# Project 2: In-Depth Analysis of Human Text vs AI Text

Author: Quan Nguyen

2/27/2024

## Introduction and Problem

- This is the continuation of the previous Project 1, where I did a simple analysis of the Human Text vs AI Text.
- The project will be implemented in Python this time and will include a simple regression model that trained to rate the robustness of the Human Text vs AI Text.

## Loading the Libraries and Data Set

In [ ]:
```
import re
import nltk
import pandas as pd
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
```

## Details of Data Set

### The same case as before, since the initial data set is too large, I will only include the first 100 samples from each type of texts

In [ ]:
```
df = pd.read_csv("AI_Human.csv")

human_df = df[df['generated'] == 0].head(100)
ai_df = df[df['generated'] == 1].head(100)
human_text = human_df['text'].tolist()
ai_text = ai_df['text'].tolist()

print("Number of human texts:", len(human_text))
print("Number of AI texts:", len(ai_text))

nltk.download('punkt', quiet = True)
nltk.download('stopwords', quiet = True)
```

```
all_texts = human_text + ai_text
labels = ['Human'] * len(human_text) + ['AI'] * len(ai_text)
```

```
Number of human texts: 100
Number of AI texts: 100
```

## Pre-Processing of Data Set (Tokenization)

- Utilizing Natural Language Toolkit Library(nltk), I was able to skip a large number of steps involving eliminating stop words, punctuations, etc. that do not contribute to the meaning of the text
- nltk Library uses the same method as in Corpus in R so there was not much of a learning curve involved.

In [ ]:
```
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    tokens = nltk.word_tokenize(text)
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word not in stop_words]
    processed_text = ' '.join(filtered_tokens)
    return processed_text

preprocessed_texts = [preprocess_text(text) for text in all_texts]
print("First preprocessed text: ", preprocessed_texts[0])
```

```
First preprocessed text:  cars cars around since became famous henry ford created bui
lt first modelt cars played major role every day lives since people starting question
limiting car usage would good thing limiting use cars might good thing like matter ar
ticle german suburb life goes without cars elizabeth rosenthal states automobiles lin
chpin suburbs middle class families either shanghai chicago tend make homes experts s
ay huge impediment current efforts reduce greenhouse gas emissions tailpipe passenger
cars responsible percent greenhouse gas emissions europeand percent carintensive area
s united states cars main reason greenhouse gas emissions lot people driving around t
ime getting need go article paris bans driving due smog robert duffer says paris days
nearrecord pollution enforced partial driving ban clear air global city also says mon
day motorist evennumbered license plates ordered leave cars home fined euro fine orde
r would applied oddnumbered plates following day cars reason polluting entire cities 
like paris shows bad cars pollution cause entire city likewise article carfree day sp
inning big hit bogota andrew selsky says programs thats set spread countries millions
columbians hiked biked skated took bus work carfree day leaving streets capital city 
eerily devoid traffic jams third straight year cars banned buses taxis permitted day 
without cars capital city million people like idea carfree days allows lesson polluti
on cars put exhaust people driving time article also tells parks sports centers bustl
ed throughout city uneven pitted sidewalks replaced broad smooth sidewalks rushhour r
estrictions dramatically cut traffic new restaurants upscale shopping districts cropp
ed cars good country columbia aloud repair things needed repairs long time traffic ja
ms gone restaurants shopping districts popped due fact less cars around conclusion us
e less cars carfree days big impact environment cities cutting air pollution cars maj
orly polluted aloud countries like columbia repair sidewalks cut traffic jams limitin
g use cars would good thing america limit use cars maybe riding bike maybe walking so
mewhere isnt far doesnt need use car get limiting use cars might good thing
```

