fetching, we combined these two methods to

ensure no missing data.

3. **Data Normalization**: Upon acquiring the titles, we implemented several normalization methods, such as converting characters to lowercase, stemming the words by using a PorterStemmer, and removing common stopwords. We have also created a dataset that normalized the data with a word lemmatizer to compare its performance to a stemmer. At this step, we also convert the news outlets to binary labels (0 for NBC, 1 for FOX) as well as split the dataset to train, validate, and test sets. A different split version with only train and test is also provided for testing with K-fold cross-validation.

When training, We often split the data into provided, extra, and combined datasets, where the first contains the provided 3,804 entries, the second the medium-large or large dataset that we crawled, and the third one with the combination of the first two. We would compare the results among different dataset configurations when testing our hypotheses.

2.2 Model Considerations and Design

In our model design process, we build our final model by iteratively testing various machine learning and deep learning approaches. Regarding the design considerations, we prioritized accuracy to ensure reliable classification while minimizing the risk of over-fitting. When testing, we normally compare the accuracy results of running a single model on our different datasets and yield the highest one.

For all the 13 models we used, we considered why each model might perform better than the others, evaluated its effectiveness, and made improvements to the original design. We started with simple traditional machine learning models like logistic regression and SVM. Later, we tried more advanced word embedding methods like Word2Vec and trained complex Deep Learning models like dense neural networks (DNN), LSTMs, customized transformers, and pre-trained transformer models like BERT to capture the deeper patterns in the data and the relation among words. We also tried ensemble methods. After many tests, hyper-parameter tunings, and k-fold validations, we finalized DistilBERT as our final model because it provided the best performance with an accuracy of 85.01% while maintaining good stability. This graph 2 illustrates what we explored, and the chart of the accuracies of all the 13 trained models is here 8.

2.2.1 Traditional Models with TF-IDF Encoding

To begin with, we used TF-IDF vectorization to convert text data into numerical features because TF-IDF can effectively capture words' importance. Initially, a logistic regression model achieved 69.25% accuracy with default parameters. By removing noisy words and down-weight commonly used words, accuracy improved to 79.89%. Extending the model to capture richer contextual patterns raised accuracy to 80.03%.

Finding it hard to further improve logistic regression, we turned to Support Vector Machines (SVMs), which handle high-dimensional data like TF-IDF well. Using LinearSVC, we achieved an accuracy of 81.60%. Because it's hard for SVMs to capture non-linear patterns in the text, we decided to explore random forests as it is good at capturing non-linear relationships. However, the Random Forest model didn't perform as well as expected, only achieving 78.98%, likely due to its inability to handle sparse, high-dimensional TF-IDF features. Given the multinomial nature of text data, we tested a multinomial Naive Bayes classifier, achieving 80.29% accuracy. K-Nearest Neighbors (KNN) struggled with sparse TF-IDF features, yielding an accuracy of only 77.14%. This led us to consider methods that utilize the contextual information better.

To find the deep non-linear relationship among words, we moved to a feed-forward neural network with TF-IDF, reaching an accuracy of 82.39%, the highest at that time, and we used this as our midterm leader-board submission model. Even though we experimented with adding dense layers, performing hyper-parameter tuning, and retraining the model with extra datasets, the model's accuracy consistently fluctuated between 78–82%. This implied the limits of DNNs with TF-IDF, so we decided to explore more advanced sequence-based models, such as LSTMs with Word2Vec and transformers, to tackle this problem. The structure of our DNN model is here 3.

2.2.2 Ensemble

Before exploring other deep learning methods, we tried ensemble approaches. Using a hard majority voting ensemble with KNN, NB, RF, SVC, LR, and DNN, we achieved an accuracy of 81.60%. Another stacking ensemble with the same base models using an LR model as the meta-classifier had a similar accuracy of 81.34%. The ensemble approaches show limited improvement over our best model. Also, we considered using AdaBoost, but because it only supports a single model type, we decided against it.

