Optimizing Candidate Selection

Semester Long Project with Machine Learning and Data Science Club
@Baruch College

Agenda

- Team Introduction
- Project Overview and Background
- Introduction to the Data
- Data Cleaning and Preprocessing
- Modeling and Evaluation
- Conclusion
- Next Steps

Team Introduction

MLDS Team

Natia Burjanadze

Vice President of Tech & Project Manager

Baruch Team

Efe Albayrak
Student

Willy
Student

Kevin De La Cruz
Student

Jae Choi
Student

Lacy Lin
Student

Project Overview and Background

Project Purpose

> The challenge is to build an open source candidate selection AI model to help Baruch organizations automatize and improve their hiring process.

Challenges:

- Candidate selection processes can be time-consuming and resource-intensive for Baruch organizations.
- Biases, both conscious and unconscious, can influence hiring decisions, leading to a lack of diversity and inclusion in the workforce.
- Traditional resume screening methods may overlook qualified candidates who do not fit conventional criteria or who have unconventional career paths.
- Identifying the right candidate from a large pool of applicants can be daunting and prone to human error.

Solution:

- Developing an AI-powered candidate selection model can streamline and optimize the hiring process Implementing machine learning algorithms can help mitigate biases by focusing solely on candidate qualifications and skills.
- Utilizing natural language processing (NLP) techniques can enable the extraction of relevant information from resumes and other candidate documents, facilitating a more comprehensive evaluation.

