DATA X

NLP Module: Learning Models
NLP Process

Text Processing
Clean up the text to make it easier to use and more consistent to increase prediction accuracy later on.

Feature Engineering & Text Representation
Learn how to extract information from text and represent it numerically.

Learning Models
Use learning models to identify parts of speech, entities, sentiment, and other aspects of the text.
NLP Examples using Learning Models

- **Spam Ham Classification**: Classifying emails into spam or ham
- **Sentiment Analysis**: Classifying tweets as negative or positive
- **Generative Chatbot**: Creating a generative based medical chatbot
Spam Ham Classification
Spam Ham Classification - Emails

Classify emails as spam (1) or ham (0)

*text processing*: stemming, removed stopwords, removing special characters

*feature engineering & text representation*: bag of words model

*learning model*: multinomial naive bayes

Link to full project:
https://www.kaggle.com/balakishan77/spam-or-ham-email-classification
Other Applications

Fake news detection

Text messages classification

Flower type

...

Really anything that can be categorized based on features
Sentiment Analysis
Sentiment Analysis - Tweets

Classify Tweets as having positive, negative, or no sentiment

<table>
<thead>
<tr>
<th>Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

*text processing: regex (remove hyperlinks, twitter handles, hashtags, retweets)
Sentiment Analysis - Tweets

# Create a function to get the subjectivity
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

# Create a function to get the polarity
def getPolarity(text):
    return TextBlob(text).sentiment.polarity

# Create two new columns 'Subjectivity' & 'Polarity'
df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)
df['Polarity'] = df['Tweets'].apply(getPolarity)

# Show the new dataframe with columns 'Subjectivity' & 'Polarity'
df

if score < 0:
    return 'Negative'
elif score == 0:
    return 'Neutral'
else:
    return 'Positive'

*feature engineering & text representation*: get subjectivity & polarity scores

*classification*: based on subjectivity and polarity

Other Applications

*Any classification having to do with sentiment/emotions*

Exp:

- Yelp Reviews
- Movie Reviews
- Emails
Generative Chatbot
Generative Based Chatbot - Medical

Generative models are not based on predefined responses but are based on seq 2 seq neural networks

Dataset:

```json
{"tag": "noanswer",
 "patterns": [],
 "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"]
},
{"tag": "options",
 "patterns": ["How you could help me?", "What you can do?", "What help you provide?", "How can I guide you through Adverse drug reaction list, Blood pressure tracking"],
 "context": ["""]
},
{"tag": "adverse_drug",
 "patterns": ["How to check Adverse drug reaction?", "Open adverse drugs module", "Give me more"]
 "responses": ["Navigating to Adverse drug reaction module"]
}
```
Generative Based Chatbot - Medical

*text processing*: lemmatizing, lower each word, and remove duplicates

*feature engineering & text representation*: bag of words model

*learning model*: Sequential model (seq 2 seq) with 3 layers

Link to full project: https://drive.google.com/file/d/1763Y5zy7HmRYsOoBLQgLxQRE6xGCqQN/view
Other Applications

Other Chatbots

Email word recommendation

Anything that has text output based off on text input