NewsGroup Classification

NLP Project

By: Team # : T016

AGENDA

01

Objective

02

Team

03

Preprocessing

04

Classification

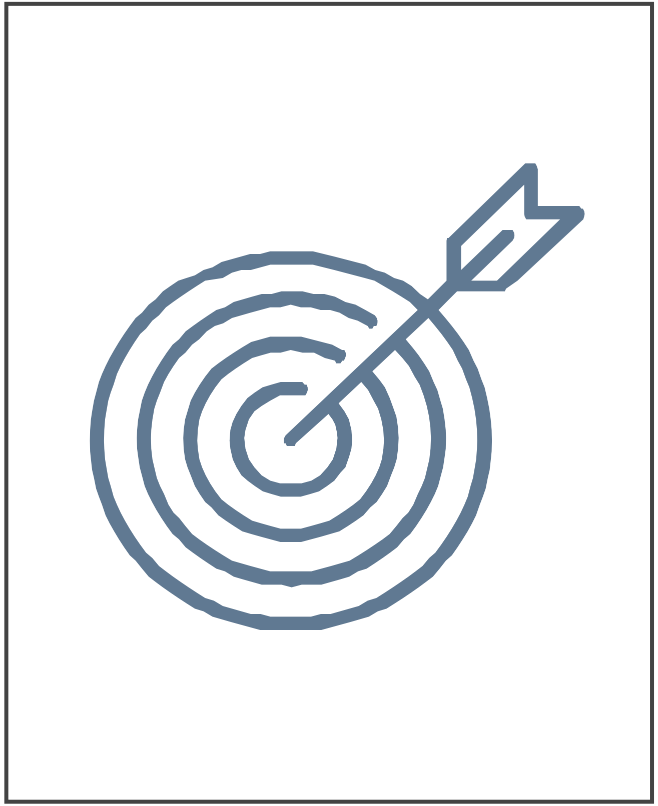
Objective

Build a model to classify news data into various categories through text classification. Feature extraction using TF-IDF

Applying any classifier like Naïve bayes to classify the news to one of 20 groups.

You can apply more than one classifier and select the best one according to its accuracy.

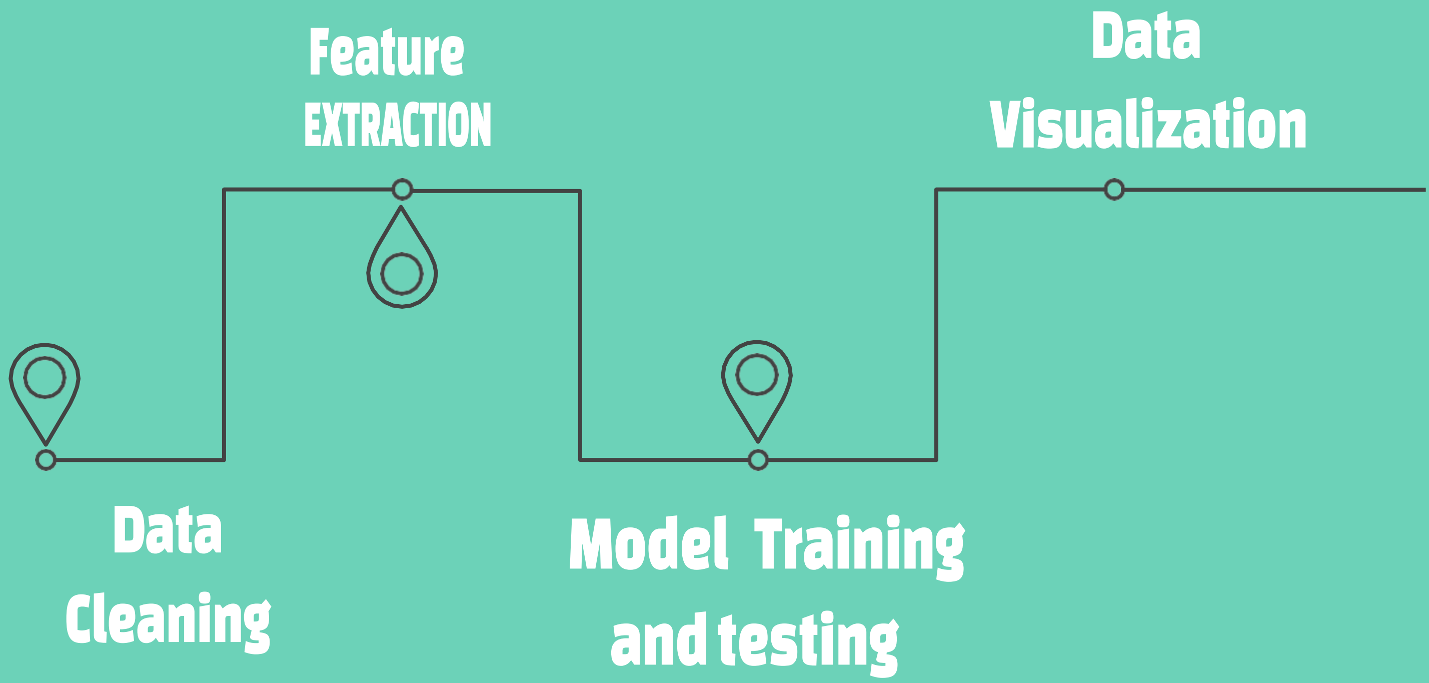
Dataset *is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.*