2.2.3 LSTMs and Word Embeddings

Knowing text data inherently follows a sequential structure, so we used Bidirectional Long Short-Term Memory (LSTM) with tokenization to train a new model, achieving 81.21% accuracy, which is even lower than our DNN model. To further enhance the model, we adopted Word2Vec embeddings, which can generate dense vectors that encode the semantic relation among words, to train our model, but we only got 64.26% accuracy. We suspected this under-performance was due to the limited dataset we used limited the quality of learned embeddings. Using GloVe embeddings improved accuracy to 76.16%, but it's still unsatisfying, motivating us to explore new approaches.

2.2.4 Transformer, Pre-trained Models, and our Best Model

To overcome the limitations of previous models, we turned to transformer-based architectures for their self-attention mechanisms to offer deeper contextual understandings. We implemented a transformer block with multi-head attention and feed-forward layers, achieving an accuracy of 80.81%. This result encouraged us to use pre-trained transformer models, such as BERT, RoBERTa, and DistilBERT, to improve it as they were trained on massive text and would capture more general language understanding.

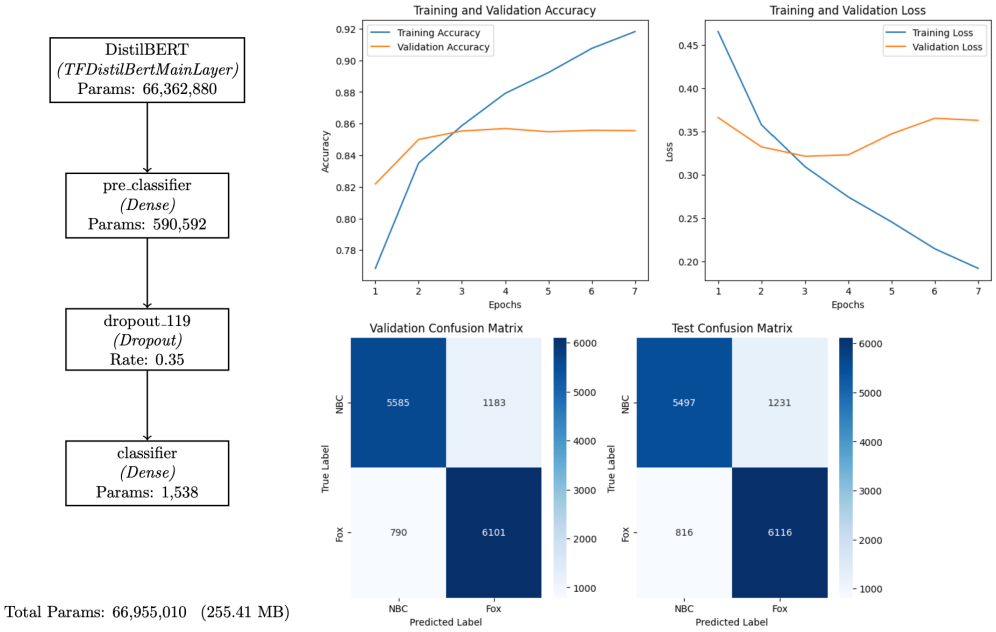
We first tried bidirectional encoder representations from transformers (BERT) and DistilBERT, getting an accuracy of 84.63% and 82.36%, respectively. After k-fold cross-validation, the initial BERT's mean accuracy was 80.65% ($\pm$0.01113), while the initial DistilBERT's mean accuracy was 82.02% ($\pm$0.0210), proving the models' robustness 6. In hyper-parameter tuning, because random search could miss optimal configurations, we used grid search as it provided more reliable and systematic results. After tuning, BERT and DistilBERT achieved an accuracy of 85.94% and 85.01%, respectively, dramatically outperforming than earlier approaches we built. We also tried RoBERTa, but it only achieved 84.12% accuracy, which was lower than BERT and DistilBERT, and was not pursued further.

