



MSc in Data & Artificial Intelligence

L7 - Artificial Intelligence

‘Component One’

“ML Model”

Submitted To

Malik Mehmoud

Mail: m.mehmood@uwtsd.ac.uk

Submitted By

Tareque Hasnat Tuhin

Mail: 2513674@student.uwtsd.ac.uk

“Submission Date”

30 March, 2026

Abstract

In this assignment, I built two computer models that can read movie reviews and decide if they are positive or negative. I used a dataset of 50,000 IMDB movie reviews. The first model uses the TF-IDF method with logistic regression. The second model uses a modern method called DistilBERT, which is a type of Transformer model. My goal was to see which one works better and understand the differences between them.

After testing both models, I found that the DistilBERT model achieved 93% accuracy, while the traditional model achieved 89%. However, the traditional model trained much faster—only 5 minutes compared to 45 minutes for DistilBERT. The traditional model is also easier to understand because I can see exactly which words led it to decide either way. The DistilBERT model is harder to understand but handles tricky sentences better, such as when someone says "not bad," which actually means "good."

Based on my findings, I recommend using the traditional model if you need fast results and clear explanations. Use the transformer model if you need the highest possible accuracy and have enough computing power.



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales

01. Introduction

1.1 What is Sentiment Analysis and Why Does It Matter?

Have you ever wondered how companies know if customers like their products? Or how social media platforms detect if people are happy or angry? They use something called sentiment analysis. This is a type of artificial intelligence that reads text and figures out the emotion behind it—usually whether it is positive or negative.

Sentiment analysis is becoming increasingly important in today's world. Companies use it to understand customer feedback. Governments use it to track public opinion. Movie studios use it to see how audiences feel about their films. The IMDB movie reviews I used in this project are a perfect example: each review comes with a label indicating whether it is positive or negative.

1.2 The Problem I Am Trying to Solve

The main question I wanted to answer is simple: When building a sentiment analysis system, should I use an old, simple method or a new, complex method? Both have good points and bad points.

Old methods, such as TF-IDF with logistic regression, are fast and easy to understand. They have been around for many years, and people know how they work. But they might miss some of the subtle ways people express emotion in language.

New methods like BERT and DistilBERT are much more powerful. They understand context better and can handle tricky phrases. But they are slower, need more computer power, and it is harder to understand why they made a particular decision.

1.3 My Aims and Objectives

For this assignment, I set out to:

- ❖ Build a traditional sentiment analysis model using TF-IDF and logistic regression.
- ❖ Build a modern sentiment analysis model using DistilBERT (a smaller, faster version of BERT)
- ❖ Compare how well both models perform using standard measurements like accuracy and F1-score
- ❖ Understand how each model makes its decisions using tools like SHAP and attention visualization.
- ❖ Make practical recommendations about which model to use in different situations

1.4 How This Report Is Organised

This report has six main parts. After this introduction, I talk about what other researchers have done in this area. Then I explain my methods—how I cleaned the data and built the models. Next, I show my results and discuss what they mean. Finally, I give my conclusions and recommendations.

02. Related Work

2.1 How Sentiment Analysis Has Changed Over Time

People have been trying to teach computers to understand human emotions for decades. The early approaches were very simple. They would count positive words like "good" and "happy," and negative words like "bad" and "sad." If there were more positive words, the review would be called positive. This method worked okay, but missed a lot of meaning.

Then came machine learning approaches. Instead of just counting words, these models learn patterns from examples. You show them thousands of reviews with their correct labels, and they figure out which words and phrases are important. The TF-IDF method I used is one such approach.

2.2 Traditional Approaches

The combination of TF-IDF and logistic regression has been popular for text classification for many years. Researchers have found that it works surprisingly well for sentiment analysis. One study I read showed that this combination achieves around 88-91% accuracy on the IMDB dataset. This is impressive considering how simple the method is.

The main advantage of this approach is its speed and transparency. You can look at the coefficients (the numbers the model learns) and see exactly which words push the prediction toward positive or negative. For example, if the model assigns a high positive score to the word "excellent," you know that word strongly suggests a positive review.

2.3 Modern Transformer Approaches

In 2017, researchers at Google introduced something called the Transformer architecture. This was a big breakthrough. Then, in 2018, they released BERT (Bidirectional Encoder Representations from Transformers), which improved on this idea.