TEAM

Nour Adel	20201700931
Ahmed Osama	20201700022
Ahmed Rabie	20201700039
Ahmed Awad	20201700065
Ahmed Sayed	20191700036
Amira Alaa	20201700148

Preprocessing



Data Cleaning

Feature EXTRACTION

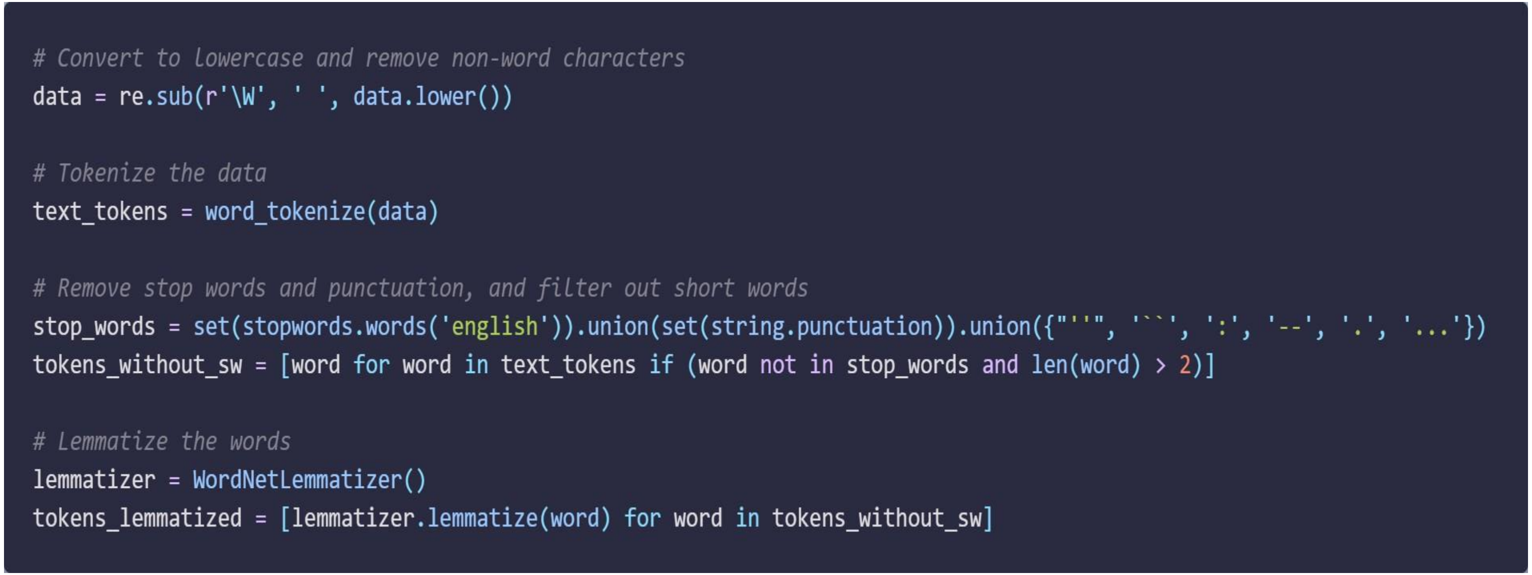
Model Training and testing

Data Visualization

DATA CLEANING

- Convert all letters to lowercase.
- Word Tokenization
- Stop Words Removal
- Lemmatization
- Feature Extraction using TfidfVectorizer or Count Vectorizer

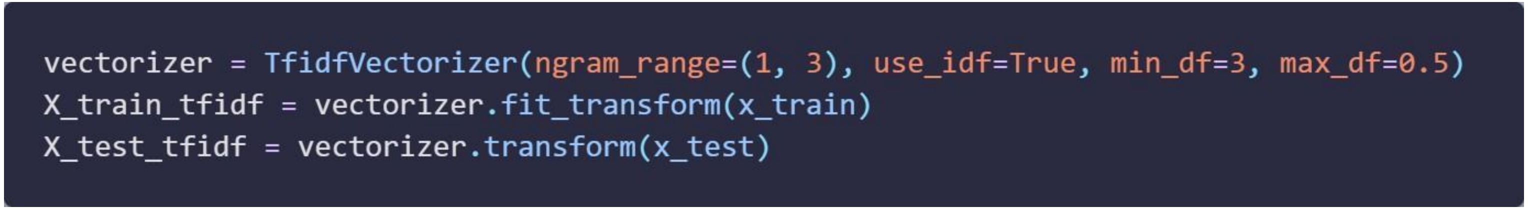
DATA CLEANING



```
# Convert to lowercase and remove non-word characters
data = re.sub(r'\W', ' ', data.lower())

# Tokenize the data
text_tokens = word_tokenize(data)

# Remove stop words and punctuation, and filter out short words
stop_words = set(stopwords.words('english')).union(set(string.punctuation)).union({"''", '``', ':', '--', '.', '...'})
tokens_without_sw = [word for word in text_tokens if (word not in stop_words and len(word) > 2)]

# Lemmatize the words
lemmatizer = WordNetLemmatizer()
tokens_lemmatized = [lemmatizer.lemmatize(word) for word in tokens_without_sw]
```

