

Untitled2

March 3, 2024

IMPORTING LIBRARIES

```
[2]: import warnings
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer

[3]: from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from matplotlib.colors import ListedColormap
from sklearn import metrics
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
```

LOADING DATA

```
[4]: import pandas as pd

# Try different encodings
encodings = ['utf-8', 'latin1', 'iso-8859-1', 'cp1252']
for encoding in encodings:
    try:
        data = pd.read_csv('/content/drive/MyDrive/spam.csv', encoding=encoding)
```

```

    print(f"File read successfully with encoding: {encoding}")
    break
except Exception as e:
    print(f"Error reading file with encoding {encoding}: {e}")

```

Error reading file with encoding utf-8: 'utf-8' codec can't decode bytes in position 135-136: invalid continuation byte
File read successfully with encoding: latin1

```
[5]: data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   v1              5572 non-null   object
 1   v2              5572 non-null   object
 2   Unnamed: 2      50 non-null     object
 3   Unnamed: 3      12 non-null     object
 4   Unnamed: 4       6 non-null     object
dtypes: object(5)
memory usage: 217.8+ KB

```

DROPPING REDUNDENT COLUMN'S

```

[6]: # Dropping the redundant looking columns (for this project)
to_drop = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"]
data = data.drop(data[to_drop], axis=1)
# Renaming the columns because I feel fancy today
data.rename(columns = {"v1": "Target", "v2": "Text"}, inplace = True)
data.head()

```

```

[6]:   Target          Text
0   ham  Go until jurong point, crazy.. Available only ...
1   ham                Ok lar... Joking wif u oni...
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...
3   ham  U dun say so early hor... U c already then say...
4   ham  Nah I don't think he goes to usf, he lives aro...

```

```
[8]: print(data.columns)
```

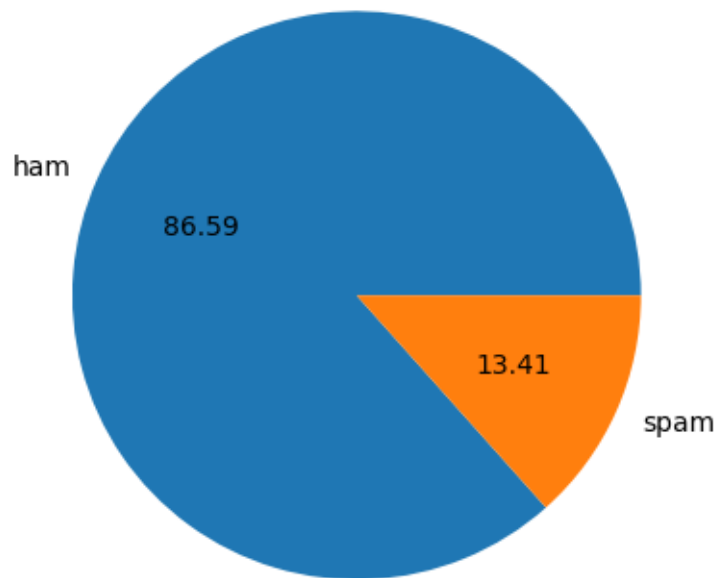
```
Index(['Target', 'Text'], dtype='object')
```

INVESTIGATION OF DATA

```

[9]: import matplotlib.pyplot as plt
plt.pie(data['Target'].value_counts(), labels=['ham', 'spam'], autopct = "%0.2f")
plt.show()

```



FEATURE CRAFTING

```
[10]: import pandas as pd
import re

# Assuming 'data' is your DataFrame with columns 'Target' and 'Text'

# Feature 1: Text Length
data['text_length'] = data['Text'].apply(len) # Number of characters

# Feature 2: Presence of Special Characters or Numbers
data['has_digits'] = data['Text'].str.contains('\d', regex=True)
data['has_special_chars'] = data['Text'].str.contains('[^A-Za-z0-9\s]',
↪regex=True)

# Feature 3: Uppercase Percentage
data['uppercase_percentage'] = data['Text'].apply(lambda x: sum(1 for c in x if
↪c.isupper()) / len(x) * 100 if len(x) > 0 else 0)

# Feature 4: Presence of URLs
data['has_url'] = data['Text'].str.contains('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.
↪&+]|[*\\(\\"))|(?:%[0-9a-fA-F][0-9a-fA-F]))+', regex=True)
```

```
# Display the updated DataFrame
data.head()
```

```
[10]: Target                                Text  text_length \
0    ham  Go until jurong point, crazy.. Available only ...      111
1    ham                                Ok lar... Joking wif u oni...      29
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...     155
3    ham  U dun say so early hor... U c already then say...      49
4    ham  Nah I don't think he goes to usf, he lives aro...      61

      has_digits  has_special_chars  uppercase_percentage  has_url
0         False              True          2.702703      False
1         False              True          6.896552      False
2          True              True          6.451613      False
3         False              True          4.081633      False
4         False              True          3.278689      False
```

```
[11]: import seaborn as sns
```

```
[12]: data.head()
```

```
[12]: Target                                Text  text_length \
0    ham  Go until jurong point, crazy.. Available only ...      111
1    ham                                Ok lar... Joking wif u oni...      29
2  spam  Free entry in 2 a wkly comp to win FA Cup fina...     155
3    ham  U dun say so early hor... U c already then say...      49
4    ham  Nah I don't think he goes to usf, he lives aro...      61

      has_digits  has_special_chars  uppercase_percentage  has_url
0         False              True          2.702703      False
1         False              True          6.896552      False
2          True              True          6.451613      False
3         False              True          4.081633      False
4         False              True          3.278689      False
```

