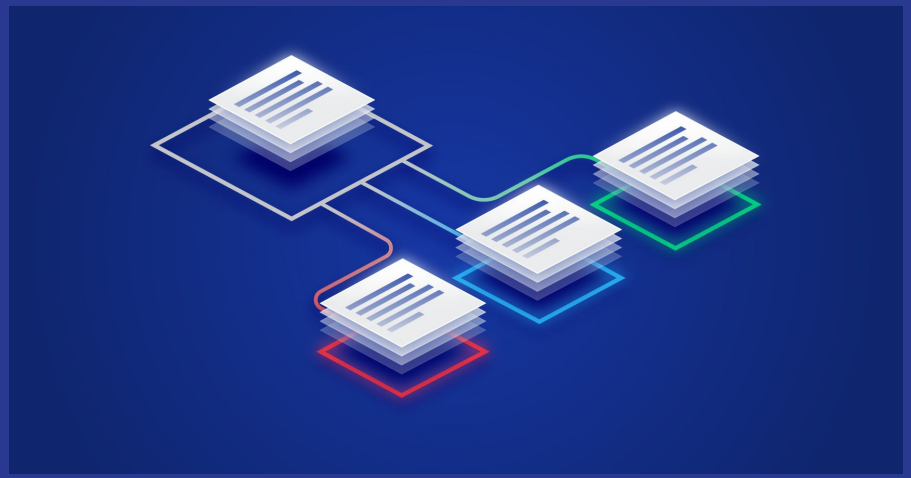
Text Classification

Haseeb Chaudhury
Animesh Deb
Sri TarunGulumuru
Al Shafian Bari

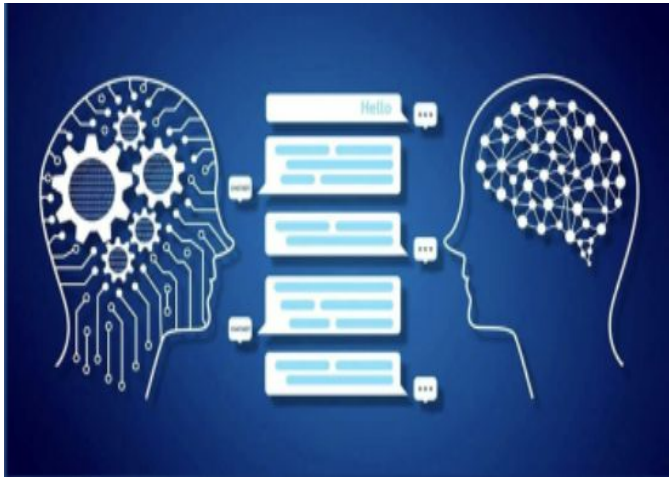


What is Text Classification

- One of the Fundamentals of Natural Language Processing.