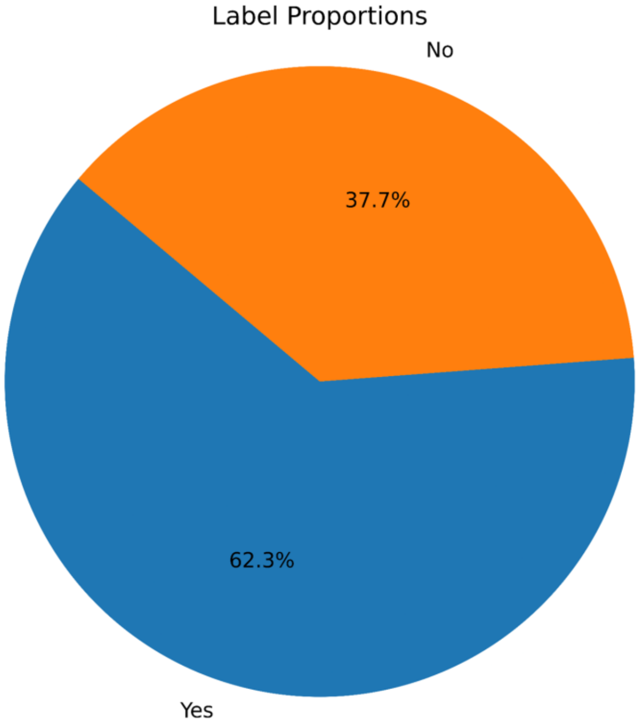
Introduction to the Data

Data

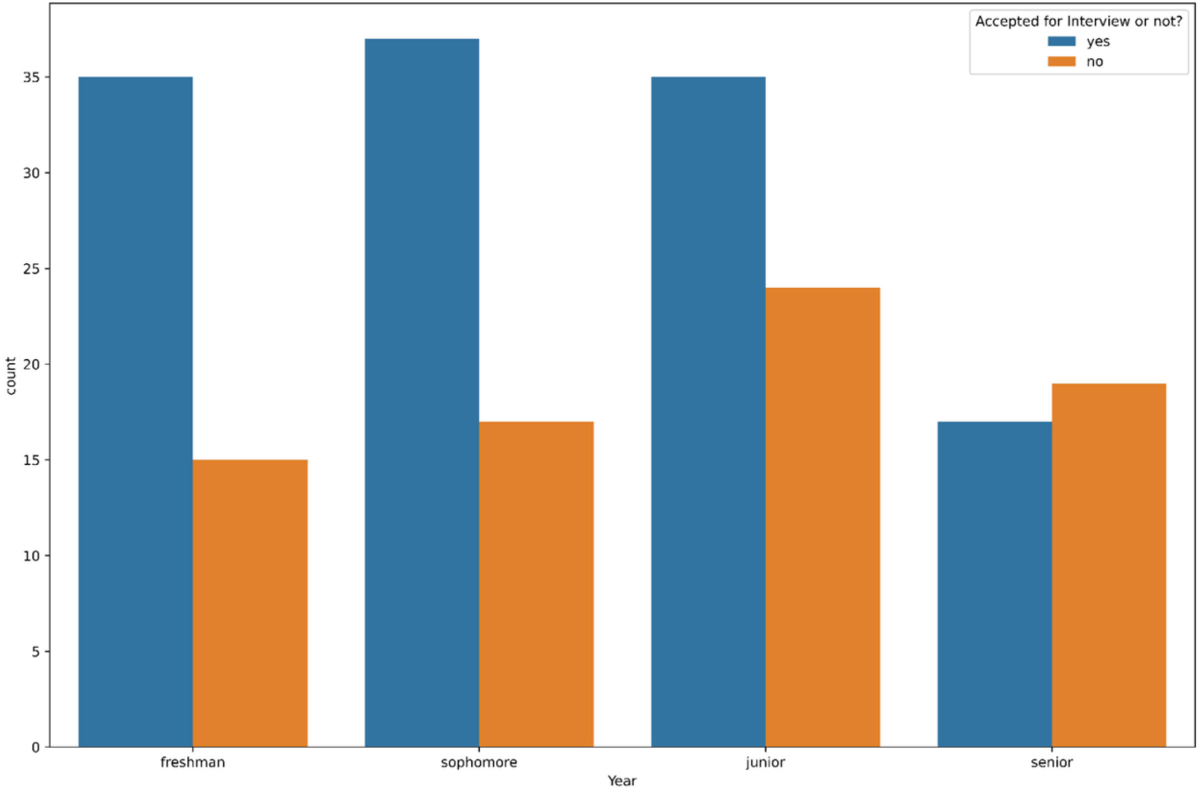
The data set comes from Baruch Organization - CYC club.

	CodeName	Year	Tell us about yourself; we want to know about your personality not your auto-biography.	Ricky's analysis_1	Ricky & ChatGPT analysis_1	ChatGPT only (what Ricky didn't count)_1	What do you think is your greatest strength and greatest weakness?	Ricky's analysis_2	Ricky & ChatGPT analysis_2	ChatGPT only (what Ricky didn't count)_2	Accepted for Interview or not?	Notes
0	S22.01	Sophomore	I am ambitious and driven when it comes to any...	outgoing, sociable, Why CYC, challenger	NaN	ambitious, team-oriented, and adaptable, stron...	I believe my greatest strength is my communica...	Resolution	Strength: Communication\nWeakness: Perfectioni...	NaN	Yes	NaN
1	S22.02	Freshman	I received a lot of help and mentorship from c...	Why CYC, caring, self awareness, teamplayer	professional developmented	sense of community, diverse experiences in con...	Greatest Strength: \nRecognize, admit my weakn...	High Self Awareness, resolution	Bad time management (but improving).	Strengths:\n\nRecognizing and admitting weakne...	Yes	NaN
2	S22.03	Junior	I am an ambivert that enjoys communicating and...	Lacking	NaN	Ambivert, Helping Orientation, Appreciation of...	My greatest strength is Problem Solving and my...	no resolution, lacking	NaN	strong analytical skills, not confident in com...	No	NaN
3	S22.04	Junior	I am more reserved in the beginning of a new s...	introvert	adaptable attitude	nuanced interpersonal style, openness to feedback	Greatest strength is the ability to interact w...	Lacks strong people management skill	work independently	Interact with others	No	NaN
4	S22.05	Sophomore	One of my favorite hobbies outside of school i...	passionate, entrepreneur mindset, goal oriente...	diverse range of skills	diverse range of interests, proactive approach	My greatest strength is being able to work wit...	NaN	collaboration, organization, Perfectionism (Im...	NaN	Yes	NaN