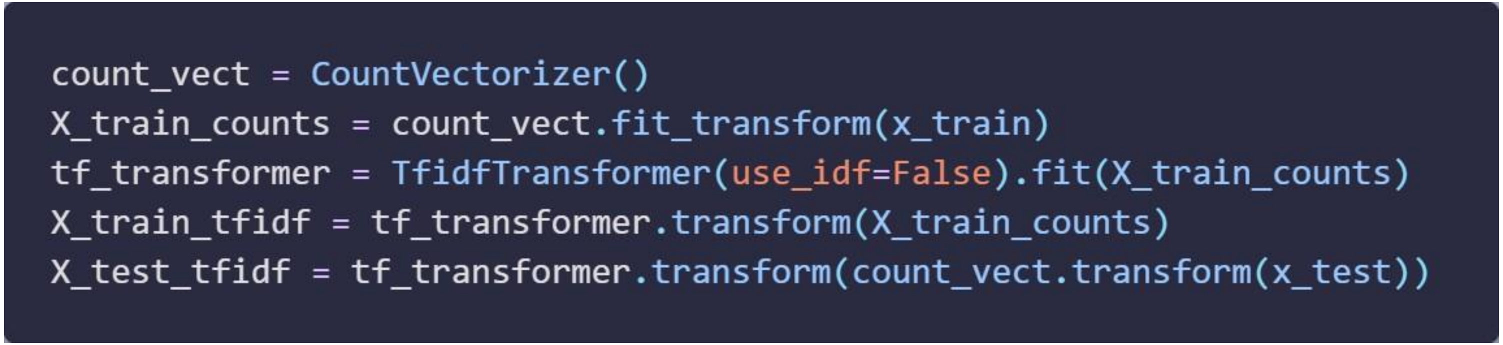
Feature Extraction

Feature Extraction using (TFidfVectorizer)



```
vectorizer = TfidfVectorizer(ngram_range=(1, 3), use_idf=True, min_df=3, max_df=0.5)
X_train_tfidf = vectorizer.fit_transform(x_train)
X_test_tfidf = vectorizer.transform(x_test)
```

Feature Extraction using (Count Vectorizer)



```
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(x_train)
tf_transformer = TfidfTransformer(use_idf=False).fit(X_train_counts)
