Gabriel, Neil, Steve, Riya

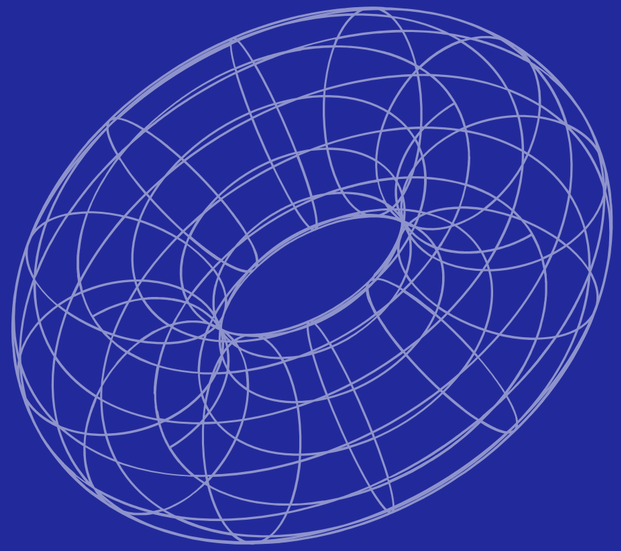
Spam Email Detection

Using Natural Language Processing to detect spam emails

Capstone Presentation

Overview

Do we still need **spam detection?**



Modern spam detection algorithms are often able to detect spam emails with a very high rate of accuracy.

However, false positives are becoming a big concern amongst businesses, organisations, and companies as their legitimate emails end up in the spam folder.

Previous Spam Detection Models

- Bayesian algorithms, often used in the early days of email in the 1990s.
- Uses bag of words features and correlating words with spam/ham.
- One of the oldest and efficient ways for spam filtering.

Modern Spam Detection Models

- Complex algorithms that are always evolving.
- Making use of LSTMs and other models suited for NLP tasks.
- Better performance with increasing sophistication of spam emails.

Project Overview

Step 1 Exploratory data analysis

Understanding the dataset as spam and 'ham' (not spam) emails; common features; potential issues.

Step 2 Dataset cleaning

Cleaning text and selecting records with relevant, readable information.

Step 3 Data preprocessing

Using NLP preprocessing techniques including stemming and stopword removal to create corpus.

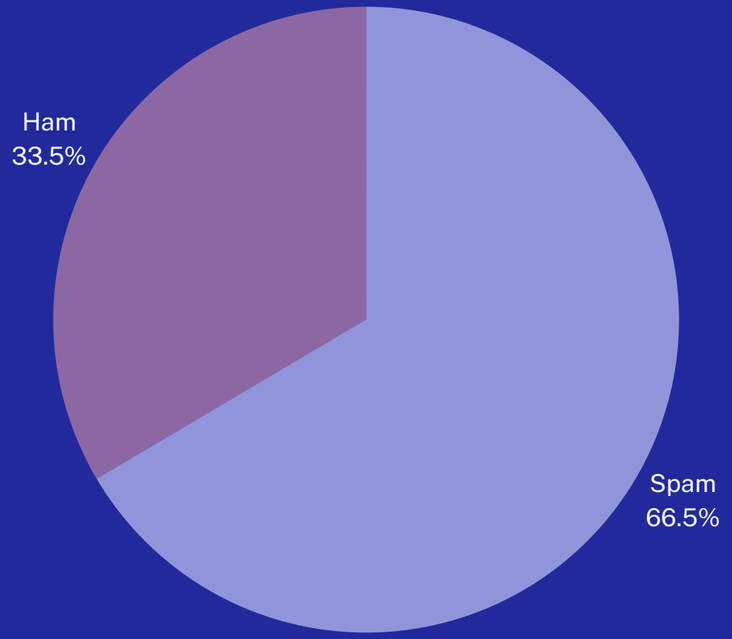
Step 4 ML and DL Model development

Building ML and DL models using cleaned dataset; overfitting analysis.

Step 5 Streamlit application

Deploying model to Streamlit to allow users to upload their emails to detect for spam detection.

Spam Email Dataset



Original Dataset

The original dataset we used for building this model and project had a slightly imbalanced dataset of 66.5% spam emails and 33.5% ham emails.