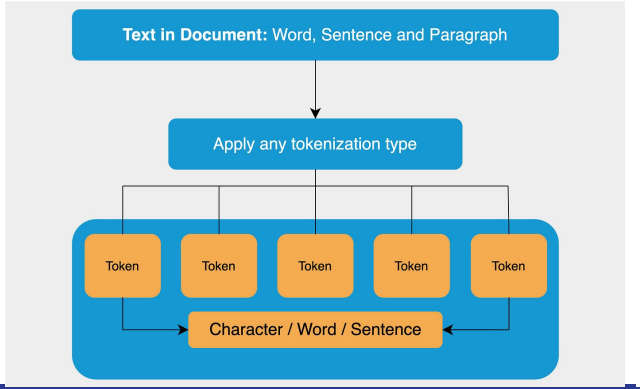
  - Natural Language is how we speak

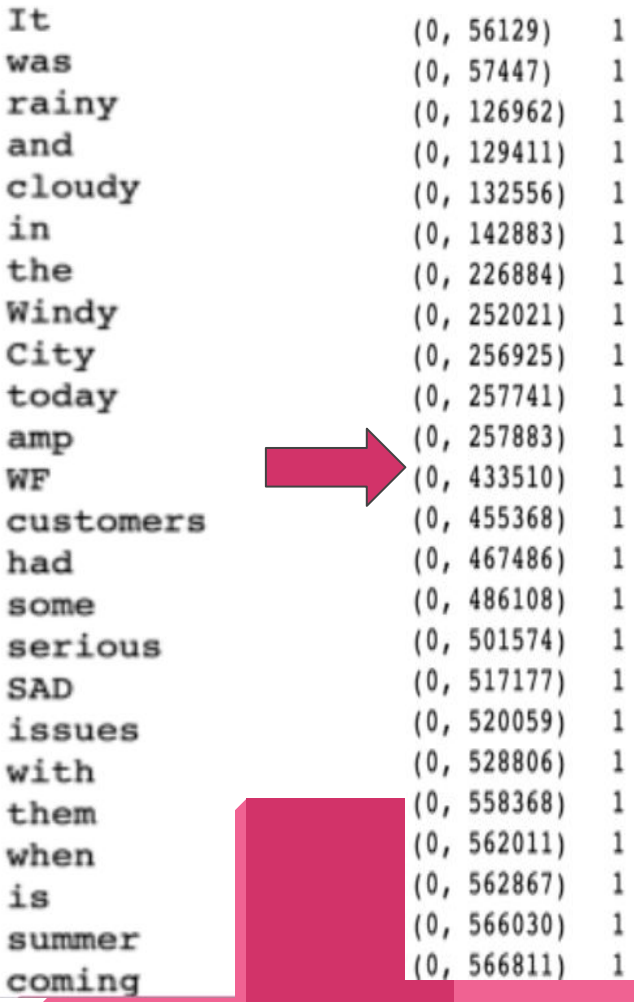
- There is so much data to analyze nowadays
- We need machines to analyze for us
- Machine do not know the meaning behind words



How does text classification work?

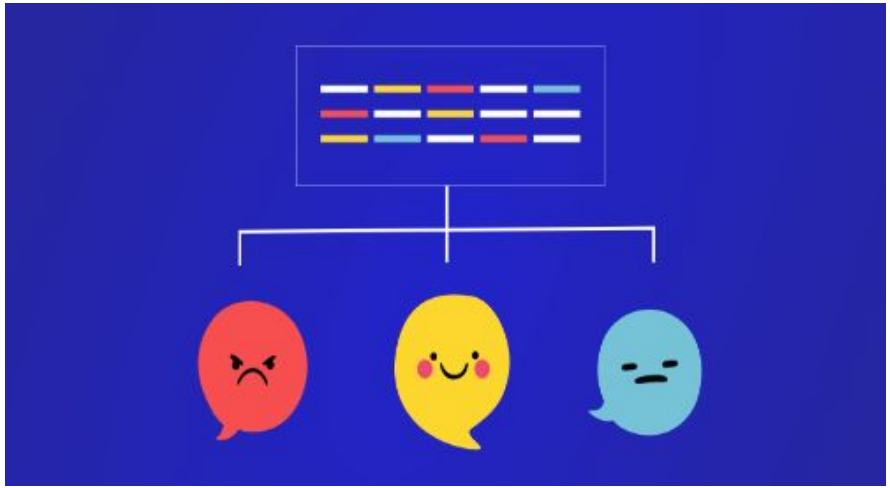
- The model must understand language and phase structure
- Methods to help machines understand human communication:
  - vectorization
    - Term frequency-inverse document frequency (Tf-idf) vectorization
    - Count vectorization
  - Tokenization - divide text into sub texts
  - Stemming - break down text to its basic stem
  - Lemmatization - break down text to its root





Text Classification Examples

- Sentiment Analysis
- Topic labeling
- Spam detection
- Intent detection



Dataset Used

- The Sentiment140 dataset with 1.6 million tweets
- This is a popular dataset used for text classification
- Target and Text fields are the primary targets of observation
- Negative text is classified with a zero (0) and positive text is classified with a one (1)

	Target	ID	Date	Flag	User	Text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
...	...	...	...	...	...	...
1599995	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599996	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	happy #charitytuesday @theNSPCC @SparksCharity...

```
df.info()
```