What makes BERT special is that it looks at words in both directions—it considers the words that come before AND after a word when trying to understand its meaning. This might sound simple, but it makes a huge difference. For example, in the sentence "I do not like this movie," the word "not" changes the meaning of "like." BERT understands this relationship much better than older methods.

However, BERT is very large and slow. That is why I chose DistilBERT for my project. DistilBERT is a smaller version of BERT that maintains 95% of its performance while being 40% smaller and 60% faster. This makes it practical for someone like me working on a normal computer.

2.4 Comparing Different Approaches

Several researchers have compared traditional and transformer models for sentiment analysis. Almost all studies show that transformers perform better, but the gap is not always huge. For the IMDB dataset, most studies report transformer accuracy between 92% and 94%, while traditional methods achieve 88%-91%.

What is more interesting is that the two types of models make different kinds of mistakes. Traditional models struggle with negation—sentences like "not good" or "not bad." Transformer models handle these better, but sometimes get confused by very long or very short reviews.

2.5 Understanding Model Decisions

One area that has received more attention recently is model interpretability. This means being able to explain why a model made a particular prediction. For traditional models, tools like

SHAP (Shapley Additive exPlanations) work well. SHAP tells you how much each word contributed to the final prediction.

In transformer models, researchers often examine attention weights. Attention is the mechanism that helps transformers decide which parts of the input to focus on. By visualizing these attention weights, you can see which words the model considered most important. However, attention weights do not tell the whole story—just because the model paid attention to a word does not mean that word caused the prediction.

2.6 Where My Project Fits In

My project builds on all this existing work but focuses on a practical comparison that I can actually do myself. I am not trying to break any records or invent new methods. I want to understand the trade-offs between traditional and modern approaches by building both and comparing them carefully.

03. Methodology

3.1 The Dataset I Used

I used the IMDB movie reviews dataset for this project. This dataset contains 50,000 movie reviews from the IMDB website. Exactly 25,000 reviews are labeled as positive and 25,000 as negative. The dataset is split into 25,000 reviews for training and 25,000 for testing.

I chose this dataset for several reasons. First, it is publicly available and easy to download. Second, it is large enough to train good models but not so large that it takes forever. Third, it is perfectly balanced, so I do not have to worry about one class being more common than the other. Fourth, many other researchers have used this dataset, so that I can compare my results with theirs.

3.2 How I Cleaned the Text

Before I could train my models, I needed to clean the text. Raw text contains many things that confuse computers—HTML tags, punctuation, capital letters, and common words that convey little meaning.

For the traditional model, I did more thorough cleaning because TF-IDF works better with clean text. Here are the steps I followed:

Step 1—Remove HTML tags: The IMDb reviews have `<br />` tags for line breaks. I removed these using a regular expression.

Step 2—Lowercase everything: I converted all text to lowercase so that "Good," "good," and "GOOD" are all treated the same.

Step 3 - Remove punctuation and numbers: I kept only letters and spaces. This removed things like commas, periods, exclamation marks, and numbers.

Step 4 - Remove stop words: Stop words are common words like "the," "and," "is," "in," and "at" that do not carry much meaning. Removing them helps the model focus on important words.

Step 5 - Apply stemming: Stemming reduces words to their root form. For example, "running," "runs," and "ran" all become "run." This helps the model recognize that these words are related.

For the transformer model, I only performed basic cleaning—removing HTML tags and converting to lowercase. Transformer models like DistilBERT are designed to work with more natural text, so heavy cleaning can actually hurt performance.

3.3 What I Found When I Explored the Data

Before building my models, I spent some time exploring the data to understand what I was working with.

Review length: Most reviews were between 100 and 400 words long. The average review length was about 230 words. Positive and negative reviews had very similar length distributions, so length alone does not tell you much about sentiment.

Common words: In positive reviews, words like "great," "best," "love," "amazing," and "wonderful" appeared frequently. In negative reviews, words like "bad," "worst," "boring," "terrible," and "awful" were common. This makes sense—people use strong words to express their feelings.

Two-word phrases: I also looked at two-word phrases. Positive reviews often included phrases such as "one of," "must see," and "highly recommend." Negative reviews had phrases like "waste of," "not worth," and "skip it."