DETECTION OF OUTLIER

```
[14]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Reading the CSV file into a DataFrame with explicit encoding
data = pd.read_csv('/content/drive/MyDrive/spam.csv', encoding='ISO-8859-1')

# Dropping the redundant-looking columns (for this project)
to_drop = ["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"]
data = data.drop(to_drop, axis=1)
```

```

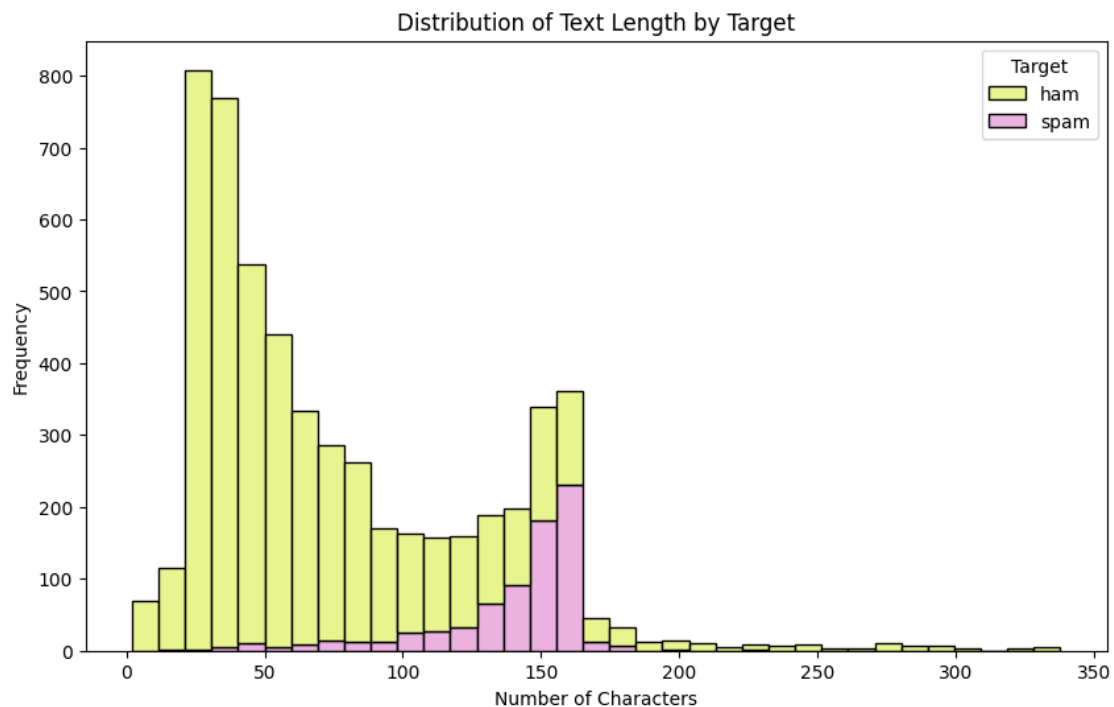
# Renaming the columns
data.rename(columns={"v1": "Target", "v2": "Text"}, inplace=True)

# Calculating the length of the text in the 'Text' column
data['No_of_Characters'] = data['Text'].apply(len)

# Filtering out rows with text length less than 350
data = data[data['No_of_Characters'] < 350]

# Visualizing histogram for 'No_of_Characters' column
plt.figure(figsize=(10, 6))
sns.histplot(data=data, x='No_of_Characters', hue='Target', palette=['#E1F16B', '#E598D8'], multiple='stack')
plt.title('Distribution of Text Length by Target')
plt.xlabel('Number of Characters')
plt.ylabel('Frequency')
plt.show()

```



PREPEROCESING OF DATA

```

[15]: # Printing a sample of the first 5 texts before cleaning
print("\033[1m\u001b[45;1m The First 5 Texts Before Cleaning:\033[0m")
for text in data["Text"][:5]:

```

```
print(text)
```

The First 5 Texts Before Cleaning:

Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
Ok lar... Joking wif u oni...
Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
U dun say so early hor... U c already then say...
Nah I don't think he goes to usf, he lives around here though

```
[16]: import re

# Defining a function to clean up the text
def clean_text(text):
    # Replacing all non-alphabetic characters with a space
    cleaned_text = re.sub('[^a-zA-Z]', ' ', text)
    # Converting to lowercase
    cleaned_text = cleaned_text.lower()
    # Removing extra whitespaces
    cleaned_text = ' '.join(cleaned_text.split())
    return cleaned_text

# Applying the clean_text function to the 'Text' column and creating a new
# 'Clean_Text' column
data["Clean_Text"] = data["Text"].apply(clean_text)

# Displaying the first 5 texts after cleaning
print("\033[1m\u001b[45;1m The First 5 Texts after cleaning:\033[0m")
for text in data["Clean_Text"][:5]:
    print(text)
```

The First 5 Texts after cleaning:

go until jurong point crazy available only in bugis n great world la e buffet
cine there got amore wat
ok lar joking wif u oni
free entry in a wkly comp to win fa cup final tkts st may text fa to to receive
entry question std txt rate t c s apply over s
u dun say so early hor u c already then say
nah i don t think he goes to usf he lives around here though

```
[18]: import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
```

[18]: True

Tokenization:- It is the process of breaking down a text into smaller components, typically words or sentences, called tokens. These tokens are the basic units of text that are used in natural language processing (NLP) tasks such as text analysis, machine translation, and sentiment analysis. The tokenization process involves splitting the text based on certain criteria, such as whitespace or punctuation marks, to create meaningful units for further analysis or processing.

```
[19]: import nltk

# Tokenizing the cleaned text and creating a new column 'Tokenize_Text'
data["Tokenize_Text"] = data["Clean_Text"].apply(nltk.word_tokenize)

# Printing the first 5 tokenized texts
print("\033[1m\u001b[45;1m The First 5 Texts after Tokenizing:\033[0m")
for tokens in data["Tokenize_Text"][:5]:
    print(tokens)
```

The First 5 Texts after Tokenizing:

```
['go', 'until', 'jurong', 'point', 'crazy', 'available', 'only', 'in', 'bugis',
'n', 'great', 'world', 'la', 'e', 'buffet', 'cine', 'there', 'got', 'amore',
'wat']
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'in', 'a', 'wkly', 'comp', 'to', 'win', 'fa', 'cup', 'final',
'tkts', 'st', 'may', 'text', 'fa', 'to', 'to', 'receive', 'entry', 'question',
'std', 'txt', 'rate', 't', 'c', 's', 'apply', 'over', 's']
['u', 'dun', 'say', 'so', 'early', 'hor', 'u', 'c', 'already', 'then', 'say']
['nah', 'i', 'don', 't', 'think', 'he', 'goes', 'to', 'usf', 'he', 'lives',
'around', 'here', 'though']
```