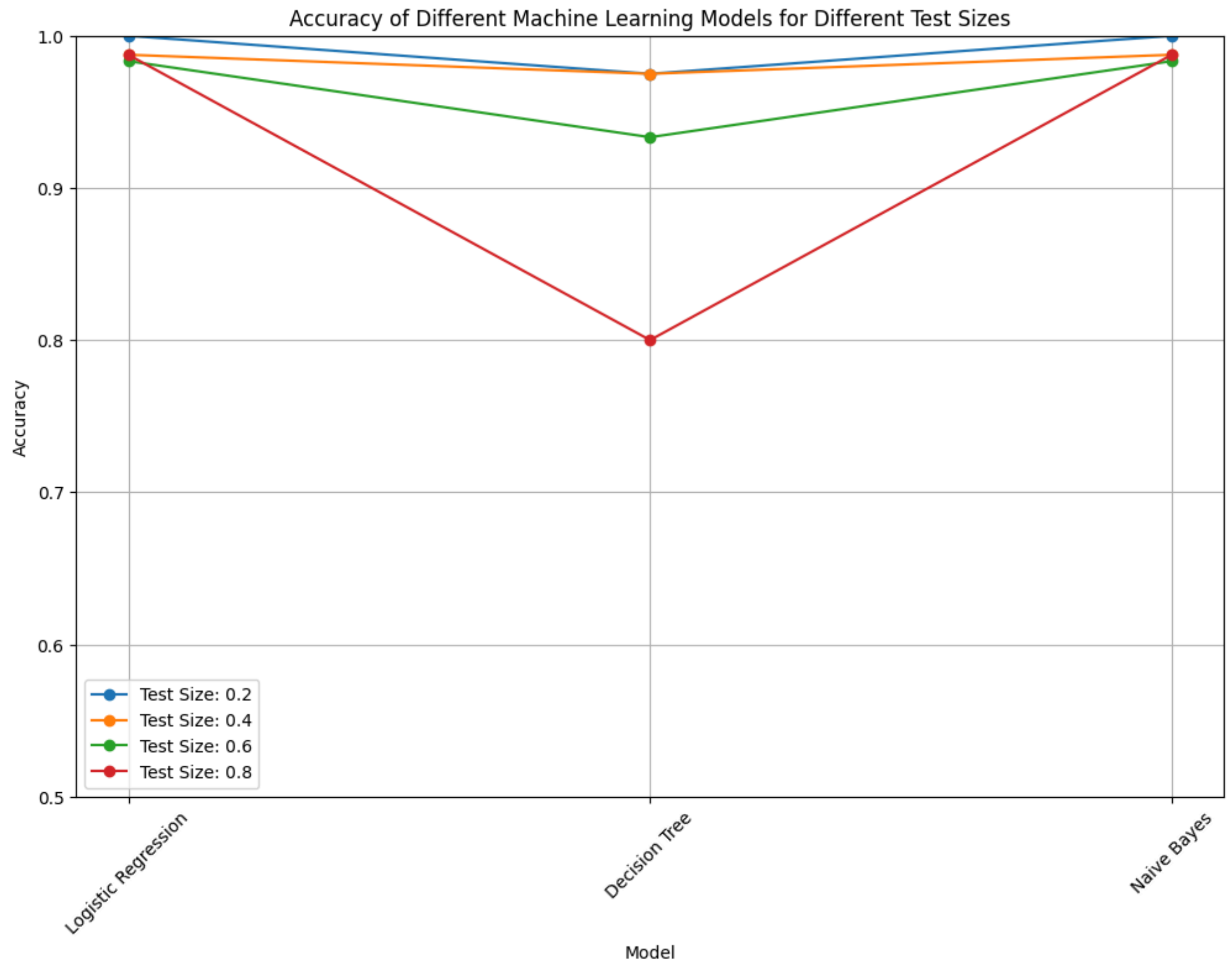
# Feature Extraction Methods and Difference Machine Learning Models