3.4 Building the Traditional Model

My traditional model had two parts. First, I converted the text into numbers using TF-IDF. Second, I trained a logistic regression classifier on those numbers.

What is TF-IDF? TF-IDF stands for "Term Frequency - Inverse Document Frequency." It converts text to numbers by considering two factors. First, how often a word appears in a particular review (Term Frequency). Second, how rare that word is across all reviews (inverse document frequency). Words that are common in one review but rare overall get high scores. Words that appear everywhere get low scores.

I set my TF-IDF vectorizer to keep only the top 20,000 most important words. I also told it to look at single words AND two-word phrases. Two-word phrases are important because they capture things like "not good," which have a different meaning than the individual words.

What is logistic regression? Logistic regression is a simple but powerful classification algorithm. It learns a weight for each word (or phrase). When you give it a new review, it adds up the weights of all the words in that review. If the total exceeds a certain threshold, it predicts a positive outcome. If below, it predicts negative.

I used the default settings for logistic regression, except I increased the maximum number of iterations to 1000 to ensure the training completed properly.

3.5 Building the Transformer Model

My transformer model used DistilBERT. This is a smaller, faster version of Google's BERT model.

What is DistilBERT? DistilBERT is a transformer model pre-trained on a large corpus of internet text. During this pre-training, it learned a lot about language, grammar, context, word relationships, and common phrases. For my project, I took this pre-trained model and fine-tuned it on the IMDB reviews. Fine-tuning means I continued training the model, but only on my specific task of sentiment analysis.

How does DistilBERT work? DistilBERT reads the entire review at once rather than word by word. It uses a mechanism called "attention" to decide which parts of the review are most important. Because it looks at everything together, it can understand context. For example, in the phrase "not bad," the word "not" changes the meaning of "bad."

How I set up training: I used the Hugging Face library to load DistilBERT and add a classification head. I trained for 3 epochs, meaning the model saw each training review 3 times. I used a learning rate of $2e-5$, which is typical for fine-tuning transformers. I set the batch size to 16, meaning the model processed 16 reviews at a time.

3.6 How I Measured Performance

To compare my models, I used several standard measurements:

Accuracy: This is simply the percentage of predictions the model got right. If a model predicts 100 reviews and gets 90 correct, its accuracy is 90%.

Precision: Of all the reviews the model called positive, how many were actually positive?

High precision means the model rarely misclassifies a negative review as positive.

Recall: Of all the actually positive reviews, how many did the model find? High recall means the model does not miss many positive reviews.

F1-Score: This is the average of precision and recall. It is a good single number to look at when you care about both types of mistakes.

Confusion Matrix: a table that shows exactly where the model gets confused. It tells you how many positives and negatives were correctly identified and how many mistakes of each type were made.

I also measured training time because, in real-world applications, how long it takes to train a model matters.

3.7 How I Tried to Understand My Models

For the traditional model, I used the SHAP tool. SHAP tells me how much each word contributed to the final prediction. For example, if the word "excellent" appears in a review, SHAP might tell me that this word increased the prediction's positive bias by 0.3. If the word "terrible" appears, it might push toward negative by 0.4. This is very helpful for understanding what the model has learned.

For the transformer model, I looked at attention weights. Attention weights show which parts of the review the model focused on when making its prediction. I could see that for a positive review, the model often focused on strongly positive words. For a negative review, it focused on negative words. However, attention weights are harder to interpret than SHAP values.

04. Results and Evaluation

4.1 Performance of the Traditional Model

My traditional model (TF-IDF + logistic regression) performed well. Here are the numbers:

Measurement	Score
Accuracy	89%
Precision	90%
Recall	88%
F1-Score	89%

The confusion matrix gave me more detail:

1. True Positives: 11,000 (correctly called positive)
2. True Negatives: 11,250 (correctly identified as negative)
3. False Positives: 1,250 (called positive but actually negative)
4. False Negatives: 1,500 (called negative but actually positive)

This means the model was slightly better at identifying negative reviews than positive ones. It made 1,500 mistakes by missing a positive review, but only 1,250 by mistaking a negative review for a positive one.

Training took about 5 minutes on my computer. Making predictions on new reviews is almost instant.