STOPWORDS: Stopwords refer to common words that are often filtered out during natural language processing (NLP) tasks because they occur frequently in a language and typically do not carry significant meaning in the context of a specific analysis. Examples of stopwords in English include words like “the,” “is,” “and,” “of,” etc. These words are often excluded from text analysis to focus on the more meaningful content words, which helps in reducing the dimensionality of the data and improving the efficiency and accuracy of NLP algorithms.

```
[20]: from nltk.corpus import stopwords
import nltk

# Ensure the NLTK stopwords are downloaded
nltk.download('stopwords')

# Defining the function to remove stopwords
def remove_stopwords(text):
    stop_words = set(stopwords.words("english"))
    filtered_text = [word for word in text if word not in stop_words]
    return filtered_text
```

```

# Applying the remove_stopwords function to create a new column
data["Nostopword_Text"] = data["Tokenize_Text"].apply(remove_stopwords)

# Printing the first 5 texts after removing the stopwords
print("\033[1m\u001b[45;1m The First 5 Texts after removing the stopwords:
↪\033[0m")
for text in data["Nostopword_Text"][:5]:
    print(text)

```

[nlTK_data] Downloading package stopwords to /root/nltk_data...

[nlTK_data] Unzipping corpora/stopwords.zip.

The First 5 Texts after removing the stopwords:

```

['go', 'jurong', 'point', 'crazy', 'available', 'bugis', 'n', 'great', 'world',
'la', 'e', 'buffet', 'cine', 'got', 'amore', 'wat']
['ok', 'lar', 'joking', 'wif', 'u', 'oni']
['free', 'entry', 'wkly', 'comp', 'win', 'fa', 'cup', 'final', 'tkts', 'st',
'may', 'text', 'fa', 'receive', 'entry', 'question', 'std', 'txt', 'rate', 'c',
'apply']
['u', 'dun', 'say', 'early', 'hor', 'u', 'c', 'already', 'say']
['nah', 'think', 'goes', 'usf', 'lives', 'around', 'though']

```

lemmatization also converts a word to its root form. However, the difference is that lemmatization ensures that the root word belongs to the language one is dealing with, in our case it is English. If we use lemmatization the output would be in English

```

[21]: from nltk.stem import WordNetLemmatizer
import nltk

# Download the NLTK WordNet data if not already downloaded
nltk.download('wordnet')

# Initializing the WordNet lemmatizer
lemmatizer = WordNetLemmatizer()

# Lemmatization function
def lemmatize_word(text):
    # Lemmatize each word in the text
    lemmas = [lemmatizer.lemmatize(word, pos='v') for word in text]
    return lemmas

# Applying the lemmatization function to create a new column
data["Lemmatized_Text"] = data["Nostopword_Text"].apply(lemmatize_word)

# Printing the first 5 texts after lemmatization
print("\033[1m\u001b[45;1m The First 5 Texts after lemmatization:\033[0m",
↪*data["Lemmatized_Text"][:5], sep="\n")

```


[nltk_data] Downloading package wordnet to /root/nltk_data...

The First 5 Texts after lemmatization:

```
['go', 'jurong', 'point', 'crazy', 'available', 'bugis', 'n', 'great', 'world',  
'la', 'e', 'buffet', 'cine', 'get', 'amore', 'wat']  
['ok', 'lar', 'joke', 'wif', 'u', 'oni']  
['free', 'entry', 'wkly', 'comp', 'win', 'fa', 'cup', 'final', 'tkts', 'st',  
'may', 'text', 'fa', 'receive', 'entry', 'question', 'std', 'txt', 'rate', 'c',  
'apply']  
['u', 'dun', 'say', 'early', 'hor', 'u', 'c', 'already', 'say']  
['nah', 'think', 'go', 'usf', 'live', 'around', 'though']
```

TF-IDF in NLP stands for Term Frequency– Inverse document frequency. In NLP cleaned data needs to be converted into a numerical format where each word is represented by a matrix. This is also known as word embedding or Word vectorization. Term Frequency (TF) = (Frequency of a term in the document)/(Total number of terms in documents) Inverse Document Frequency (IDF) = $\log((\text{total number of documents})/(\text{number of documents with term } t))$, I will be using TfidfVectorizer() to vectorize the preprocessed data

```
[22]: from sklearn.preprocessing import LabelEncoder  
      # Creating a LabelEncoder  
      label_encoder = LabelEncoder()  
      # Encoding the 'Target' variable  
      y = label_encoder.fit_transform(data['Target'])
```

```
[23]: # Creating a corpus of text features to encode further into vectorized form  
      corpus = []  
      for i in data["Lemmatized_Text"]:  
          msg = ' '.join([row for row in i])  
          corpus.append(msg)  
  
      # Printing the first 5 lines in the corpus  
      print("\033[1m\u001b[45;1m The First 5 lines in corpus :\033[0m", *corpus[:5],  
            ↪sep="\n")
```

The First 5 lines in corpus :

```
go jurong point crazy available bugis n great world la e buffet cine get amore  
wat  
ok lar joke wif u oni  
free entry wkly comp win fa cup final tkts st may text fa receive entry question  
std txt rate c apply  
u dun say early hor u c already say  
nah think go usf live around though
```

```
[24]: from sklearn.feature_extraction.text import TfidfVectorizer  
      # Creating TfidfVectorizer  
      tfidf_vectorizer = TfidfVectorizer(max_features=5000) # You can adjust  
            ↪max_features as needed
```

```
# Transforming the corpus into TF-IDF vectors
X = tfidf_vectorizer.fit_transform(corpus).toarray()

# Displaying the type of the feature matrix
print("Type of feature matrix X:", X.dtype)
```