After tuning, BERT and DistilBERT had similar accuracies. To make a final decision, we tested them on the 20 rows of data released by the TA. DistilBERT achieved an accuracy of 85%, whereas BERT only got 80%. Given Distill's stronger performance, robustness, and higher computational efficiency, we selected it as our final model. Our final model design is shown below 1a. The final model configurations are: (1) Dropout rate is 0.35 for both general dropout and attention dropout. (2) Optimizer is AdamWeightDecay with a learning rate of 3×10^{-5}. (3) Batch size is 32; number of epochs is 7.

2.3 Evaluation and Model Performance

We applied a set of evaluation protocols throughout the model design process to have reliable comparisons across models. Internally, we rely on accuracy scores, classification reports, and k-fold to assess models. Accuracy was our primary metric, while precision, recall, and F1 scores provided insights into class-specific performance. K-fold was essential in validating the models' robustness. Also, based on the leader-board grade, our DNN model only got 78.36% accuracy, 3-4% lower than our internal testing results. To address this, we incorporated a validation dataset into all the models' training processes.

When evaluating our final DistilBERT model, we utilized accuracy, precision, recall, and the confusion matrix as our primary performance metrics. As mentioned above, our final model's test accuracy is 85.01%, and its validation accuracy is 85.56%. From the k-fold cross-validation results 9, the model achieved a mean accuracy of 82.73% with a standard deviation of 0.0184, indicating the model performed well across folds with minimal variation. From the training and validation accuracy graph (Figure 1b), training accuracy



(a) Final Model Design

(b) Final Model Evaluation Results

Figure 1: Final Model

steadily increased to 92%, while validation accuracy plateaued around 86%. This divergence indicates potential overfitting, where the model continues improving on the training data but fails to generalize further. The slightly fluctuating validation loss curve reinforces this observation. Additionally, the plateau at 86% validation accuracy may suggest a performance ceiling for this task, reflecting either the inherent difficulty of differentiating Fox and NBC articles or the model's limited capacity to capture deeper distinctions.

The confusion matrices provide further insights into the predictions. While most labels are correctly classified, a higher number of NBC News samples are misclassified as Fox News compared to the reverse. Given the balanced and sufficiently large dataset, this trend could indicate that NBC News articles share subtle stylistic or topical similarities with Fox News, making them more challenging for the model to distinguish effectively. Another potential explanation is that, although the number of titles in the two classes is balanced, the NBC data comes from the recent three years while the FOX data only comes from the recent two years. The longer period for NBC may introduce a greater diversity of topics and styles into the NBC dataset, increasing intra-class variability. As a result, the model could struggle to identify consistent patterns within the NBC class, leading to a higher mis-classification rate. Conversely, the Fox data, being limited to a shorter time-frame, may exhibit less variability, allowing the model to learn more uniform patterns for that class.

3 Exploratory Questions

3.1 Performance Comparison with Different Dataset Sizes

Question: Will increasing the dataset size significantly improve performance? Our news classification models operate under the assumption that the two media outlets, Fox and NBC, tend to report with distinct perspectives. From a machine learning perspective, as discussed in class and supported by numerous studies in the literature, increasing the dataset size typically reduces variance, helps prevent overfitting, and enhances generalization. This should lead to better predictive performance. However, if the data itself inherently does not pose a significant difference, prediction accuracy will plateau even if we continue to add more data. To validate this hypothesis, we propose comparing the performance of the same model trained and tested on datasets of varying sizes.

Approach: We utilized three datasets collected during the data acquisition process, containing 3,804, 29,284, and 91,063 entries, respectively. To benchmark performance and minimize confounding variables, we combined the training, validation, and test splits of each dataset in a vertical stacking approach when transitioning from one dataset to another. Our primary model was DistilBERT, which we trained with a fixed set of hyperparameters (dropout = 0.5, batch size = 32, learning rate = 2e-5, epochs = 7).