X_train_tfidf = tf_transformer.transform(X_train_counts)
X_test_tfidf = tf_transformer.transform(count_vect.transform(x_test))
```

Classification

1- Naive Bayes classifier

MultinomialNB Test Score: 0.8940

MultinomialNB Train Score: 0.9497

2- SGD classifier

SGDClassifier Test Score: 0.9437

SGDClassifier Train Score: 0.9729

Classification

3-Random Forest Classifier

RandomForestClassifier Test Score: 0.9295

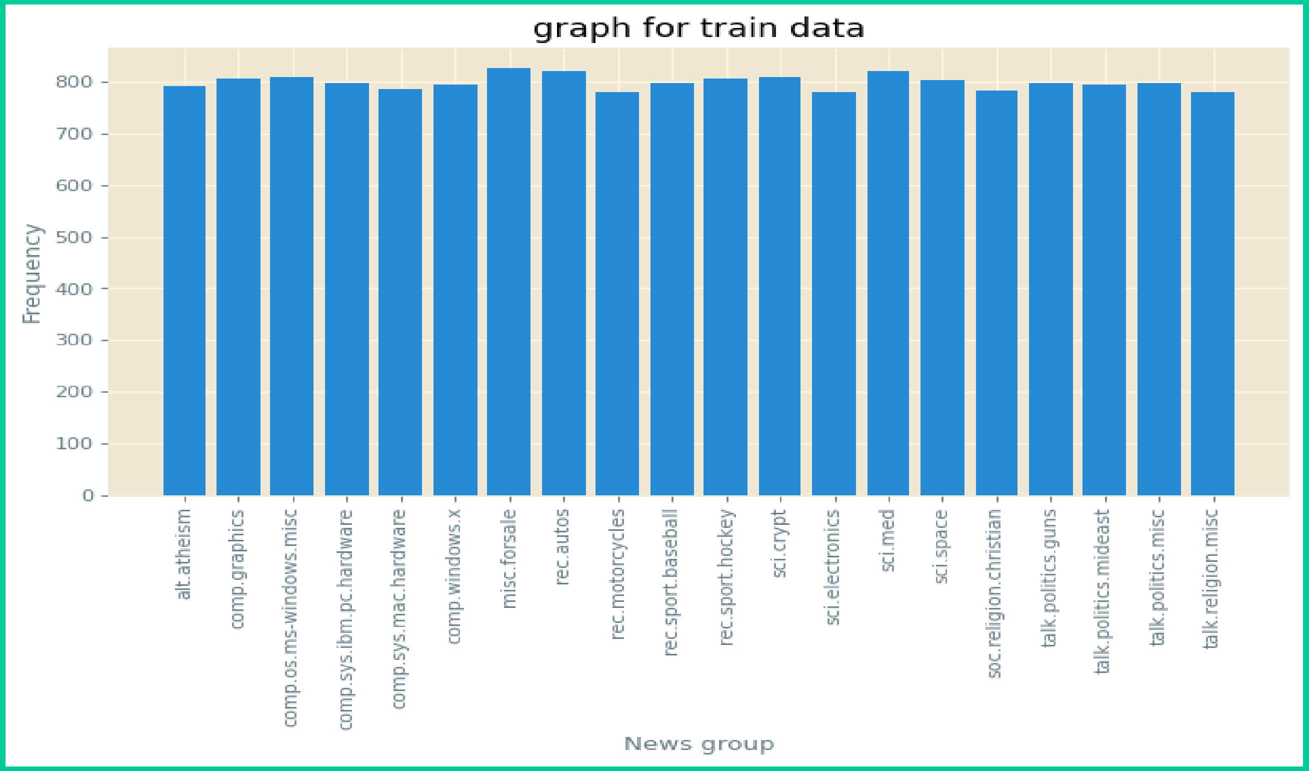
RandomForestClassifier Train Score: 0.9801