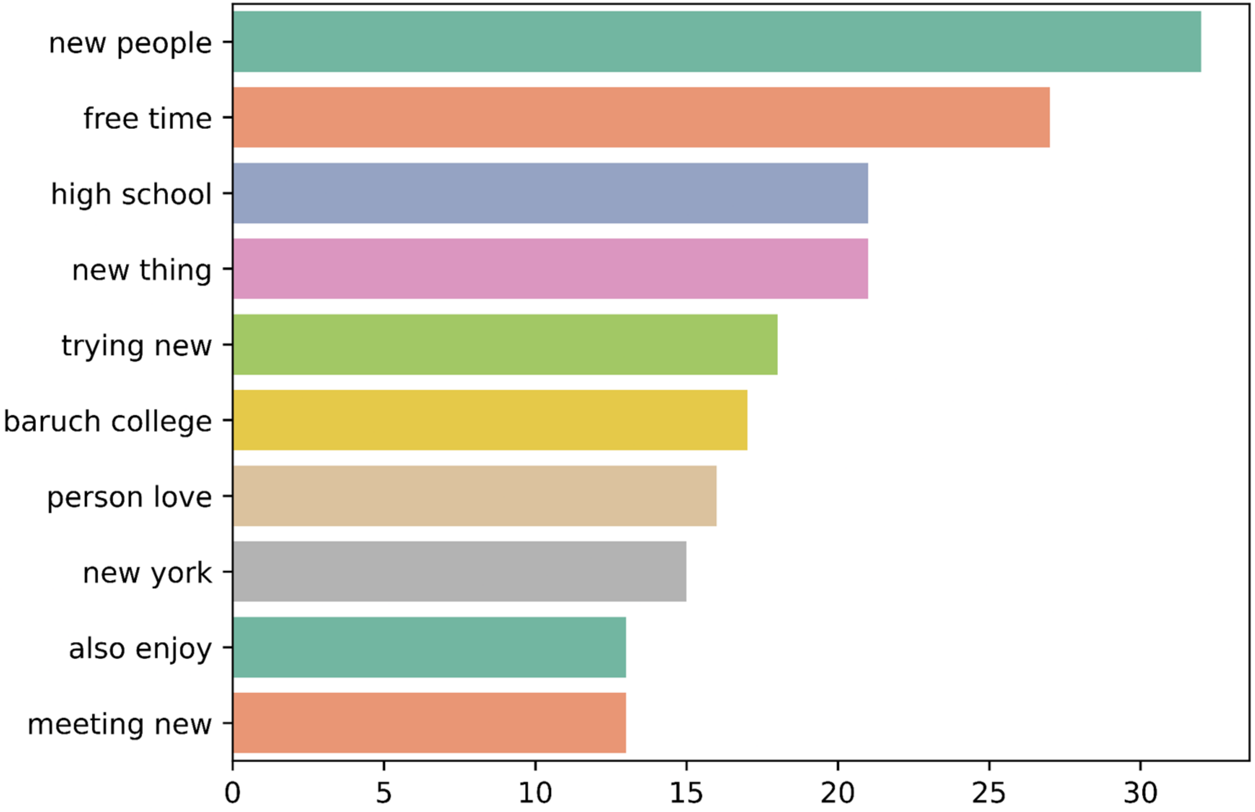
Accepted vs Declined the interview in Data