Result: When training with DistilBERT, we observed similar testing and validation accuracy trends across the three datasets. As shown in Figure 4, all three models achieved a training accuracy of approximately 82% while the provided dataset achieved a much lower validation accuracy of approximately 77%, indicating a strong overfit. These findings support the hypothesis that increasing dataset size improves the representativeness of the data, thereby enhancing the model's generalization capability. However, the diminishing returns observed between the medium-large and largest datasets suggest that beyond a certain point, increasing dataset size yields minimal performance gains.

Overall, we observed a slight performance improvement as the dataset size increased. For complex models such as DistilBERT, larger datasets appeared to aid the model in capturing subtle differences and achieving better generalization, but it seems that the increase is very limited, indicating that there is just not an extremely strong difference between Fox and NBC. However, for simpler models like the DNN, the benefits of increasing dataset size were less pronounced, suggesting limitations in the model's capacity to represent the data effectively.

3.2 BERT Stacking Ensemble for Title Classification

Question: Does creating a BERT stacking ensemble further increase the accuracy? BERT models can easily overfit due to their high complexity. To lower the variance and potentially boost accuracy, recent advancements in text classification have adopted stacking ensembles for BERT-like deep learning models, which combine the outputs of multiple models to enhance overall performance [1]. Therefore, we hypothesize that using a stacking ensemble with BERT models would result in a performance increase, as it reduces the impact of individual model biases and variations, thereby yielding a more robust and accurate prediction system.

Approach: We took our best-performing model, the DistilBERT model, and attempted to form an ensemble by leveraging a stacking technique. The ensemble approach involved training multiple DistilBERT models with varying hyperparameters to introduce model diversity and applied out-of-fold (OOF) predictions with k-fold cross-validation. The predictions will serve as the input features for the meta-learner, which will act as the second layer of the ensemble. The meta-learner was tasked with learning from the aggregated predictions of the base models to produce the final classification. For comparison, we experimented with many meta-learners, such as Logistic Regression, simple neural networks, and random forests. These meta-learners were trained on the OOF predictions and evaluated on validation and test datasets to assess their performance.

Result: Using OOF = 5 and the number of base models = 5, our random forest meta-data classifier provides a 82.6% accuracy, logistic regression provides a 82.07% accuracy, and the single layer neural network provides 82.6%. As we increase the number of DistilBERT models that we train for the model, we can see an increase in the model accuracy. Specifically, when using 25 base models, the accuracy increases to 85.06%. We would imagine that, as we increase the ensemble size while ensuring enough diversity, we would be able to achieve even higher results. Besides adjusting the ensemble size, there are also newly proposed meta-data classifier structures that would better learn patterns from the ensemble, which may further boost the ensemble's accuracy. [2]

4 Team Contributions

Kevin Dong handled data collection, cleaning, and traditional model training. Richard Zhang led the design of other models. Both Kevin and Richard explored key questions and co-wrote the report.

5 Suggestions for future iterations of this project

- **Collect more data.** We would like to crawl even more data to create a comprehensive, representative dataset that could actually represent the two media. In that case, we may get a different testing result since the data is more generalizable. If the accuracy is still not very high in this case, it may suggest that there is no significance between FOX and NBC in terms of what kind of news they report and how they frame their questions.
- **Try a domain-specific BERT model.** There are several BERT models that are trained specifically for news classification. We may do a transfer learning to further train upon this pre-trained model to achieve a better result since the model has already been familiarized with new classification tasks.
- **Design larger and more diverse ensemble.** Hypothetically, increasing more base models in an ensemble may further reduce the overall variance and potentially give us a better result. We can try to form an ensemble with different variations of BERT to see if the prediction can go higher.

References

[1] S. Abarna, J. I. Sheeba, and S. Pradeep Devaneyan. An ensemble model for idioms and literal text classification using knowledge-enabled bert in deep learning. *Measurement: Sensors*, 24:100434, 2022.

[2] Ammar Mohammed and Rania Kora. An effective ensemble deep learning framework for text classification. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part A):8825–8837, 2022.

[3] Dhiren Rohera, Harshal Shethna, Keyur Patel, Urvish Thakker, Sudeep Tanwar, Rajesh Gupta, Wei-Chiang Hong, and Ravi Sharma. A taxonomy of fake news classification techniques: Survey and implementation aspects. *IEEE Access*, 10:30367–30394, 2022.