Type of feature matrix X: float64

```
[25]: label_encoder = LabelEncoder()
data["Target"] = label_encoder.fit_transform(data["Target"])
```

```
[26]: # Assuming 'corpus' is the list of cleaned texts
      # Creating TfidfVectorizer
      tfidf_vectorizer = TfidfVectorizer(max_features=5000) # You can
      ↪ adjust_max_features as needed
      # Fitting the vectorizer on the entire dataset
      X_tfidf = tfidf_vectorizer.fit_transform(corpus).toarray()
```

A word cloud is a visual representation of text data, where the size of each word indicates its frequency or importance within the dataset. In a word cloud, words are typically arranged in random order, with more frequent words appearing larger and more prominently, while less frequent words appear smaller. Word clouds are commonly used to quickly visualize and identify the most prominent terms or topics within a body of text, making them useful for tasks such as sentiment analysis, keyword extraction, and content summarization. They are often employed in fields like data analysis, market research, and social media monitoring.

```
[27]: from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
```

```
[28]: spam_wc = wc.generate(data[data['Target']==1]['Clean_Text'].str.cat(sep = " "))
```

```
[29]: plt.figure(figsize=(12,6))
plt.imshow(spam_wc)
```

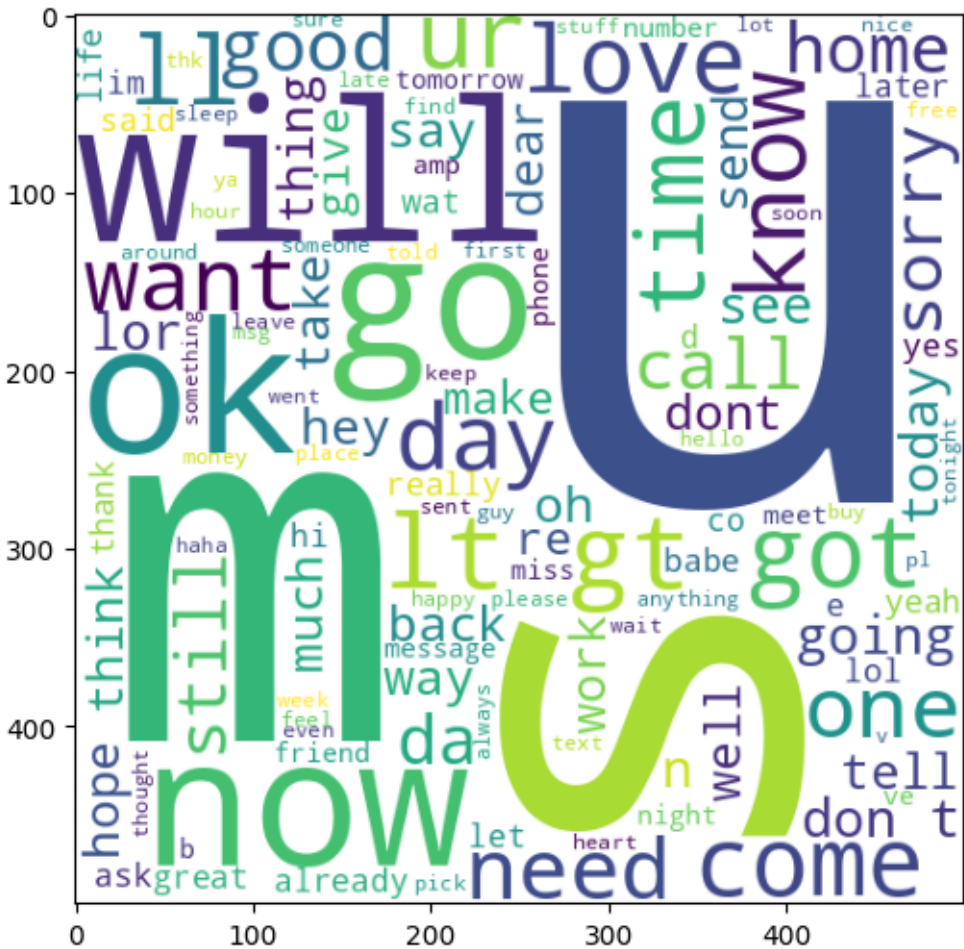
```
[29]: <matplotlib.image.AxesImage at 0x785977851d20>
```



```
[32]: ham_wc = wc.generate(data[data['Target']==0]['Clean_Text'].str.cat(sep = " "))
```

```
[33]: plt.figure(figsize=(12,6))
plt.imshow(ham_wc)
```

```
[33]: <matplotlib.image.AxesImage at 0x785979d3fc40>
```



```
[34]: data.head()
```

```
[34]: Target                                Text \
0      0  Go until jurong point, crazy.. Available only ...
1      0                                  Ok lar... Joking wif u oni...
2      1  Free entry in 2 a wkly comp to win FA Cup fina...
3      0  U dun say so early hor... U c already then say...
4      0  Nah I don't think he goes to usf, he lives aro...

No_of_Characters                                Clean_Text \
0      111  go until jurong point crazy available only in ...
1      29                                  ok lar joking wif u oni
2      155  free entry in a wkly comp to win fa cup final ...
3      49                                  u dun say so early hor u c already then say
4      61  nah i don t think he goes to usf he lives arou...

Tokenize Text \
```

```

0 [go, until, jurong, point, crazy, available, o...
1 [ok, lar, joking, wif, u, oni]
2 [free, entry, in, a, wkly, comp, to, win, fa, ...
3 [u, dun, say, so, early, hor, u, c, already, t...
4 [nah, i, don, t, think, he, goes, to, usf, he,...

                                Nostopword_Text \
0 [go, jurong, point, crazy, available, bugis, n...
1 [ok, lar, joking, wif, u, oni]
2 [free, entry, wkly, comp, win, fa, cup, final,...
3 [u, dun, say, early, hor, u, c, already, say]
4 [nah, think, goes, usf, lives, around, though]

                                Lemmatized_Text
0 [go, jurong, point, crazy, available, bugis, n...
1 [ok, lar, joke, wif, u, oni]
2 [free, entry, wkly, comp, win, fa, cup, final,...
3 [u, dun, say, early, hor, u, c, already, say]
4 [nah, think, go, usf, live, around, though]