Student’s Year Vs Acceptance



Top Bigrams in the Data



Word Cloud of Accepted to the interview candidates

Word Cloud of Not Accepted to the interview candidates

Data Cleaning and Preprocessing

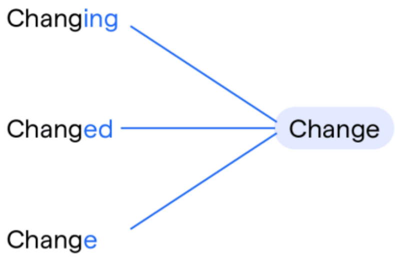
Preprocessing Textual Data

- Formatting
    - Capitalization, removed urls & emojis
- Tokenization
    - Broke text into individual words or tokens to facilitate further analysis at the word level
- Punctuation Removal
    - Excluded punctuation marks from the text

```
def preprocess_text(text):
    # Check if the text is not NaN (float type)
    if isinstance(text, str):
        # Use preprocessor to clean the text
        cleaned_text = p.clean(text)

        # Convert text to lowercase
        cleaned_text = cleaned_text.lower()

        # Tokenize the text
        tokens = word_tokenize(cleaned_text)

        # Remove punctuation
        tokens = [token for token in tokens if token not in string.punctuation]
        tokens = [token for token in tokens if token not in ['``', "''"]]
```

Preprocessing Textual Data

- Stopword Removal
  - Removed common words: “the,” “is,” “in,” “for,” “where,” “when,” “to,” “at,” etc.
- Lemmatization
  - Transformed words into their base or root form to ensure consistency in word




```
    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    stop_words.discard('no')
    tokens = [token for token in tokens if token not in stop_words]

    # Lemmatize words
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # Join tokens back into a cleaned text
    cleaned_text = ' '.join(tokens)

    return cleaned_text
else:
    # Return an empty string or handle missing values as needed
    return ' '
```

Preprocessing Textual Data

- Textual Data before preprocessing:

	Year	Tell us about yourself	Analysis 1	Greatest strength and weakness	Analysis 2	Accepted for Interview
0	Sophomore	I am ambitious and driven when it comes to any...	outgoing, sociable, Why CYC, challenger	I believe my greatest strength is my communica...	Resolution	Yes
1	Freshman	I received a lot of help and mentorship from c...	Why CYC, caring, self awareness, teamplayer	Greatest Strength: \nRecognize, admit my weakn...	High Self Awareness, resolution	Yes

- Textual Data after preprocessing:

0	Sophomore	ambitious driven come sort work assignment tas...	outgoing sociable cyc challenger	believe greatest strength communication always...	resolution	Yes
1	Freshman	received lot help mentorship community-based o...	cyc caring self awareness teamplayer	greatest strength recognize admit weakness imp...	high self awareness resolution	Yes
2	Junior	ambivert enjoys communicating helping others a...	lacking	greatest strength problem solving greatest wea...	resolution lacking	No
3	Junior	reserved beginning new setting become comforta...	introvert	greatest strength ability interact others over...	lack strong people management skill	No
4	Sophomore	one favorite hobby outside school makeup enjoy...	passionate entrepreneur mindset goal oriented ...	greatest strength able work others group impor...		Yes

Feature Engineering

Additional Features added to the Data Frame

- **One-Hot Encoding**
  Converted categorical variables into binary vectors.
  Utilized pd.get_dummies() in pandas.
  Features: Year_Freshman, Year_Senior, etc.

- **Sentiment Analysis**
  Assigned sentiment scores to text data.
  Utilized sentiment analysis libraries like NLTK
  Feature: Sentiment

- **Word Count**
  Calculated the number of words in text data
  Split text into tokens and counted the tokens.
  Feature: Word_count

Year_Freshman	Year_Junior	Year_Senior	Year_Sophomore	combined_text	word_count	sentiment
False	False	False	True	ambitious driven come sort work assignment tas...	130	0.243667
True	False	False	False	received lot help mentorship community-based o...	216	0.181693
False	True	False	False	ambivert enjoys communicating helping others a...	21	0.480000
False	True	False	False	reserved beginning new setting become comforta...	40	0.301136
False	False	False	True	one favorite hobby outside school makeup enjoy...	136	0.261932

Features that weren't added

- **Total Grammar Mistakes**
  Utilized the language tool python library to calculate each applicant's total grammar mistakes