Data Sourcing

Preprocessed TREC 2007 Public Corpus Dataset

- **75,419 real** emails made publicly available by the University of Waterloo
- **50199** spam emails
- **25220** ham (not spam) emails
- **4 features:** email subject, email receipient, email sender, message body
- **1 target:** spam or not spam

label	subject	email_to	email_from	message
1	the reply for your request for a job place [le...	"Gnitpick" <gnitpick@flax9.uwaterloo.ca>	"Sydney Car Centre" <Merrill8783@168city.com>	Content-Type: text/html;\nContent-Transfer-Enc...
1	the reply for your request for a job place [le...	"Gnitpick" <gnitpick@flax9.uwaterloo.ca>	"Sydney Car Centre" <Merrill8783@168city.com>	Content-Type: text/html;\nContent-Transfer-Enc...
0	Re: [R] Me again, about the horrible documenta...	Duncan Murdoch <murdoch@stats.uwo.ca>	Philippe Grosjean <phgrosjean@sciviews.org>	For those who are interested, I just cook a li...
0	Re: [R] RODBC problem	<r-help@stat.math.ethz.ch>	=?iso-8859-1?Q?Bernhard_Wellh=F6fer?=\n\t<Bern...	Hello,\n\nas I wrote I call\n\n sqlFetch(chan...
1	I wanted the desk at his own laws: of the. Bu...	the00@plg.uwaterloo.ca	"Danny" <pwcusnt@noblecoffee.com>	Content-Type: multipart/alternative;\n\tbounda...

Dataset Preprocessing

Dataset size shrunk to about 33,800 records after preprocessing.

Remove null & duplicate records

Over 12,000 records removed as a result due to many duplicates and null values.

Remove messages containing "Content-Type"

Unreadable HTML 'content' embedded into emails, which cannot be understood as language.

Drop sender, receipient, and subject columns

Focus on email message body to determine spam/not spam, as it contains the most textual information.

Dataset Preprocessing

Dataset size shrunk to about 33,800 records after preprocessing.

label	subject	message
0	Typo in /debian/README	Hi, i've just updated from the gulus and I che...
1	Nice talking with ya	Hey Billy, \n\nit was really fun going out the...
0	[R] Confidence-Intervals.... help...	Hi...\n\nI have to use R to find out the 90% c...
1	hi man	Hey Billy, \n\nit was really fun going out the...
0	Re: [R] Confidence-Intervals.... help...	Hm... sounds like a homework problem to me...\...
...	...	...
1	1-4 extra inches makes a massive difference, e...	MIME Ver: 1.79\nX-OriginalArrivalTime: Fri, 06...
0	Re: [R] Me again, about the horrible documenta...	On 06/07/2007 3:51 AM, Mike Meredith wrote:\n>...
0	Bug#431883: dcraw license does not give permis...	On Fri, 06 Jul 2007, Steve King wrote:\n> > Yo...
0	Re: [R] Me again, about the horrible documenta...	For those who are interested, I just cook a li...
0	Re: [R] RODBC problem	Hello,\n\nas I wrote I call\n\n sqlFetch(chan...

33832 rows × 3 columns

Text Data Cleaning

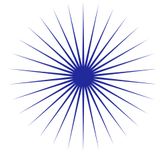
Cleaning message text to correct inconsistencies, unreadable content, and improve text vectorization.

Lowercase all message text

Prevent case-sensitive word parsing when creating the vocabulary.

Replace newlines with spaces

Format text to prevent newlines from separating message bodies.