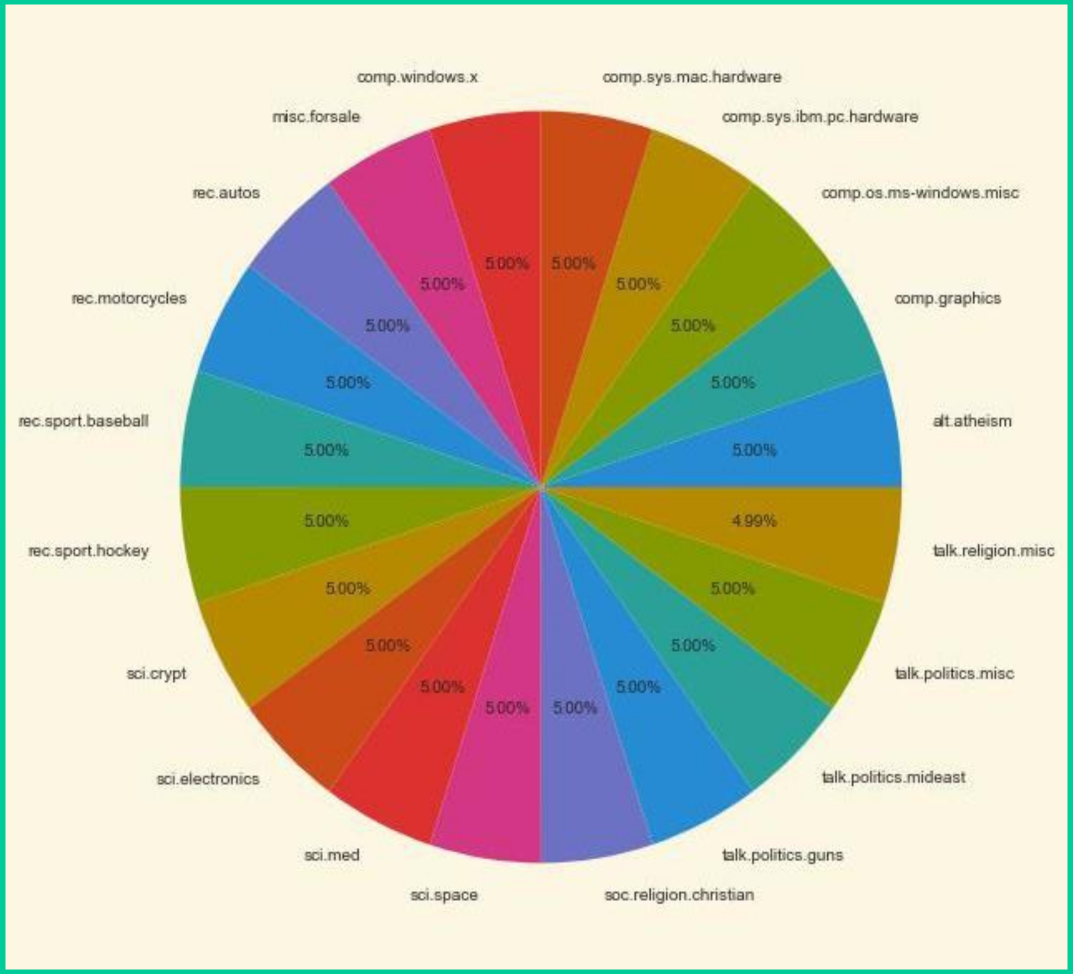
4-SVM Classifier

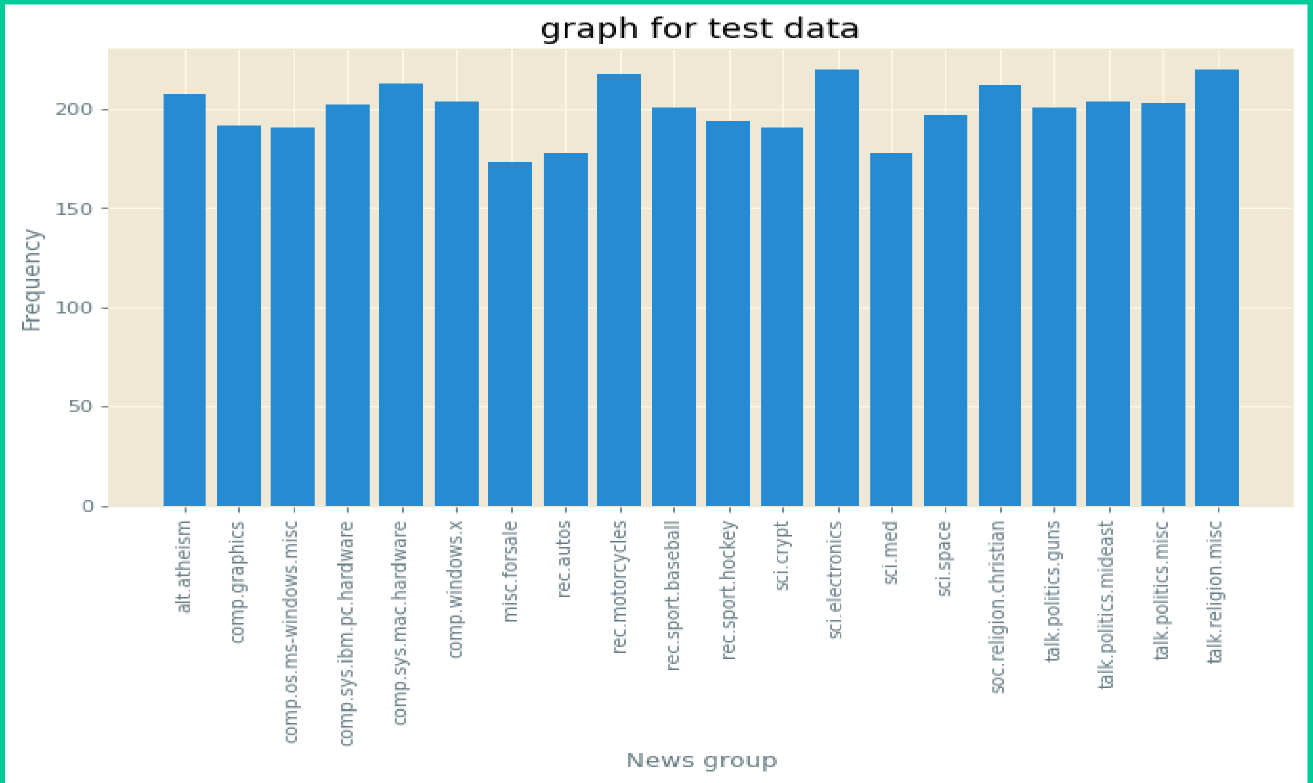
SVM Test Score: 0.9267

SVM Train Score: 0.9801

Visualization







Conclusion

- Random Forest Classifier is the best Classifier.

- Naïve Bias is The Worst Classifier.