- **Total Grammar Mistakes/Word Count**
  Total grammar mistakes were divided by word
count
  to account for the linear relationship between word count and total mistakes.

```python
def check_grammar(text):
    tool = language_tool_python.LanguageToolPublicAPI('en-US')

    # Check the text for grammar mistakes
    matches = tool.check(text)

    # Return the total number of mistakes
    return len(matches)
```

	Total Mistakes	Mistakes/Word Count
0	6	0.022901
1	3	0.007463
2	1	0.027778
3	1	0.012195
4	10	0.034247
5	2	0.039216
6	10	0.088496
7	1	0.017544
8	5	0.023256

Vectorization

- We used term frequency and inverse document frequency (TF-IDF) to transform the raw text data into vectors which can be processed by a machine learning algorithm.
- TF-IDF uses term frequency (how important a term is within a document) and inverse document frequency (which reduces the weight of a term if it is common between documents)
- Our vectorizer had a vocabulary of 3459 words

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Source: Medium

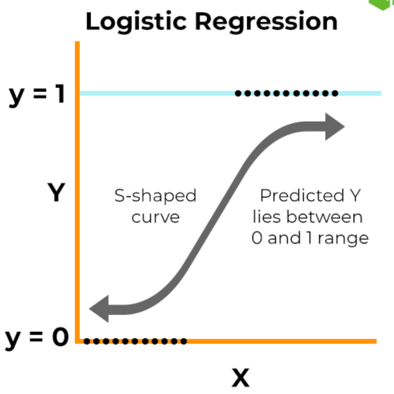
Modeling & Evaluation

Logistic Regression with textual features

- This model estimates the probability that each piece of text (each row in our dataframe) belongs to one of the classes.
- Why did we choose this model?

  Logistic regression is a common baseline model used for classification problems
- Results:

Best hyperparameters	C=10
Average accuracy score after 5-Fold cross-validation:	**0.725**



Logistic Regression with Numerical Features

Best hyperparameters	C=10
Average accuracy score after 5-Fold cross-validation:	**0.80**

Decision Tree with Textual Features

What is a Decision Tree?

- Creating a classification model based on the input data
- DecisionTreeClassifier = creates the decision tree model
- 0 = False, 1 = True
- Precision: % of correctly predicted instances
- Recall: % of relevant predictions
- F1-score: harmonic mean of the precision and recall
- Textual Feature: Vectorization

```
Training Accuracy: 1.0
Testing Accuracy: 0.675
Classification Report:
              precision    recall  f1-score   support

           0       0.61      0.65      0.63        17
           1       0.73      0.70      0.71        23

    accuracy                           0.68        40
   macro avg       0.67      0.67      0.67        40
weighted avg       0.68      0.68      0.68        40
```

- Precision = (True +) / [(True +) + (False +)]
- Recall = (True +) / [(True +) + (False -)]

Decision Tree with Numerical Features

- Columns: Years, Word Count, Sentiment
- Higher Testing Accuracy
- Higher Precision
- Higher Recall
- Higher f1-score

```
Training Accuracy: 1.0
Testing Accuracy: 0.8
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.76      0.76        17
           1       0.83      0.83      0.83        23

    accuracy                           0.80        40
   macro avg       0.80      0.80      0.80        40
weighted avg       0.80      0.80      0.80        40
```