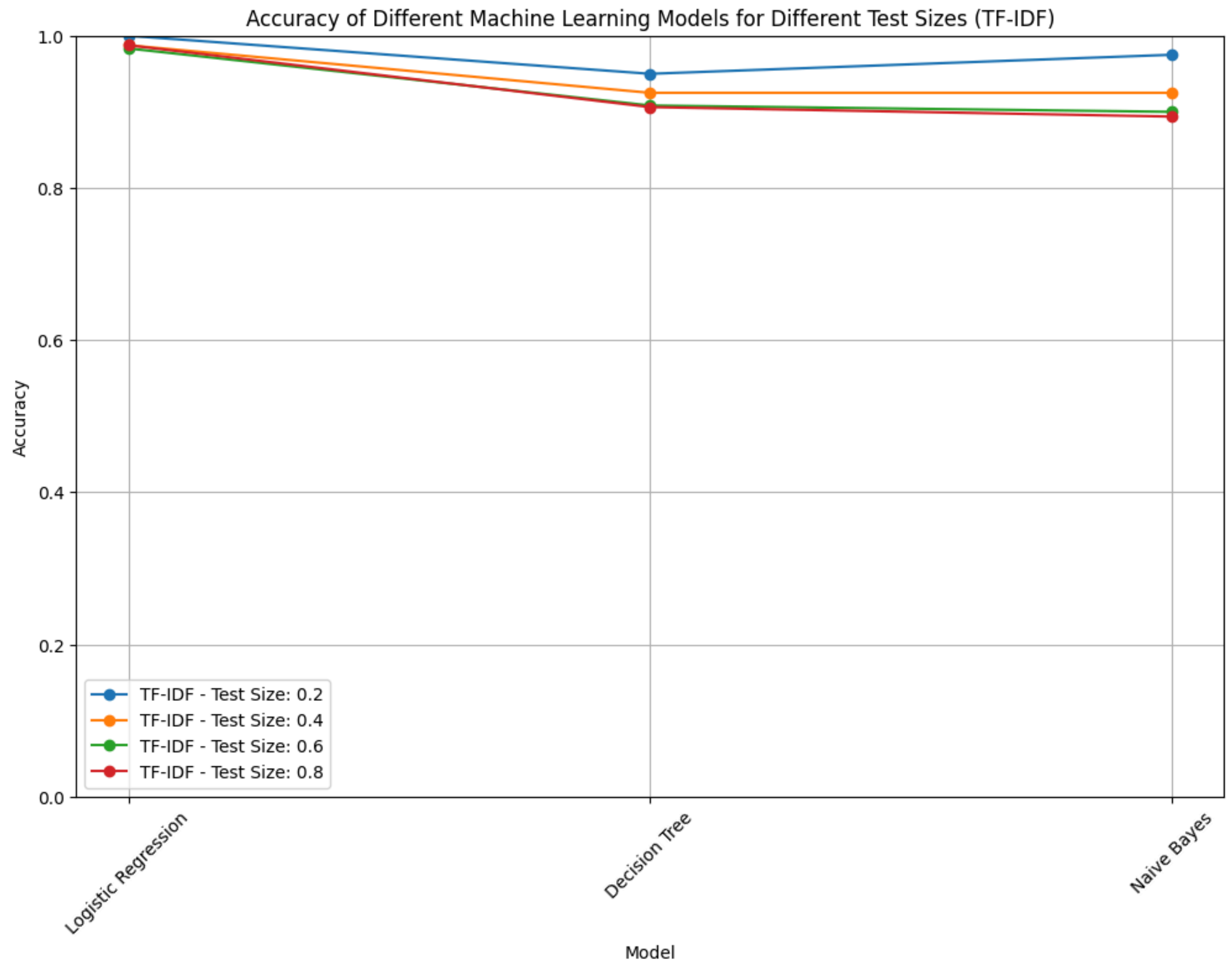
## Next, this is byfar the most important step in the implementation of my project, which includes:

- Testing and comparing the performance of different existing machine learning models: Logistic Regression, Decision Tree, and Naive Bayes to see which one is the best in predicting the generated text.

-Performed training the model using train_test_split from sklearn library, which ask for the user the percent of the samples will be used towards training and the rest will be for testing. -In the same method, I also want to see the importance of splitting the data into training and testing so I set different parameters for the test sizes. -Below is a visualization of the results of training the model where I set the range to be from 0.5 to 1 because upon examning the models, all three of them are over 70% accurate on the testing.

In [ ]:
```
test_sizes = [0.2, 0.4, 0.6, 0.8]
accuracies = {test_size: {} for test_size in test_sizes}

for test_size in test_sizes:
    X_train, X_test, y_train, y_test = train_test_split(preprocessed_texts, labels, te

    bow_vectorizer = CountVectorizer()
    X_train_bow = bow_vectorizer.fit_transform(X_train)
    X_test_bow = bow_vectorizer.transform(X_test)

    models = {
        "Logistic Regression": LogisticRegression(max_iter=1000),
        "Decision Tree": DecisionTreeClassifier(),
        "Naive Bayes": MultinomialNB()
    }

    for name, model in models.items():
        model.fit(X_train_bow, y_train)
        y_pred = model.predict(X_test_bow)
        accuracy = accuracy_score(y_test, y_pred)
        accuracies[test_size][name] = accuracy

plt.figure(figsize=(12, 8))

for test_size, model_accuracies in accuracies.items():
    plt.plot(model_accuracies.keys(), model_accuracies.values(), marker='o', label=f"T

plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.title('Accuracy of Different Machine Learning Models for Different Test Sizes')
plt.ylim(0.5, 1)
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()


test_sizes = [0.2, 0.4, 0.6, 0.8]

accuracies_tfidf = {test_size: {} for test_size in test_sizes}

for test_size in test_sizes:
    X_train, X_test, y_train, y_test = train_test_split(preprocessed_texts, labels, te

    tfidf_vectorizer = TfidfVectorizer()
    X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
    X_test_tfidf = tfidf_vectorizer.transform(X_test)

    models = {
        "Logistic Regression": LogisticRegression(max_iter=1000),
        "Decision Tree": DecisionTreeClassifier(),
        "Naive Bayes": MultinomialNB()
    }

    for name, model in models.items():
        model.fit(X_train_tfidf, y_train)
        y_pred = model.predict(X_test_tfidf)
        accuracy = accuracy_score(y_test, y_pred)
        accuracies_tfidf[test_size][name] = accuracy

plt.figure(figsize=(12, 8))

for test_size, model_accuracies in accuracies_tfidf.items():
    plt.plot(model_accuracies.keys(), model_accuracies.values(), marker='o', label=f"T

plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.title('Accuracy of Different Machine Learning Models for Different Test Sizes (TF-
plt.ylim(0, 1)
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```


Accuracy of Different Machine Learning Models for Different Test Sizes
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Test Size: 0.2
Test Size: 0.4
Test Size: 0.6
Test Size: 0.8
Logistic Regression
Decision Tree
Naive Bayes
Model



## Testing and Evaluating Future Problems

**While the tool is trained with high accuracy, there are still many problems that require deeper training. The following example demonstrates one of the edge cases of the model, where the model, even though it has high AI probability, determined the text is Human Text due to the nature of the probabilistic classification approach. In this example, the model predicts the text has ~71% chance of being AI text but the word "respect" is very common in human generated text so it could not be confident enough to override the decision.**