[4] Pramod Sunagar, Anita Kanavalli, Sushmitha S. Nayak, Shriya Raj Mahan, Saurabh Prasad, and Shiv Prasad. News topic classification using machine learning techniques. In V. Bindhu, João Manuel R. S. Tavares, Alexandros-Apostolos A. Boulogeorgos, and Chandrasekar Vuppalapati, editors, *International Conference on Communication, Computing and Electronics Systems*, pages 461–474, Singapore, 2021. Springer Singapore.

[5] Fan Zhang, Wang Gao, and Yuan Fang. News title classification based on sentence-lda model and word embedding. In *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pages 237–240, 2019.

A Appendix

A.1 Model Design Process

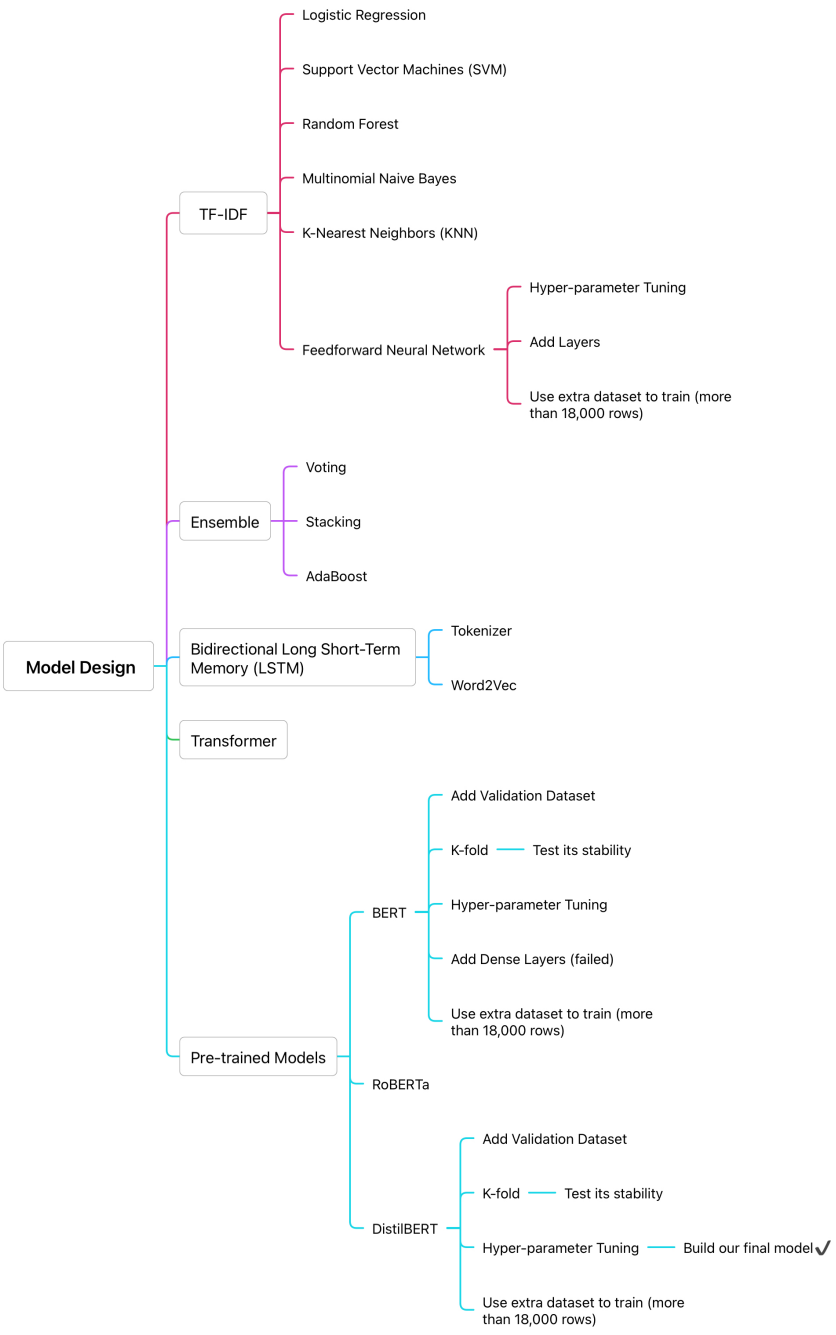


Figure 2: Model Design Process Visualization

A.2 Revised DNN

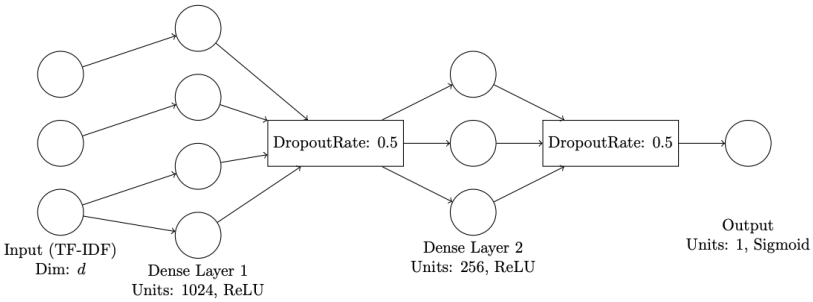


Figure 3: Revised DNN Visualization

A.3 Training and Validation Accuracy on the Three Datasets

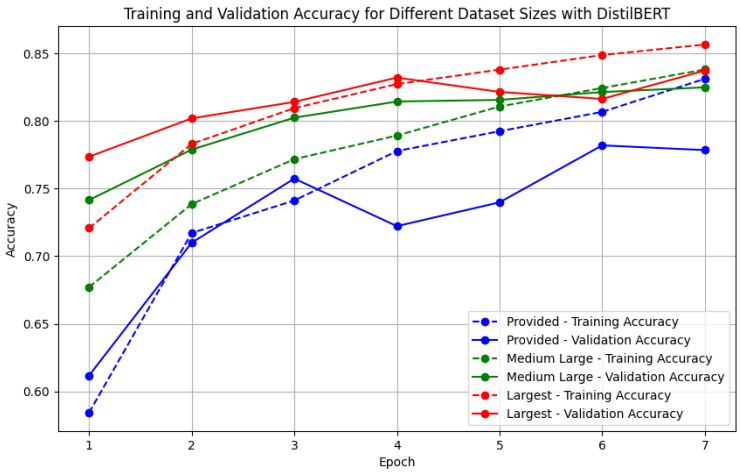


Figure 4: Training and Validation Accuracy Running DistlBERT on the Three Datasets