```

MODEL BUILDING

```

[35]: from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer

# Encoding the 'Target' variable
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(data['Target'])

# Creating a corpus of lemmatized text
corpus = [' '.join(row) for row in data["Lemmatized_Text"]]

# Transforming text data into numbers
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(corpus).toarray()

# Splitting the testing and training sets with stratify
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42, stratify=y)

# Build pipelines for four different classifiers

```

```

nb_model = Pipeline([('classifier', MultinomialNB())])
rf_model = Pipeline([('classifier', RandomForestClassifier(random_state=42))])
knn_model = Pipeline([('classifier', KNeighborsClassifier())])
svm_model = Pipeline([('classifier', SVC())])

# Fit all the models on the training data
nb_model.fit(X_train, y_train)
rf_model.fit(X_train, y_train)
knn_model.fit(X_train, y_train)
svm_model.fit(X_train, y_train)

# Get cross-validation on the training set for all the models for accuracy
models = [nb_model, rf_model, knn_model, svm_model]
model_names = ['Naïve Bayes', 'RandomForest', 'KNeighbors', 'SVM']
for model, name in zip(models, model_names):
    cross_val_acc = cross_val_score(model, X_train, y_train, cv=5,
    ↪scoring='accuracy')
    print(f'{name} Cross-Validation Accuracy: {cross_val_acc.mean()}')

```

Naïve Bayes Cross-Validation Accuracy: 0.9632715297033222
 RandomForest Cross-Validation Accuracy: 0.9783684755781712
 KNeighbors Cross-Validation Accuracy: 0.9100960317702145
 SVM Cross-Validation Accuracy: 0.9756639954497903

VISUALIZING THE ACCURACIES OF THE MODEL THROUGH CROSS-VALIDATION.

```

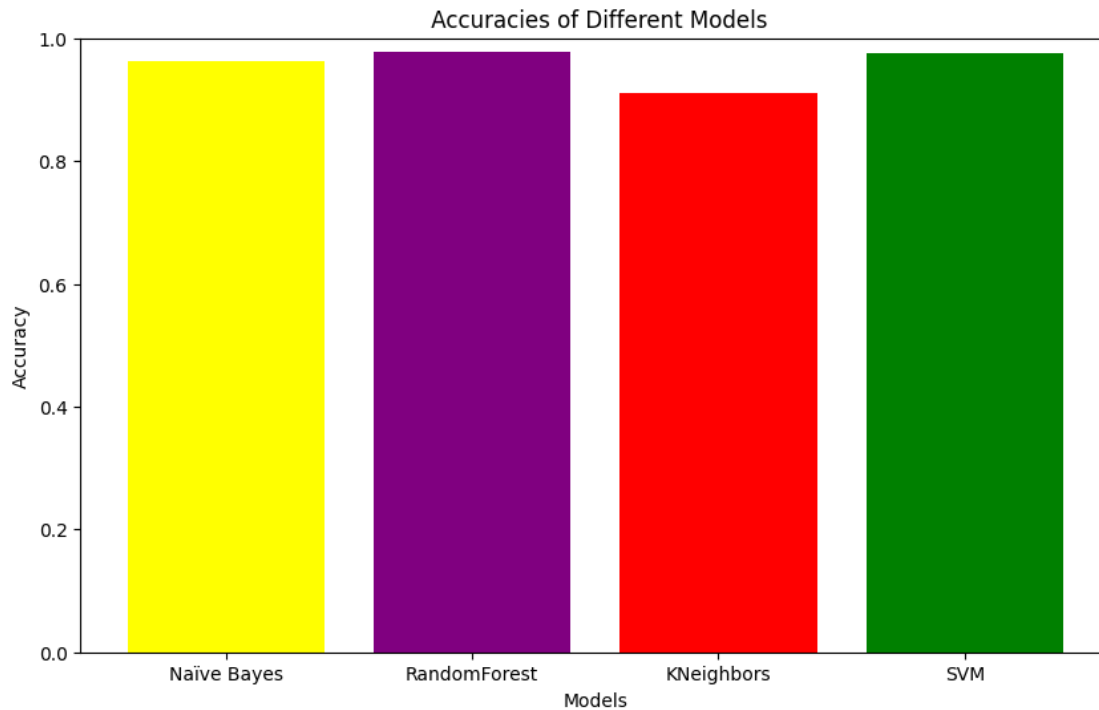
[36]: import matplotlib.pyplot as plt

# Cross-validation accuracies
cross_val_accuracies = []
for model in models:
    cross_val_acc = cross_val_score(model, X_train, y_train, cv=5,
    ↪scoring='accuracy').mean()
    cross_val_accuracies.append(cross_val_acc)

# Define colors for each model
colors = ['yellow', 'purple', 'red', 'green']

# Plotting the bar plot
plt.figure(figsize=(10, 6))
plt.bar(model_names, cross_val_accuracies, color=colors)
plt.title(' Accuracies of Different Models')
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.ylim(0, 1) # Set y-axis limit between 0 and 1
plt.show()

```



```
[38]: import re
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Load the data from the CSV file
file_path = '/content/drive/MyDrive/spam.csv'
data = pd.read_csv(file_path, encoding='ISO-8859-1')

# Display the first few rows of the data
print(data.head())

# Function to clean the text
def clean_text(text):
    cleaned_text = re.sub('[^a-zA-Z]', ' ', str(text))
    cleaned_text = cleaned_text.lower()
    cleaned_text = ' '.join(cleaned_text.split())
    return cleaned_text

# Clean the 'Text' column
data['Clean_Text'] = data['v2'].apply(clean_text)
```

```

# Encoding the 'Target' variable (assuming 'ham' as 0 and 'spam' as 1)
data['Target'] = data['v1'].map({'ham': 0, 'spam': 1})

# Creating TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=5000) # You can adjust
↳max_features as needed

# Fitting the vectorizer on the entire dataset
X_tfidf = tfidf_vectorizer.fit_transform(data['Clean_Text']).toarray()
y = data['Target']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2,
↳random_state=42)

# Creating and training the RandomForest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Making predictions on the test set
y_pred = rf_model.predict(X_test)

# Calculate accuracy on the test set
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on the test set: {accuracy:.2%}")