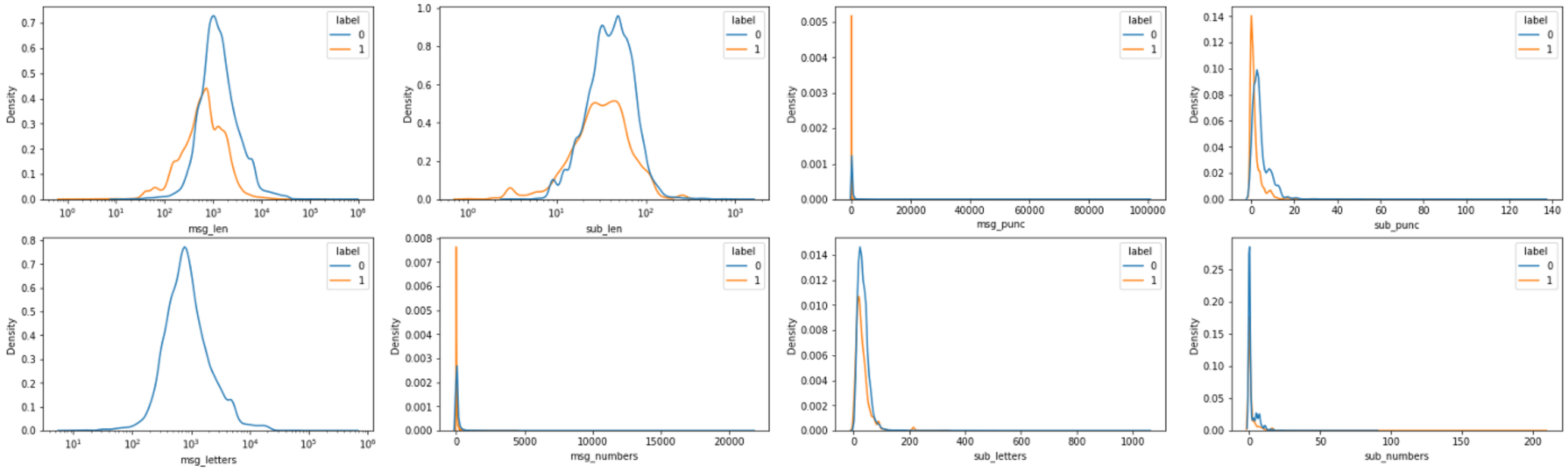


Remove records with unreadable content.

Conduct EDA to find records that are completely unreadable, but not null.

Text Data Cleaning

EDA using numerical features of text data (length, count of punctuation/numbers/letters) demonstrated significant overlap.

