NLP model

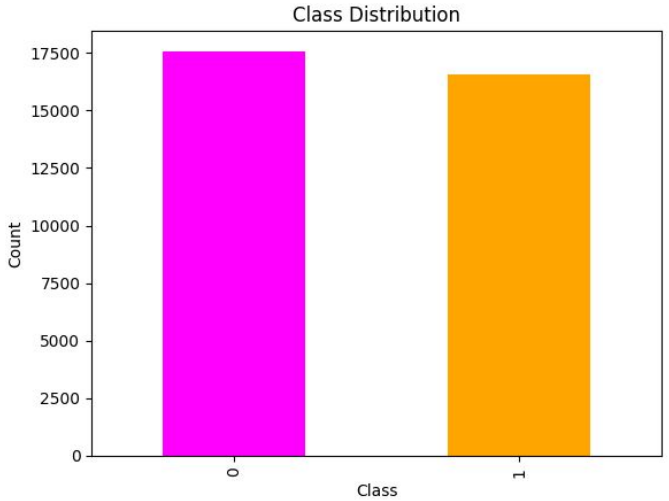
Freddy Roldán Saiqa Mehdi Marina Castillo

Executive summary

- Final result: **91,70%** accuracy
- Model used:
  - **Naive Bayes** + **CountVectorizer**
- Tried:
  - Naive Bayes + TF-IDF / + CountVectorizer
  - Random Forest + TF-IDF / + CountVectorizer

Methods (preprocessing)

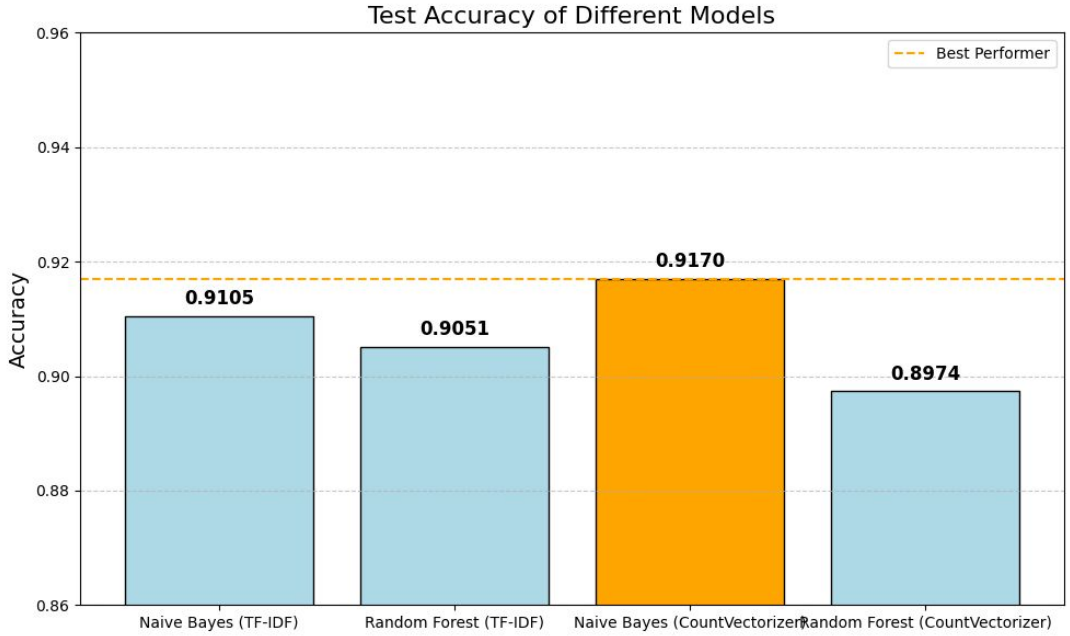
- Data exploration
- Lemmatization
- Special Character Removal
- StopWords Removal
- Tokenization
- WordNet



Methods

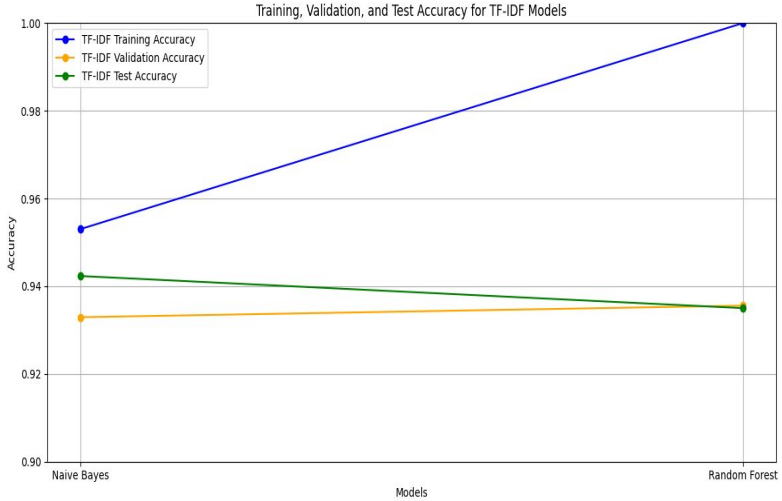
- Naive Bayes and Decision Trees
- TF-IDF and CountVectorizer (Faster and Accurate)
- Sentiment Analysis (TextBlobs)