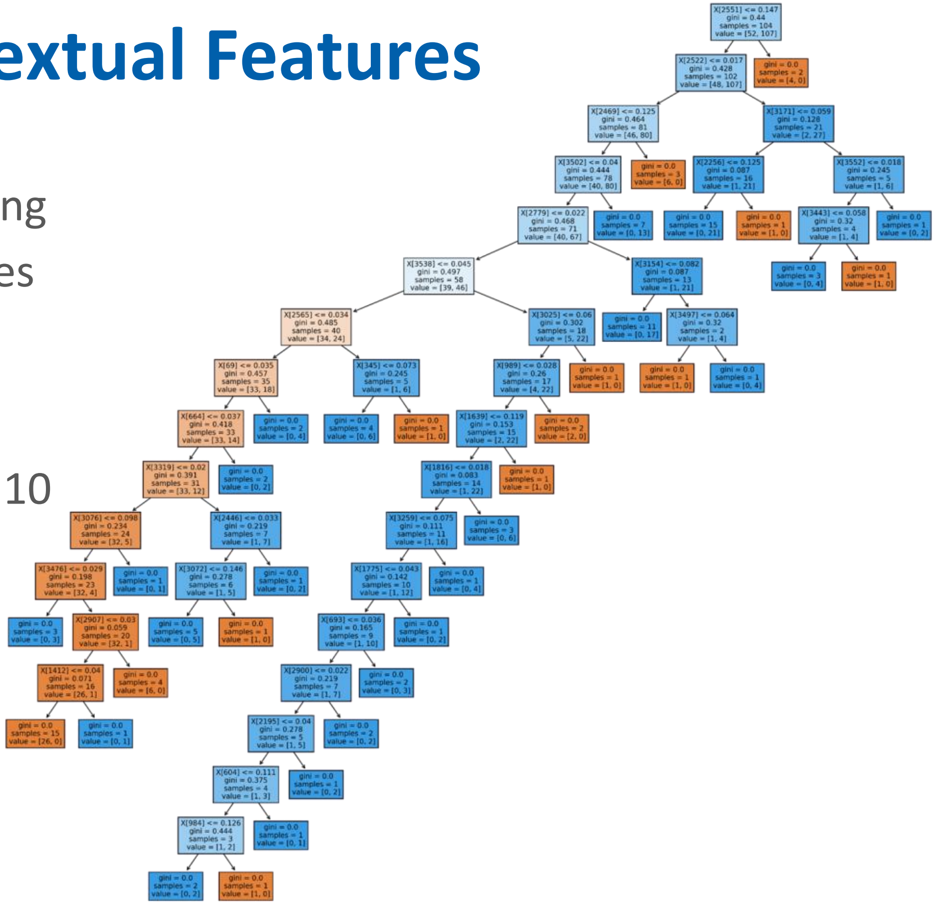
Random Forest with Textual Features

Random Forest is a ensemble learning method of decision trees, which takes the average accuracy of all trees