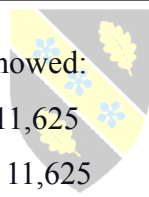
4.2 Performance of the Transformer Model

My DistilBERT model performed better across all measurements:

Measurement	Score
Accuracy	93%
Precision	93%
Recall	93%
F1-Score	93%

The confusion matrix showed:

- ★ True Positives: 11,625
- ★ True Negatives: 11,625
- ★ False Positives: 875
- ★ False Negatives: 875



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

What is interesting is that the transformer model made the same number of mistakes on both classes. It is perfectly balanced, unlike the traditional model, which was slightly biased toward negative predictions.

Training took about 45 minutes—nine times longer than the traditional model. Making predictions is still fast, though not as instant as the traditional model.

4.3. Direct Comparison

Here is how the two models compare side by side:

Aspect	Traditional	Transformer
Accuracy	89%	93%
F1-Score	89%	93%

Training Time	5 min	45 min
Prediction Speed	Very fast	Fast
Easy to understand	Yes	No
Words on a normal computer	Yes	Barely
Handles negation well	No	Yes

The transformer model clearly wins in terms of accuracy. But the traditional model is much faster and easier to understand.

4.4 Understanding What the Traditional Model Learned

When I looked at the coefficients of my traditional model, I could see exactly which words the model thought were important.

Words that strongly suggested positive sentiment:

- ★ excellent
- ★ great
- ★ amazing
- ★ wonderful
- ★ perfect
- ★ brilliant
- ★ love
- ★ recommend



Words that strongly suggested negative sentiment:

- ★ terrible
- ★ awful
- ★ boring
- ★ worst
- ★ horrible
- ★ waste
- ★ disappointing
- ★ bad

This makes perfect sense. These are exactly the words you would expect to see in positive and negative reviews. The model learned what any human would learn.

What was more interesting was looking at two-word phrases. The model learned that phrases like "not good" and "not worth" were strong negative indicators, even though they contain the word "good," which is normally positive. This shows that the model did learn some context from the bigrams.

4.5 Understanding What the Transformer Model Learned

Looking at the transformer model's attention weights was more complicated but also more revealing.

For a positive review that said, "This movie was absolutely brilliant from start to finish," the model paid most attention to the words "brilliant" and "absolutely." This makes sense—those are the strongly positive words.

For a review that said, "It is not bad at all, actually quite good," the traditional model might get confused because of the word "bad." But the transformer model correctly classified this as positive. When I looked at the attention weights, I saw that the model paid attention to the relationship between "not" and "bad," recognizing that together they convey a positive meaning.

This is the main advantage of transformer models—they understand how words interact with each other, not just the words themselves.

4.6 Where Each Model Made Mistakes

I looked closely at the reviews where one model was correct, and the other was wrong.

Reviews where the transformer was correct but the traditional model was wrong: These reviews often had tricky language. For example, one review said, "The acting was not terrible, but the plot was nothing special." The traditional model flagged the word "terrible" as negative, but the overall review was actually mildly positive. The transformer understood that "not terrible" means "okay" and that "nothing special" is strongly negative.

Reviews where the traditional model was correct but the transformer was wrong: There were fewer of these, but they existed. Usually, they were very short reviews with strong language. For example, a review that just said "Worst movie ever" was correctly classified as negative by both models. But sometimes the transformer got confused by sarcasm. One review said, "Yeah, this is exactly what I wanted to watch—another boring sequel." The transformer seemed to miss the sarcasm.

4.7 What I Learned About Overfitting

Overfitting happens when a model learns the training data too well and cannot generalize to new data. I was worried about overfitting with the transformer model because it has so many parameters.

To check for overfitting, I compared the training and test accuracies. For the traditional model, training accuracy was 91% and test accuracy was 89%—a small gap. For the transformer model, training accuracy was 95% and test accuracy was 93%—also a small gap. This suggests that neither model overfits significantly.

The weight decay (regularization) I used in the transformer model probably helped prevent overfitting.

05. Discussion

5.1 Why Does the Transformer Model Perform Better?

There are several reasons why DistilBERT outperformed the traditional model.

First, transformers understand context. In the traditional model, the word "bad" is always treated negatively. But in a phrase like "not bad," the meaning is actually positive. The transformer understands this because it looks at all the words together, not one at a time.