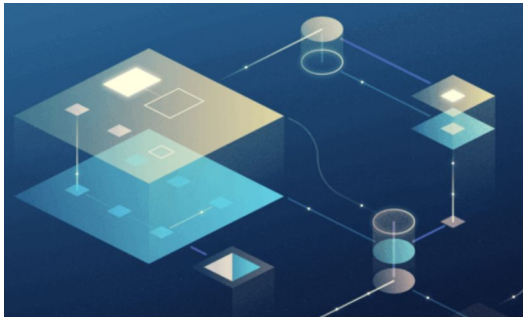
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600000 entries, 0 to 1599999
Data columns (total 6 columns):
 #   Column  Non-Null Count    Dtype
---  ------  --------------    -----
 0   Target  1600000 non-null  int64
 1   ID      1600000 non-null  int64
 2   Date    1600000 non-null  object
 3   Flag    1600000 non-null  object
 4   User    1600000 non-null  object
 5   Text    1600000 non-null  object
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
```

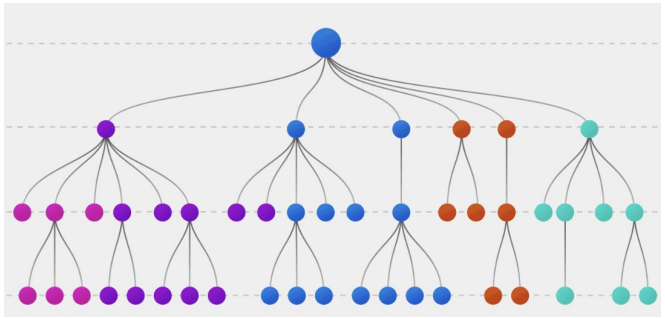
Models Used

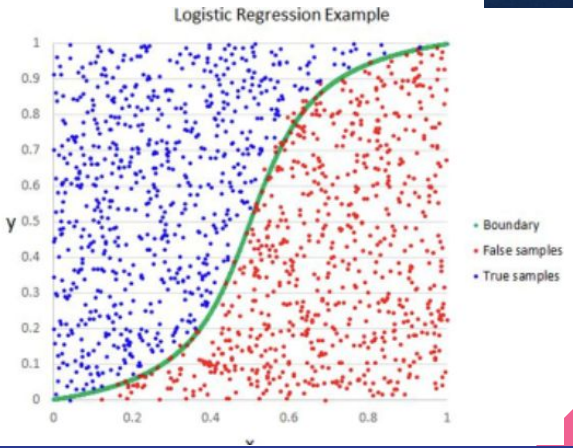
$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

These are the models that classified text with:

- Logistic Regression
- Naive Bayes
- Tensorflow Neural Network
- Decision Trees





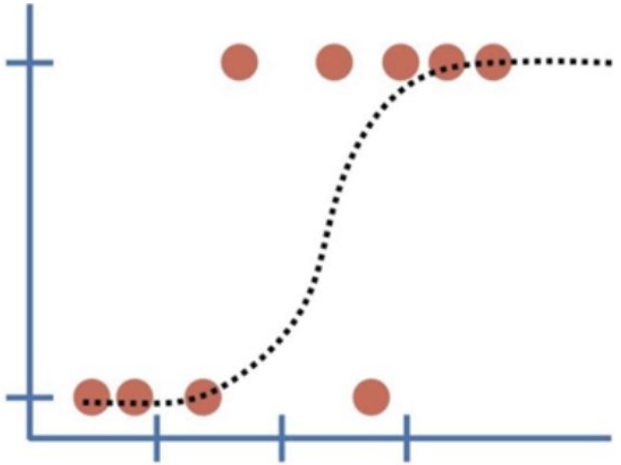


Logistic Regression

- Logistic Regression is a binary classification model, originally used in statistics and chosen for testing due to its widespread use and familiarity.
- Dataset imported, specifically selecting text and sentiment value fields, with an 80-20 split for training and testing.
- Utilized Count Vectorization, resulting in an accuracy of 0.7983125, with 126,496 true positives and 129,606 true negatives, indicating strong performance in classifying sentiment.
- Tested TF-IDF vectorization, yielding a higher accuracy of 0.802515625, with 131,229 true positives and 116,348 true negatives, showcasing improved performance over Count Vectorization.
- TF-IDF effectiveness in reducing the impact of "stopping words" benefited the Logistic Regression model, contributing to its overall strong performance in sentiment classification.

Code

```
df = pd.read_csv(DATASET)
X = df['Text'].values
Y = df['Target'].values
training = train_test_split(X, Y, test_size = 0.20, random_state=32)

# Count Vectorization
NB = make_pipeline(CountVectorizer(), LogisticRegression())
NB.fit(x_train, y_train)
score = NB.score(x_test, y_test)
print(score)