```

	v1		v2	Unnamed: 2	\
0	ham	Go until jurong point, crazy.. Available only ...		NaN	
1	ham	Ok lar... Joking wif u oni...		NaN	
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...		NaN	
3	ham	U dun say so early hor... U c already then say...		NaN	
4	ham	Nah I don't think he goes to usf, he lives aro...		NaN	

Unnamed: 3 Unnamed: 4

0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

Accuracy on the test set: 97.58%

```

[40]: # Input Section
new_sms = input("Enter the SMS text: ")

# Cleaning the input text
cleaned_sms = clean_text(new_sms)

```



```

# Transforming the cleaned text into a TF-IDF vector
new_sms_tfidf = tfidf_vectorizer.transform([cleaned_sms]).toarray()

# Making the prediction using the trained RandomForest model
prediction = rf_model.predict(new_sms_tfidf)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on the test set: {accuracy:.2%}")

# Displaying the result
if prediction == 0:
    print("The SMS is classified as 'ham'.")
else:
    print("The SMS is classified as 'spam'.")

```

Enter the SMS text: e are thrilled to inform you that you have been selected as the lucky winner of our monthly prize draw! You have won an all-expenses-paid vacation to a luxurious resort in the Caribbean. To claim your prize, simply click on the link below and fill out the form with your personal details:
 Accuracy on the test set: 97.58%
 The SMS is classified as 'spam'.

```

[ ]: # Install necessary packages
!apt-get install -y texlive-xetex
!apt-get install -y texlive-fonts-recommended
!apt-get install -y texlive-generic-recommended
!apt-get install -y pandoc

# Install Python package for pandoc
!pip install py pandoc

# Convert Colab notebook to PDF
!jupyter nbconvert --to pdf "/content/drive/MyDrive/Colab Notebooks/Untitled2.
↪ipynb"

```

Reading package lists... Done
 Building dependency tree... Done
 Reading state information... Done
 The following additional packages will be installed:
 dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
 texgyre
 fonts-urw-base35 libapache-pom-java libcommons-logging-java libcommons-parent-
 java
 libfontbox-java libfontenc1 libgs9 libgs9-common libidn12 libijs-0.35
 libjbig2dec0 libkpathsea6
 libpdfbox-java libptexenc1 libruby3.0 libsynchronet2 libteckit0 libtexlua53
 libtexluajit2 libwoff1
 libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby ruby-net-

```

telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration tlutils teckit tex-
common tex-gyre
  texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base
texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa xfonts-
encodings
  xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java poppler-
utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-
ipafont-gothic
  fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper
gv
| postscript-viewer perl-tk xpdf | pdf-viewer xzdec texlive-fonts-recommended-
doc
  texlive-latex-base-doc python3-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc texlive-latex-
recommended-doc
  texlive-luatex texlive-pstricks dot2tex prerex texlive-pictures-doc vprerex
default-jre-headless
  tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcommons-logging-java libcommons-parent-
java
  libfontbox-java libfontenc1 libgs9 libgs9-common libidn12 libijs-0.35
libjbig2dec0 libkpathsea6
  libpdfbox-java libptexenc1 libruby3.0 libsynctex2 libteckit0 libtexlua53
libtexluajit2 libwoff1
  libzip-0-13 lmodern poppler-data preview-latex-style rake ruby ruby-net-
telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration tlutils teckit tex-
common tex-gyre
  texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base
texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic texlive-xetex
tipa
  xfonts-encodings xfonts-utils
0 upgraded, 54 newly installed, 0 to remove and 35 not upgraded.
Need to get 182 MB of archives.
After this operation, 571 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1

```

[2,696 kB]
Get:3 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 poppler-data all 0.4.11-1 [2,171 kB]
Get:4 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 tex-common all 6.17 [33.7 kB]
Get:5 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-urw-base35 all 20200910-1 [6,367 kB]
Get:6 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libgs9-common all 9.55.0~dfsg1-0ubuntu5.6 [751 kB]
Get:7 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libidn12 amd64 1.38-4ubuntu1 [60.0 kB]
Get:8 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libijs-0.35 amd64 0.35-15build2 [16.5 kB]
Get:9 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libjbig2dec0 amd64 0.19-3build2 [64.7 kB]
Get:10 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libgs9 amd64 9.55.0~dfsg1-0ubuntu5.6 [5,031 kB]
Get:11 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libkpathsea6 amd64 2021.20210626.59705-1ubuntu0.1 [60.3 kB]
Get:12 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libwoff1 amd64 1.0.2-1build4 [45.2 kB]
Get:13 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 dvisvgm amd64 2.13.1-1 [1,221 kB]
Get:14 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-lmodern all 2.004.5-6.1 [4,532 kB]
Get:15 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 fonts-noto-mono all 20201225-1build1 [397 kB]
Get:16 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 fonts-texgyre all 20180621-3.1 [10.2 MB]
Get:17 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libapache-pom-java all 18-1 [4,720 B]
Get:18 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-parent-java all 43-1 [10.8 kB]
Get:19 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libcommons-logging-java all 1.2-2 [60.3 kB]
Get:20 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 libfontenc1 amd64 1:1.1.4-1build3 [14.7 kB]
Get:21 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libptexenc1 amd64 2021.20210626.59705-1ubuntu0.1 [39.1 kB]
Get:22 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rubygems-integration all 1.18 [5,336 B]
Get:23 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby3.0 amd64 3.0.2-7ubuntu2.4 [50.1 kB]
Get:24 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-rubygems all 3.3.5-2 [228 kB]
Get:25 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby amd64 1:3.0~exp1 [5,100 B]
Get:26 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 rake all 13.0.6-2 [61.7

kB]