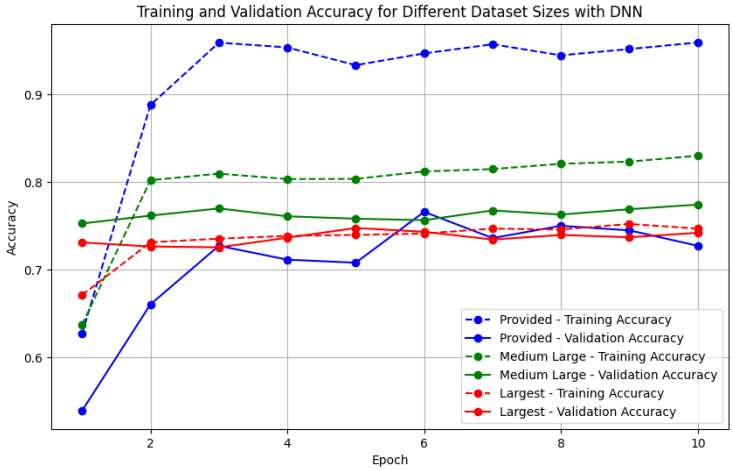


Figure 5: Training and Validation Accuracy Running DNN on the Three Datasets

A.4 BERT and DistilBERT K-fold Results

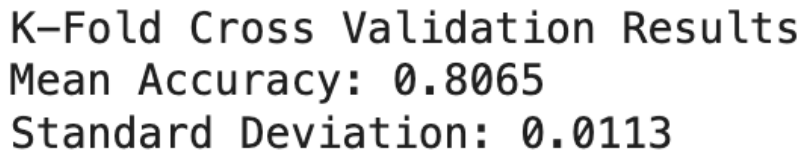


Figure 6: BERT K-fold Results

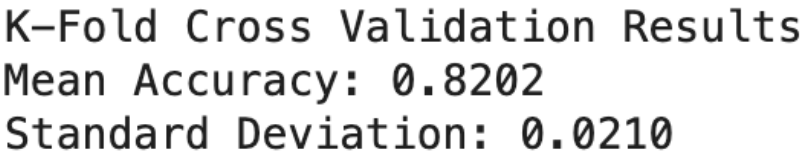


Figure 7: DistilBERT K-fold Results

A.5 Testing Accuracies Comparison

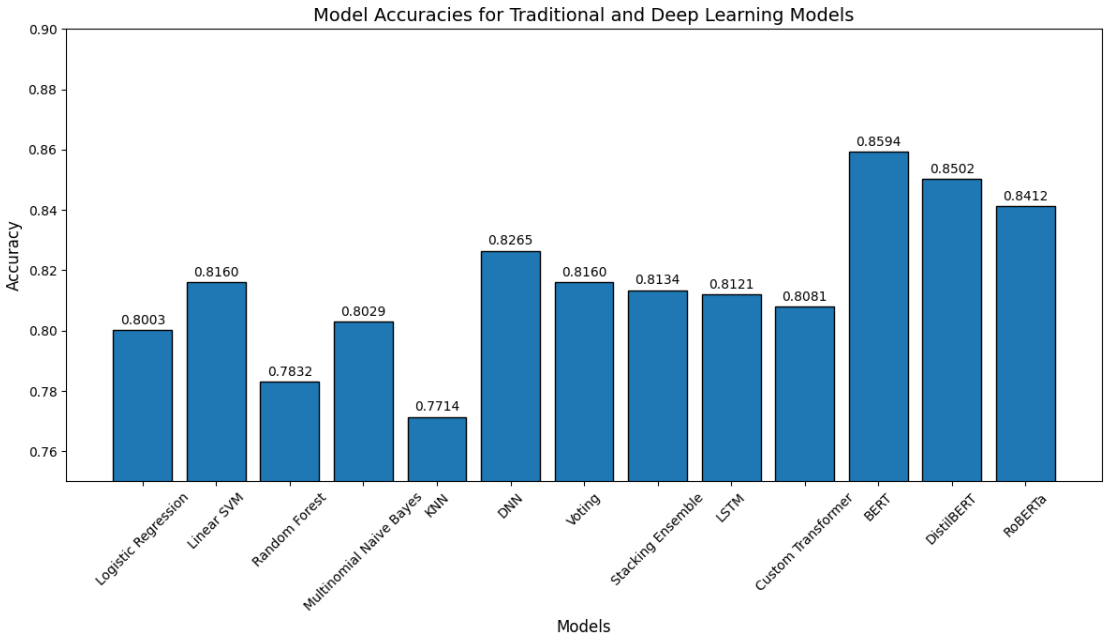


Figure 8: Testing Accuracy Comparisons Among the 13 Trained Models

A.6 Final DistilBERT K-fold

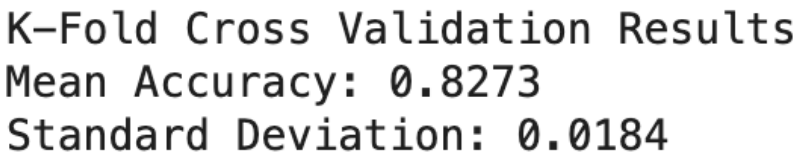


Figure 9: Final DistilBERT K-fold Results