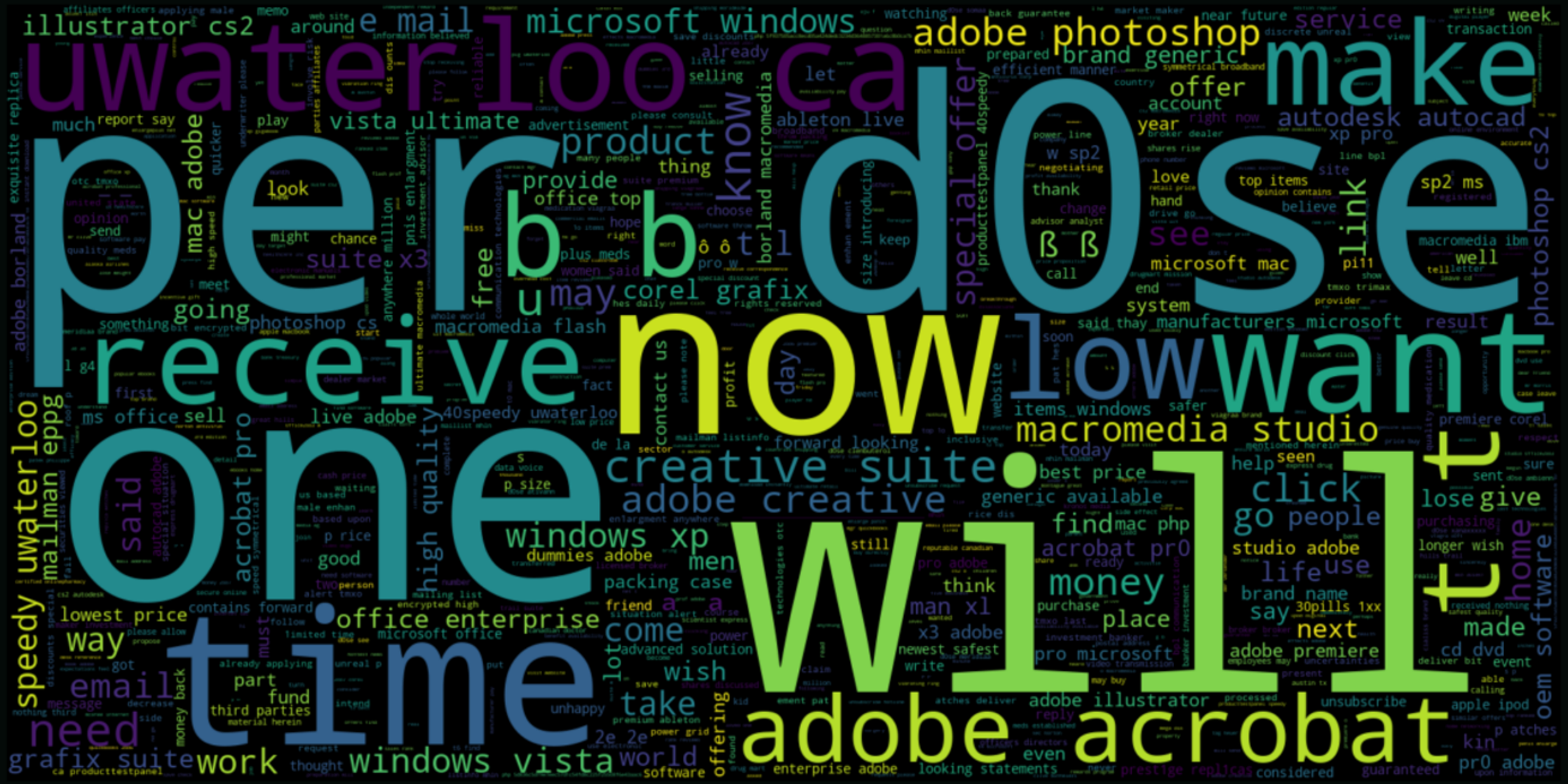


Text Data Cleaning

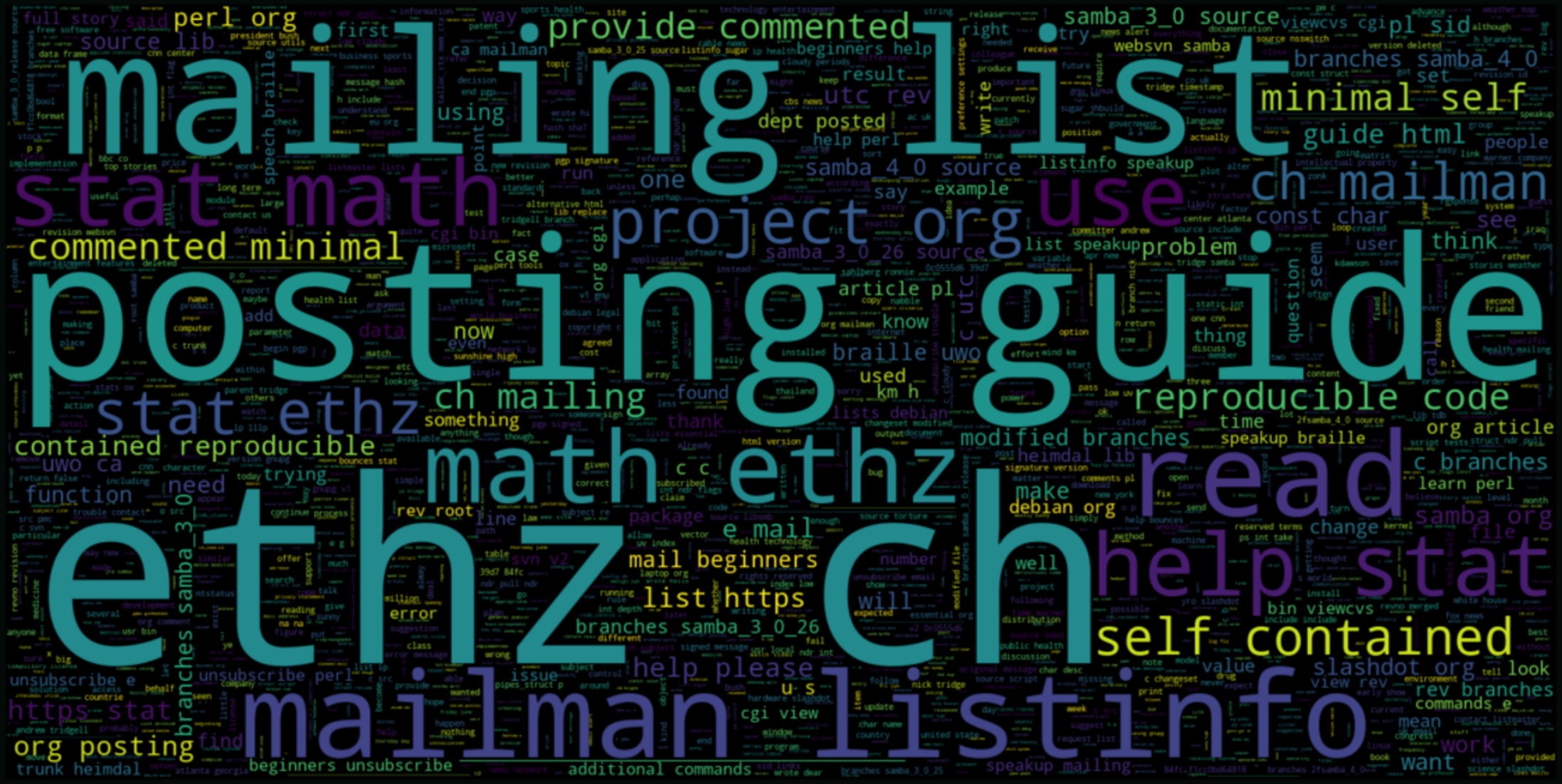
Records were found to have 0 (ASCII) letters but significant message length. These contained unreadable information, and records were removed.