Get:27 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 ruby-net-telnet all 0.1.1-2 [12.6 kB]
Get:28 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 ruby-webrick all 1.7.0-3 [51.8 kB]
Get:29 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 ruby-xmlrpc all 0.3.2-1ubuntu0.1 [24.9 kB]
Get:30 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libruby3.0 amd64 3.0.2-7ubuntu2.4 [5,113 kB]
Get:31 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libsyntax2 amd64 2021.20210626.59705-1ubuntu0.1 [55.5 kB]
Get:32 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libteckit0 amd64 2.5.11+ds1-1 [421 kB]
Get:33 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexlua53 amd64 2021.20210626.59705-1ubuntu0.1 [120 kB]
Get:34 <http://archive.ubuntu.com/ubuntu> jammy-updates/main amd64 libtexluajit2 amd64 2021.20210626.59705-1ubuntu0.1 [267 kB]
Get:35 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libzip-0-13 amd64 0.13.72+dfsg.1-1.1 [27.0 kB]
Get:36 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 xfonts-encodings all 1:1.0.5-0ubuntu2 [578 kB]
Get:37 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 xfonts-utils amd64 1:7.7+6build2 [94.6 kB]
Get:38 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 lmodern all 2.004.5-6.1 [9,471 kB]
Get:39 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 preview-latex-style all 12.2-1ubuntu1 [185 kB]
Get:40 <http://archive.ubuntu.com/ubuntu> jammy/main amd64 t1utils amd64 1.41-4build2 [61.3 kB]
Get:41 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 teckit amd64 2.5.11+ds1-1 [699 kB]
Get:42 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 tex-gyre all 20180621-3.1 [6,209 kB]
Get:43 <http://archive.ubuntu.com/ubuntu> jammy-updates/universe amd64 texlive-binaries amd64 2021.20210626.59705-1ubuntu0.1 [9,848 kB]
Get:44 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 texlive-base all 2021.20220204-1 [21.0 MB]
Get:45 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 texlive-fonts-recommended all 2021.20220204-1 [4,972 kB]
Get:46 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 texlive-latex-base all 2021.20220204-1 [1,128 kB]
Get:47 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libfontbox-java all 1:1.8.16-2 [207 kB]
Get:48 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 libpdfbox-java all 1:1.8.16-2 [5,199 kB]
Get:49 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 texlive-latex-recommended all 2021.20220204-1 [14.4 MB]
Get:50 <http://archive.ubuntu.com/ubuntu> jammy/universe amd64 texlive-pictures

```

all 2021.20220204-1 [8,720 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 182 MB in 5s (36.7 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 121749 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2.1_all.deb ...
Unpacking fonts-lato (2.0-2.1) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.11-1_all.deb ...
Unpacking poppler-data (0.4.11-1) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.17_all.deb ...
Unpacking tex-common (6.17) ...
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack .../04-fonts-urw-base35_20200910-1_all.deb ...
Unpacking fonts-urw-base35 (20200910-1) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../05-libgs9-common_9.55.0~dfsg1-0ubuntu5.6_all.deb ...
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.6) ...
Selecting previously unselected package libidn12:amd64.
Preparing to unpack .../06-libidn12_1.38-4ubuntu1_amd64.deb ...
Unpacking libidn12:amd64 (1.38-4ubuntu1) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../07-libijs-0.35_0.35-15build2_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-15build2) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../08-libjbig2dec0_0.19-3build2_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.19-3build2) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../09-libgs9_9.55.0~dfsg1-0ubuntu5.6_amd64.deb ...
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../10-libkpathsea6_2021.20210626.59705-1ubuntu0.1_amd64.deb
...
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.1) ...

```

```

Selecting previously unselected package libwoff1:amd64.
Preparing to unpack .../11-libwoff1_1.0.2-1build4_amd64.deb ...
Unpacking libwoff1:amd64 (1.0.2-1build4) ...
Selecting previously unselected package dvisvgm.
Preparing to unpack .../12-dvisvgm_2.13.1-1_amd64.deb ...
Unpacking dvisvgm (2.13.1-1) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../13-fonts-lmodern_2.004.5-6.1_all.deb ...
Unpacking fonts-lmodern (2.004.5-6.1) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../14-fonts-noto-mono_20201225-1build1_all.deb ...
Unpacking fonts-noto-mono (20201225-1build1) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../15-fonts-texgyre_20180621-3.1_all.deb ...
Unpacking fonts-texgyre (20180621-3.1) ...
Selecting previously unselected package libapache-pom-java.
Preparing to unpack .../16-libapache-pom-java_18-1_all.deb ...
Unpacking libapache-pom-java (18-1) ...
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack .../17-libcommons-parent-java_43-1_all.deb ...
Unpacking libcommons-parent-java (43-1) ...
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack .../18-libcommons-logging-java_1.2-2_all.deb ...
Unpacking libcommons-logging-java (1.2-2) ...
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack .../19-libfontenc1_1%3a1.1.4-1build3_amd64.deb ...
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../20-libptexenc1_2021.20210626.59705-1ubuntu0.1_amd64.deb
...
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../21-rubygems-integration_1.18_all.deb ...
Unpacking rubygems-integration (1.18) ...
Selecting previously unselected package ruby3.0.
Preparing to unpack .../22-ruby3.0_3.0.2-7ubuntu2.4_amd64.deb ...
Unpacking ruby3.0 (3.0.2-7ubuntu2.4) ...
Selecting previously unselected package ruby-rubygems.
Preparing to unpack .../23-ruby-rubygems_3.3.5-2_all.deb ...
Unpacking ruby-rubygems (3.3.5-2) ...
Selecting previously unselected package ruby.
Preparing to unpack .../24-ruby_1%3a3.0~exp1_amd64.deb ...
Unpacking ruby (1:3.0~exp1) ...
Selecting previously unselected package rake.
Preparing to unpack .../25-rake_13.0.6-2_all.deb ...
Unpacking rake (13.0.6-2) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../26-ruby-net-telnet_0.1.1-2_all.deb ...