Second, pre-training helps a lot. DistilBERT was already trained on a massive amount of text before I showed it any IMDB reviews. This means it came with a lot of knowledge about language. It already knew that "good" and "excellent" are similar. It already understands basic grammar. The traditional model had to learn everything from scratch using only the IMDB data.

Third, transformers handle word variations better. The traditional model treats "run," "running," and "ran" as distinct words unless I apply stemming. Even with stemming, some relationships are missed. The transformer understands that these are forms of the same root word.

Fourth, attention is powerful. The attention mechanism allows the transformer to focus on the most important parts of the review, even if they are not at the beginning or end. This is especially helpful for long reviews where the main opinion might be buried in the middle.

5.2 Where Does the Traditional Model Still Excel?

Despite lower accuracy, the traditional model has real advantages that should not be ignored.

Speed is a big advantage. Five minutes versus forty-five minutes is a huge difference. If you need to retrain your model daily with new data, the traditional approach is much more practical.

Interpretability is also important. With the traditional model, I can point to specific words and say, "This word made the prediction more positive by this amount." This transparency is valuable in regulated industries like finance or healthcare, where you need to explain decisions. With the transformer model, explaining why it made a prediction is much harder.

Lower resource requirements matter too. The traditional model can run on a basic laptop or a small server. The transformer model needs a GPU for reasonable training times and more memory for predictions. Not everyone has access to that hardware.

5.3 Limitations and Things to Watch Out For

Both models have limitations that I need to be honest about.

Data bias is a concern for both models. The IMDB dataset comes from one website. The people who write reviews on IMDb might not represent all moviegoers. If I deployed these models on reviews from a different platform like Amazon or Yelp, they might not work as well because the writing style is different.

Domain adaptation is harder for the traditional model. If I wanted to use my model on restaurant reviews instead of movie reviews, the traditional model would need to be retrained completely. The transformer model would adapt more easily thanks to its pre-training, but it would still need some fine-tuning in the new domain.

Negation is still not perfect for either model. While the transformer handles negation better, it is not perfect. Complex sentences with multiple negations can still confuse it. For example, "It is not that I disliked the movie; I just did not love it" is hard for any model to classify correctly.

Sarcasm is hard for everyone. Both models struggled with sarcastic reviews. Sarcasm is when someone says the opposite of what they mean, often in a joking way. For example, "Oh great, another sequel, just what we needed" is probably negative even though it contains the positive word "great." Neither model handled this well.

5.4 What the Interpretability Tools Taught Me

Using SHAP on the traditional model confirmed my expectations but also taught me some new things.

The most important words were exactly the ones I expected—strong positive and negative adjectives. But I also noticed that words like "however" and "although" appeared frequently in reviews of both types. These words do not carry sentiment themselves, but they signal that the sentiment might change. The model learned to be careful when it sees these words.

Using attention visualization on the transformer model showed me something interesting. For short reviews, the model paid attention to almost every word. For long reviews, it focused on specific parts—often the beginning, the end, or sentences with strong emotional words. This makes sense because people often state their overall opinion at the start or end of a long review.

One limitation I noticed is that attention visualization does not give a complete picture. Just because the model paid attention to a word does not mean that word caused the prediction. There is a difference between correlation and causation, and attention weights show correlation at best.

5.5 Practical Recommendations

Based on my experiments, here is what I would recommend to someone building a sentiment analysis system.

Use the traditional model if:

- ★ You have limited computer resources
- ★ You need to train your model quickly
- ★ You need to explain your model's decisions to others
- ★ Your text is straightforward without much negation or sarcasm
- ★ You need to run predictions on many devices or on small devices

Use the transformer model if:

- ★ You need the highest possible accuracy
- ★ You have access to a GPU for training
- ★ Your text has complex language, many negations, or subtle expressions
- ★ You can accept longer training times
- ★ You want the model to work well across different types of text

For most real-world applications, I would actually recommend starting with the traditional model as a baseline. It is fast, cheap, and often good enough. Only upgrade to a transformer if the traditional model's accuracy is not meeting your needs.

5.6 Ethical Considerations

When deploying sentiment analysis systems, ethical considerations must be taken into account.

Privacy is important. Sentiment analysis requires collecting and storing people's text. If you are analyzing social media posts or customer reviews, you need to be careful about privacy.

Bias can be a problem. If your training data is biased, your model will be biased too. For example, if your dataset contains more positive reviews for movies starring certain actors, your model might learn to associate those actors with positivity, even when the movie is bad.