Parameters: n_estimator =100

Max_depth = 10

Accuracy: 81%



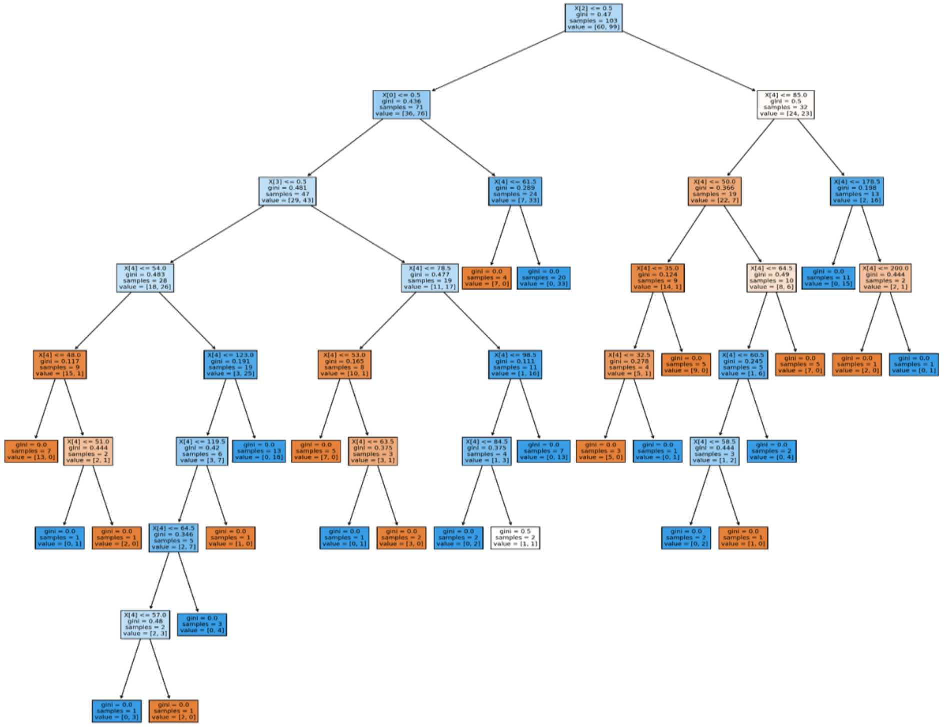
Random Forest with Numerical Features

Parameters: n_estimator =100

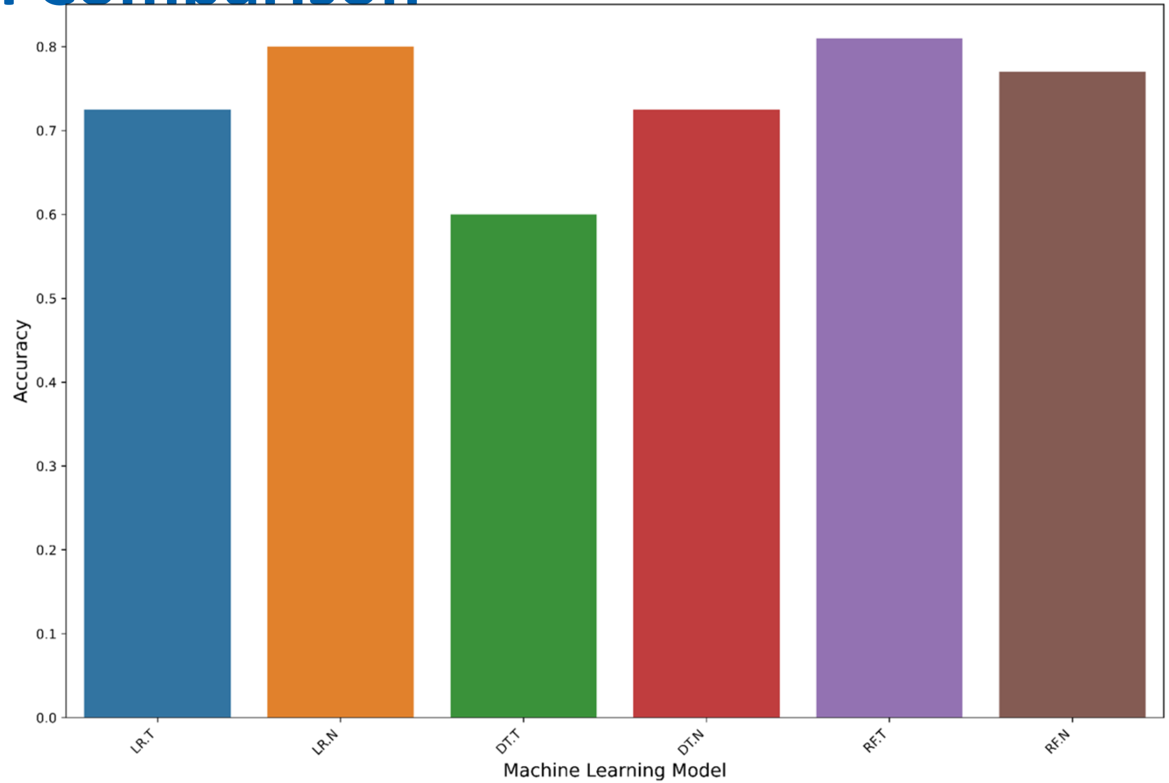
Max_depth = 10

Accuracy: 77%

Insights: Our model using textual data performed slightly better than this model.



Model Comparison



Conclusion and Next Steps

- Train on more data
- Conduct more in-depth data analysis to identify patterns, trends and potential challenges
- Conduct more thorough hyperparameter tuning
- Diversify model types
- Alternative vectorization techniques
- Deployment and real-world application

Questions?