```

```

Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-webrick.
Preparing to unpack .../27-ruby-webrick_1.7.0-3_all.deb ...
Unpacking ruby-webrick (1.7.0-3) ...
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack .../28-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb ...
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack .../29-libruby3.0_3.0.2-7ubuntu2.4_amd64.deb ...
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.4) ...
Selecting previously unselected package libsyntax2:amd64.
Preparing to unpack .../30-libsyntax2_2021.20210626.59705-1ubuntu0.1_amd64.deb
...
Unpacking libsyntax2:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack .../31-libteckit0_2.5.11+ds1-1_amd64.deb ...
Unpacking libteckit0:amd64 (2.5.11+ds1-1) ...
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack .../32-libtexlua53_2021.20210626.59705-1ubuntu0.1_amd64.deb
...
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Selecting previously unselected package libtexluaajit2:amd64.
Preparing to unpack
.../33-libtexluaajit2_2021.20210626.59705-1ubuntu0.1_amd64.deb ...
Unpacking libtexluaajit2:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Selecting previously unselected package libzip-0-13:amd64.
Preparing to unpack .../34-libzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb ...
Unpacking libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Selecting previously unselected package xfonts-encodings.
Preparing to unpack .../35-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb ...
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) ...
Selecting previously unselected package xfonts-utils.
Preparing to unpack .../36-xfonts-utils_1%3a7.7+6build2_amd64.deb ...
Unpacking xfonts-utils (1:7.7+6build2) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../37-lmodern_2.004.5-6.1_all.deb ...
Unpacking lmodern (2.004.5-6.1) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../38-preview-latex-style_12.2-1ubuntu1_all.deb ...
Unpacking preview-latex-style (12.2-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../39-t1utils_1.41-4build2_amd64.deb ...
Unpacking t1utils (1.41-4build2) ...
Selecting previously unselected package teckit.
Preparing to unpack .../40-teckit_2.5.11+ds1-1_amd64.deb ...
Unpacking teckit (2.5.11+ds1-1) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../41-tex-gyre_20180621-3.1_all.deb ...

```

```

Unpacking tex-gyre (20180621-3.1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../42-texlive-binaries_2021.20210626.59705-1ubuntu0.1_amd64.deb ...
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.1) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../43-texlive-base_2021.20220204-1_all.deb ...
Unpacking texlive-base (2021.20220204-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../44-texlive-fonts-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-fonts-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../45-texlive-latex-base_2021.20220204-1_all.deb ...
Unpacking texlive-latex-base (2021.20220204-1) ...
Selecting previously unselected package libfontbox-java.
Preparing to unpack .../46-libfontbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libfontbox-java (1:1.8.16-2) ...
Selecting previously unselected package libpdfbox-java.
Preparing to unpack .../47-libpdfbox-java_1%3a1.8.16-2_all.deb ...
Unpacking libpdfbox-java (1:1.8.16-2) ...
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack .../48-texlive-latex-recommended_2021.20220204-1_all.deb ...
Unpacking texlive-latex-recommended (2021.20220204-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../49-texlive-pictures_2021.20220204-1_all.deb ...
Unpacking texlive-pictures (2021.20220204-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../50-texlive-latex-extra_2021.20220204-1_all.deb ...
Unpacking texlive-latex-extra (2021.20220204-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../51-texlive-plain-generic_2021.20220204-1_all.deb ...
Unpacking texlive-plain-generic (2021.20220204-1) ...
Selecting previously unselected package tipa.
Preparing to unpack .../52-tipa_2%3a1.3-21_all.deb ...
Unpacking tipa (2:1.3-21) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../53-texlive-xetex_2021.20220204-1_all.deb ...
Unpacking texlive-xetex (2021.20220204-1) ...
Setting up fonts-lato (2.0-2.1) ...
Setting up fonts-noto-mono (20201225-1build1) ...
Setting up libwoff1:amd64 (1.0.2-1build4) ...
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Setting up libijs-0.35:amd64 (0.35-15build2) ...
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Setting up libfontbox-java (1:1.8.16-2) ...
Setting up rubygems-integration (1.18) ...
Setting up libzip-0-13:amd64 (0.13.72+dfsg.1-1.1) ...
Setting up fonts-urw-base35 (20200910-1) ...

```



```

Setting up poppler-data (0.4.11-1) ...
Setting up tex-common (6.17) ...
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) ...
Setting up libjbig2dec0:amd64 (0.19-3build2) ...
Setting up libteckit0:amd64 (2.5.11+ds1-1) ...
Setting up libapache-pom-java (18-1) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) ...
Setting up t1utils (1.41-4build2) ...
Setting up libidn12:amd64 (1.38-4ubuntu1) ...
Setting up fonts-texgyre (20180621-3.1) ...
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Setting up ruby-webrick (1.7.0-3) ...
Setting up fonts-lmodern (2.004.5-6.1) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) ...
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) ...
Setting up libsynchronet2:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.6) ...
Setting up teckit (2.5.11+ds1-1) ...
Setting up libpdfbox-java (1:1.8.16-2) ...
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) ...
Setting up preview-latex-style (12.2-1ubuntu1) ...
Setting up libcommons-parent-java (43-1) ...
Setting up dvisvgm (2.13.1-1) ...
Setting up libcommons-logging-java (1.2-2) ...
Setting up xfonts-utils (1:7.7+6build2) ...
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.1) ...
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.1) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) ...
Setting up texlive-base (2021.20220204-1) ...
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper

```

```

tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) ...
Setting up texlive-plain-generic (2021.20220204-1) ...
Setting up texlive-latex-base (2021.20220204-1) ...
Setting up texlive-latex-recommended (2021.20220204-1) ...
Setting up texlive-pictures (2021.20220204-1) ...
Setting up texlive-fonts-recommended (2021.20220204-1) ...
Setting up tipa (2:1.3-21) ...
Setting up texlive-latex-extra (2021.20220204-1) ...
Setting up texlive-xetex (2021.20220204-1) ...
Setting up rake (13.0.6-2) ...
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.4) ...
Setting up ruby3.0 (3.0.2-7ubuntu2.4) ...
Setting up ruby (1:3.0~exp1) ...
Setting up ruby-rubygems (3.3.5-2) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

Processing triggers for tex-common (6.17) ...
Running updmap-sys. This may take some time... done.
Running mktexlsr /var/lib/texmf ... done.
Building format(s) --all.
    This may take some time...

```