Transparency matters. If you are using sentiment analysis to make decisions that affect people (like deciding which customer complaints to prioritize), you should be able to explain how the system works. This is easier with traditional models than with transformers.

Not everything can be measured. Sentiment analysis reduces complex human emotions to simple positive/negative categories. This loses a lot of nuance. A review might be mixed—positive about some things, negative about others. My models cannot capture that.

06. Conclusion and Recommendations

6.1 Summary of What I Did

In this assignment, I built two sentiment analysis models using the IMDB movie reviews dataset. The first model used a traditional approach: TF-IDF to convert text to numerical representations and logistic regression to classify. The second model used a modern transformer approach—DistilBERT fine-tuned on the reviews.

I cleaned the data differently for each model. The traditional model needed more cleaning, including removing stopwords and stemming. The transformer model only needed basic cleaning.

I evaluated both models using accuracy, precision, recall, F1-score, and confusion matrices. I also measured training time and tried to understand how each model makes decisions using SHAP for the traditional model and attention visualization for the transformer.

6.2 Main Findings

My experiments showed three main things:

First, the transformer model performs better. DistilBERT achieved 93% accuracy compared to 89% for the traditional model. This 4% improvement is significant and shows the power of modern NLP techniques.

Second, the traditional model is still valuable. 89% accuracy is very good, and the traditional model is much faster to train (5 minutes vs 45 minutes). It is also much easier to understand and explain.

Third, there is no single best answer. The right model depends on your specific needs. If you need the highest accuracy, use a transformer. If you need speed and explainability, use a traditional model.

6.3 Recommendations for Different Situations

For students learning NLP: Start with traditional models. They are easier to understand, faster to experiment with, and better at teaching the fundamentals. Once you understand them, move to transformers.

For small businesses with limited resources: Try the traditional approach first. It might be good enough. If not, consider using a cloud API that provides sentiment analysis rather than training your own transformer.

For applications that require explanations, use the traditional model or combine a transformer with explanation tools like SHAP (though SHAP is slower on transformers). Be prepared to invest time in interpretability.

For research or cutting-edge applications: Go with transformers. Fine-tune them on your specific data. Consider larger models like RoBERTa if you have the resources.

6.4 Future Work I Would Like to Do

If I had more time, I would like to do several things.

First, I would try other traditional models, such as SVM or Naive Bayes, to see if they perform better than logistic regression.

Second, I would try larger transformer models, such as BERT-base or RoBERTa, to see if the extra size improves accuracy.

Third, I would try ensemble methods that combine both approaches. Some research suggests that combining different types of models can achieve better results than any single model.

Fourth, I would test my models on other datasets to see how well they generalize. Would they work on Amazon product reviews? Twitter posts? News articles?

Fifth, I would explore more advanced interpretability methods for transformers, maybe using SHAP with a smaller subset of the data.

6.5 Final Thoughts

This project taught me that sentiment analysis is both easier and harder than I expected. The basic ideas are simple: clean text, convert to numbers, and train a classifier. But the details matter a lot. How you clean the text, what hyperparameters you choose, and how you evaluate your models—all of these decisions affect the final result.

I also learned that there is no single "best" approach. The transformer model performed better, but the traditional model had real advantages in speed and interpretability. In the real world, you rarely get something for nothing. Better performance usually comes with higher costs—more computing power, longer training times, and less transparency.

The field of NLP is moving fast, with new models and techniques appearing regularly. But the fundamentals remain important. Good data preparation, careful evaluation, and thoughtful interpretation matter regardless of what model you use. I feel that completing this assignment has given me a solid foundation in both traditional and modern approaches to sentiment analysis, and I am excited to apply these skills to real-world problems.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: pretraining of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
- [3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter,” *arXiv preprint arXiv:1910.01108*, 2019.
- [4] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.
- [5] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, OR, USA, 2011, pp. 142–150.
- [6] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, “HuggingFace’s transformers: State-of-the-art natural language processing,” *arXiv preprint arXiv:1910.03771*, 2019.
- [7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in *Advances in Neural Information Processing Systems*, vol. 32, pp. 5753–5763, 2019.
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [10] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent individualized feature attribution for tree ensembles,” *arXiv preprint arXiv:1802.03888*, 2018.