# Tf-idf Vectorization
NB = make_pipeline(TfidfVectorizer(), LogisticRegression())
NB.fit(x_train, y_train)
score = NB.score(x_test, y_test)
print(score)
```



Logistic Regression Results

Count Vectorization: 0.80031875

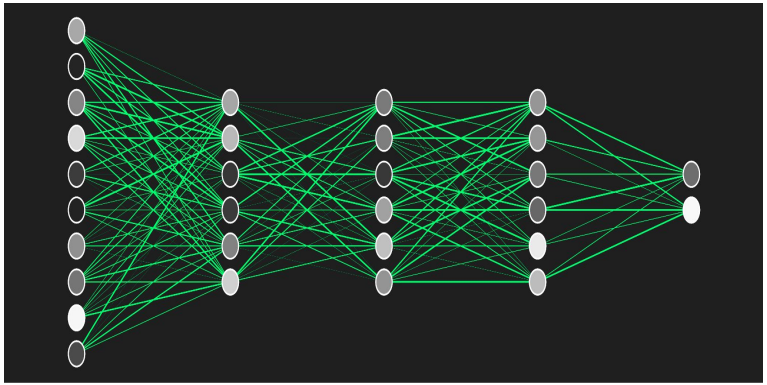
Classified Target \ Actual Target	0	4
0	126496	33825
4	30073	129606

TF-IDF Vectorization: 0.802515625

Classified Target \ Actual Target	0	4
0	131229	29092
4	43331	116348

Neural Network

- TensorFlow neural network designed for unsupervised learning, requiring extensive training and data breakdown compared to other models.
- Data cleanup involved removing stopping words, punctuation, duplicates, email addresses, URL links, and numbers to enhance analysis.
- Further data breakdown through tokenization, stemming, and lemmatization aimed at facilitating the neural network classification.
- Neural network architecture includes an input layer, seven hidden layers for understanding data, and an output layer for results.
- Despite multiple iterations to improve the model over time, the final accuracy was 0.745550, the lowest among all models; potential improvements include more iterations, better training ratios, and additional feature extraction methods for enhanced text classification performance.

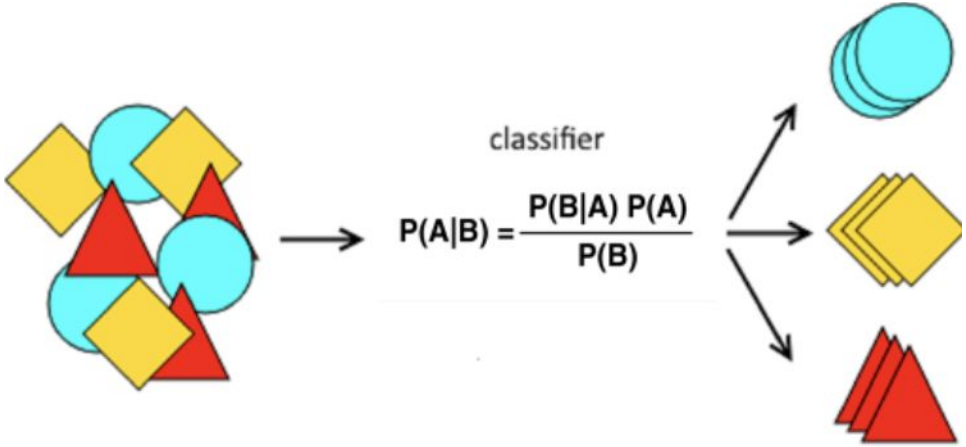


Neural Network Results

```
In [46]: model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])

In [47]: history=model.fit(X_train,Y_train,batch_size=80,epochs=6, validation_split=0.1)

Epoch 1/6
WARNING:tensorflow:From C:\Users\chaud\anaconda3\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTe
nsorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\chaud\anaconda3\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.execut
ing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

360/360 [==============================] - 114s 312ms/step - loss: 0.5990 - accuracy: 0.6662 - val_loss: 0.5415 - val_accuracy:
0.7234
Epoch 2/6
360/360 [==============================] - 113s 315ms/step - loss: 0.5106 - accuracy: 0.7547 - val_loss: 0.5257 - val_accuracy:
0.7300
Epoch 3/6
360/360 [==============================] - 113s 314ms/step - loss: 0.4933 - accuracy: 0.7634 - val_loss: 0.5248 - val_accuracy:
0.7344
Epoch 4/6
360/360 [==============================] - 112s 312ms/step - loss: 0.5135 - accuracy: 0.7511 - val_loss: 0.5271 - val_accuracy:
0.7406
Epoch 5/6
360/360 [==============================] - 113s 313ms/step - loss: 0.4905 - accuracy: 0.7689 - val_loss: 0.5263 - val_accuracy:
0.7381
Epoch 6/6
360/360 [==============================] - 108s 300ms/step - loss: 0.4663 - accuracy: 0.7801 - val_loss: 0.5394 - val_accuracy:
0.7387