```
In [ ]: tfidf_vectorizer = TfidfVectorizer()
        X_tfidf = tfidf_vectorizer.fit_transform(all_texts)
        lr_model_tfidf = LogisticRegression(max_iter=1000)
        lr_model_tfidf.fit(X_tfidf, labels)

        def predict_text_type(text):

            preprocessed_text = preprocess_text(text)
            X = tfidf_vectorizer.transform([preprocessed_text])
            prediction = lr_model_tfidf.predict(X)[0]
            probabilities = lr_model_tfidf.predict_proba(X)[0]
            max_coeff_index = X.indices[X.data.argmax()]
            features = tfidf_vectorizer.get_feature_names_out()
            max_coeff_word = features[max_coeff_index]
```

```
    print("Input text:")
    print(text)
    print("Predicted label:", prediction)
    print("Probability (AI):", probabilities[1])
    print("Reason for prediction:")
    print(f"The presence of the word '{max_coeff_word}' contributes most to the predic
    print()
ai_text = "It's essential to approach discussions about any group of people with sensi
predict_text_type(ai_text)
```

```
Input text:
It's essential to approach discussions about any group of people with sensitivity and
recognition of the vast diversity within that group. When speaking about "Asian peopl
e," it's crucial to acknowledge the immense cultural, ethnic, linguistic, and histori
cal variations encompassed by the term. Asia is the largest and most populous contine
nt, comprising numerous countries such as China, India, Japan, South Korea, Vietnam,
and many others, each with its distinct traditions, beliefs, and customs. Asian peopl
e have contributed significantly to various fields like art, science, philosophy, and
technology throughout history. They exhibit a rich tapestry of languages, including M
andarin, Hindi, Bengali, Japanese, Korean, and many more, reflecting centuries of cul
tural evolution. Asian cuisine is renowned worldwide for its diversity, flavors, and
culinary techniques, ranging from spicy curries in South Asia to delicate sushi in Ja
pan. Moreover, Asians have a profound respect for family and community, often placing
great emphasis on filial piety and collective well-being. Traditional values such as
humility, perseverance, and respect for authority continue to influence many Asian so
cieties, alongside rapid modernization and globalization. However, it's essential to
recognize that generalizations about Asian people can be reductive and misleading, as
individuals within this vast demographic encompass a wide spectrum of experiences, be
liefs, and identities. Therefore, fostering understanding and appreciation for the co
mplexities and nuances within Asian cultures is crucial in promoting inclusivity and
combating stereotypes.
Predicted label: Human
Probability (AI): 0.7116881807316123
Reason for prediction:
The presence of the word 'respect' contributes most to the prediction.
```

## Conclusion

- The experiment demonstrates the effectiveness of logistic regression models trained on TF-IDF representations in classifying human-generated and AI-generated text, with consistent accuracy across different test sizes, showcasing its robustness.
- Despite achieving high accuracy rates, the probabilistic nature of the classification reveals the complexity of distinguishing between human and AI-generated text, emphasizing the need for nuanced interpretation and further exploration of feature importance.
- The project underscores the significance of employing natural language processing techniques, such as text preprocessing and machine learning algorithms, to address the evolving challenges of identifying AI-generated content in various contexts, contributing to advancements in content moderation and authenticity verification.

### P/S: Try using my model to see if my conclusion was generated by AI or not!