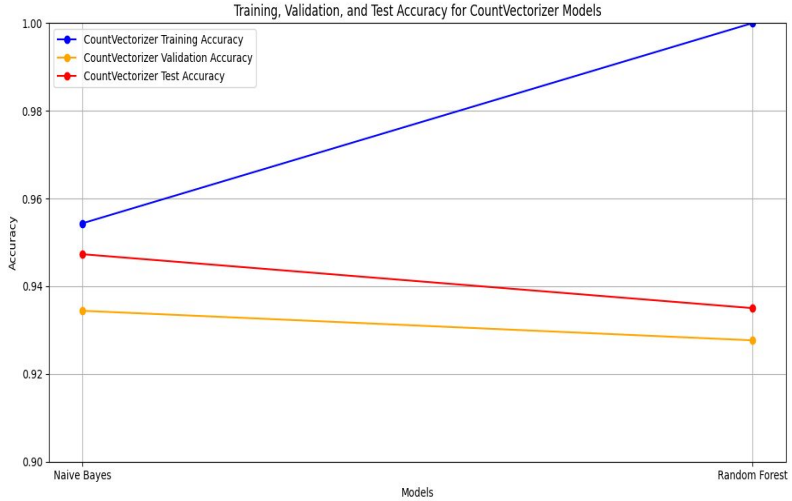
Methods


Test Accuracy of Different Models
Best Performer
Accuracy
0.96
0.94
0.92
0.90
0.88
0.86
0.9105
0.9051
0.9170
0.8974
Naive Bayes (TF-IDF)
Random Forest (TF-IDF)
Naive Bayes (CountVectorizer)
Random Forest (CountVectorizer)

Training, Testing and Validation accuracy

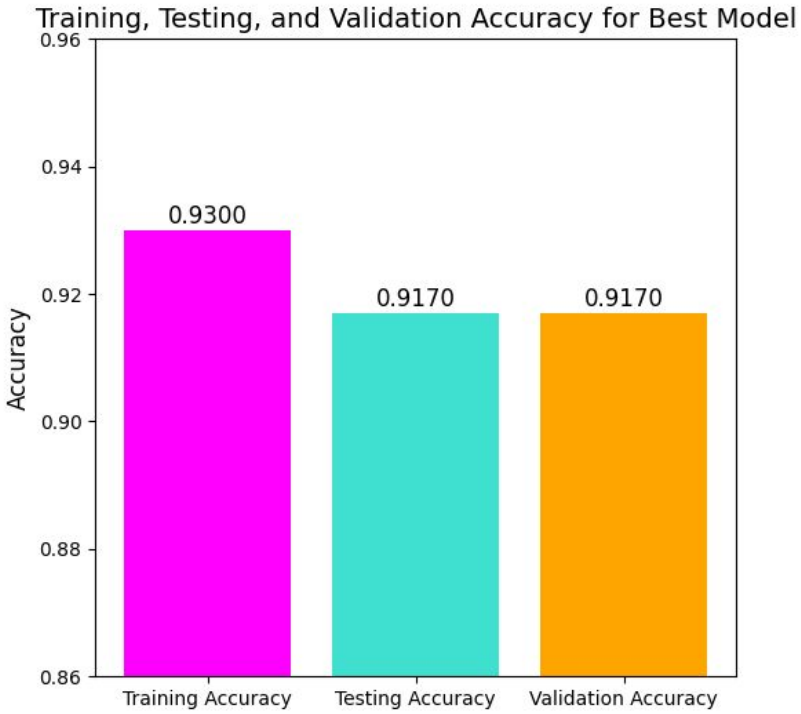
10% for model tuning
10% for evaluation





Results of Naive Bayes + CountVectorizer

- Cross-Validation Accuracy: **0.9170**
- Cross-Validation F1 Score: **0.9166**
- Insights:
  - Effective Text **Classification**
  - Strong **Generalization** Capability
  - **Consistent Performance** Across Cross-Validation Folds
- Comparison: **Slightly lower** than training accuracy (0.9300)
  - Good **generalization**
  - Slightly **overfitting**



Takeaways

- Recap / conclusions
- Challenges
  - Compatibility
  - Negative Values
- Key learnings
  - Time management
- Steps to improve project:
  - Hyperparameter Tuning
  - Use Pre Trained Embeddings
  - More Complex Models

Thank you.

Questions?