In [50]: accr1 = model.evaluate(X_test,Y_test)

250/250 [==============================] - 14s 55ms/step - loss: 0.5289 - accuracy: 0.7455

In [51]: print(accr1[1])

0.7455000281333923

In [ ]:
```

Naive Bayes

- Naïve Bayes model is popular for text classification, excelling with high-dimensional data and effective feature extraction.
- Characterized by higher bias and lower variance compared to Logistic Regression, making it suitable with less data.
- Based on a blend of classification algorithms following Bayes' Theorem.
- Tested using Count Vectorization and TF-IDF Vectorization, with Count Vectorization achieving an accuracy of 0.780703125, slightly lower than Logistic Regression.
- Despite a minor dip in accuracy with TF-IDF Vectorization (0.773678125), Naïve Bayes maintains reasonable accuracy, indicating its robust performance in text classification.



Code

```
df = pd.read_csv(DATASET)
X = df['Text'].values
Y = df['Target'].values
training = train_test_split(X, Y, test_size = 0.20, random_state=32)

# Count Vectorization
NB = make_pipeline(CountVectorizer(), MultinomialNB())
NB.fit(x_train, y_train)
score = NB.score(x_test, y_test)


# Tf-idf Vectorization
NB = make_pipeline(TfidfVectorizer(), MultinomialNB())
NB.fit(x_train, y_train)
score = NB.score(x_test, y_test)
print(score)
```

Naive Bayes Results

Count Vectorization: 0.780703125

Classified Target \ Actual Target	0	4
0	131110	29211
4	40964	118715

TF-IDF Vectorization: 0.802515625

Classified Target \ Actual Target	0	4
0	131229	29092
4	43331	116348

Decision Trees

- Decision tree structure: Root node, decision nodes, branches, and leaf nodes form the hierarchy.
- Decision tree building process: Involves steps like feature selection, dataset splitting, recursive processes, and optional pruning.
- Evaluation outputs: Accuracy reveals correct predictions, confusion matrix breaks down results, and a qualitative assessment inspects individual instances.
- Purpose of evaluation: Assessing generalization and identifying biases or areas for improvement in the model.
- Overall, the text emphasizes the anatomy, construction, and evaluation of decision trees in machine learning.

Decision Tree Results

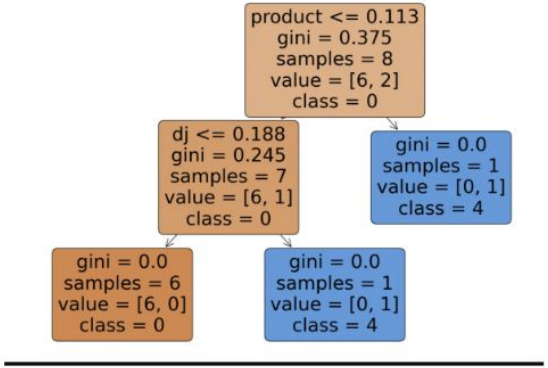


Figure 12: Decision Tree

```
Accuracy on subset: 100.00%
Confusion Matrix on subset:
[[2]]
```

	Text	Target	Predicted
541200	@chrishasboobs AHHH I HOPE YOUR OK!!!	0	0
750	@misstoriblack cool , i have no tweet apps fo...	0	0
766711	@TiannaChaos i know just family drama. its la...	0	0
285055	School email won't open and I have geography ...	0	0
705995	upper airways problem	0	0
379611	Going to miss Pastor's sermon on Faith...	0	0
1189018	on lunch....dj should come eat with me	4	4
667030	@piginthepoke oh why are you feeling like that?	0	0
93541	gahh noo!peyton needs to live!this is horrible	0	0
1097326	@mrstessyman thank you glad you like it! There...	4	4

Figure 11: Matrix on Subset

Conclusion

- Text classification is fundamental for Natural Language Processing, allowing machines to understand human communication in text efficiently.
- Data preprocessing involves removing unnecessary words and details, and techniques like tokenization, stemming, and lemmatization break down text further.
- Vectorization methods like Count Vectorization and TF-IDF Vectorization assign weights based on word frequency.
- Tested models include Decision Tree, Logistic Regression, Naive Bayes, and a Recurrent Neural Network (RNN) using TensorFlow.
- Logistic Regression with TF-IDF Vectorization performed best with an accuracy score of 0.802515625, while the neural network had the lowest accuracy (0.7457500100135803), indicating method-model interactions.

Thank You