	label	message_clean	subject_clean	content	msg_len	sub_len	msg_punc	sub_punc	msg_letters	msg_numbers	s
51	1	×ð¾´µä¹ó¹«ë¾áìµ¼/²æîñ£ºäú°ã£¡ îòêçéîã¯£¨éî...	=?gb2312?b?sk7wrrzso6gjoq==?=	=?gb2312?b?sk7wrrzso6gjoq==?= = ×ð¾´µä¹ó¹«ë¾áìµ¼...	685	29	4	8	0	13	
8481	1	äú°ã£¡ iiii iiii±¾¹«ë¾³éáç¶àäêóð×åá¼°ãµäéç»á¹ø...	=?gb2312?b?xvpstc+y0by=?=	=?gb2312?b?xvpstc+y0by=?= äú°ã£¡ iiii iiii±¾¹«...	440	25	1	8	0	11	
11104	1	ïèéúäú°ã£ºîòë¾óå»ý´ú¿³½¨öþ°²×° ¡¢æû³µðþàí¡¢ôëêä...	=?gb2312?b?tpq/qreixre=?=	=?gb2312?b?tpq/qreixre=?= ïèéúäú°ã£ºîòë¾óå»ý´ú...	184	25	0	8	0	23	
16828	1	·ð é½ êð ìú ïè êµ òµ óð ïþ ¹« ë¾ ...	=?gb2312?b?0rxo8ceiycwh?=	=?gb2312?b?0rxo8ceiycwh?= ·ð é½ êð ìú ïè ...	1213	25	109	6	0	34	
16898	1	×ð¾´µä¸°ôðèë£¨¾àí£¯²æîñ£©äú°ã£¡ ±¾¹«...	=?gb2312?b?xvpstddfz6i=?=	=?gb2312?b?xvpstddfz6i=?= ×ð¾´µä¸°ôðèë£¨¾àí£¯...	862	25	3	7	0	11	

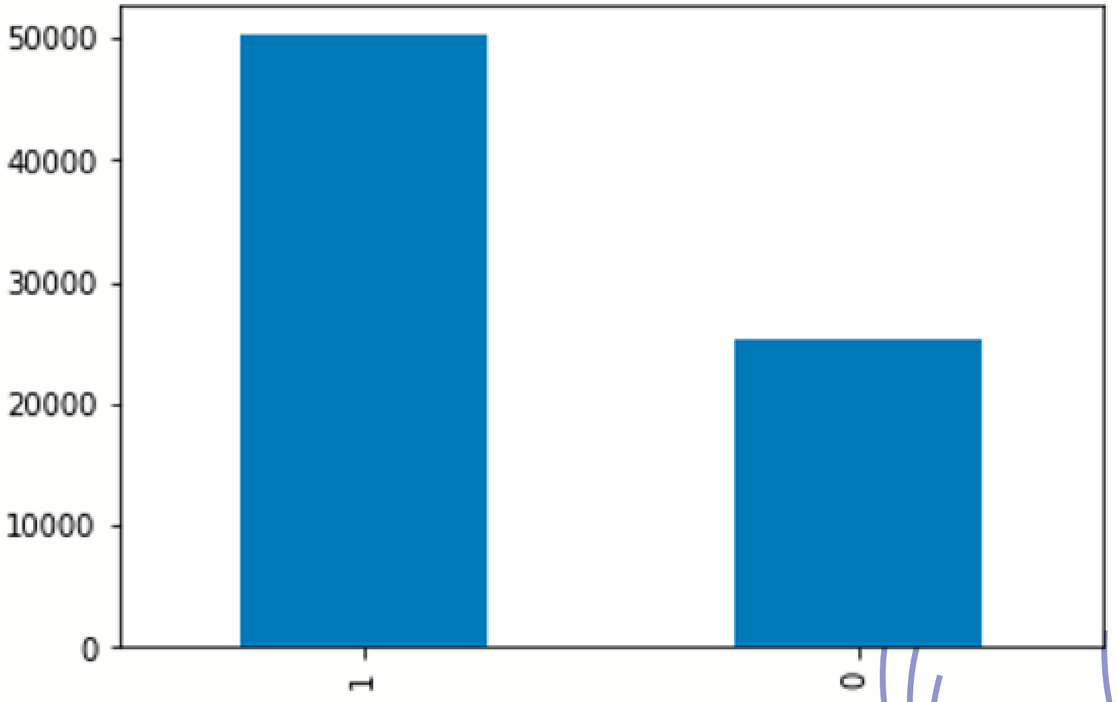
Spam Email Message Word Cloud



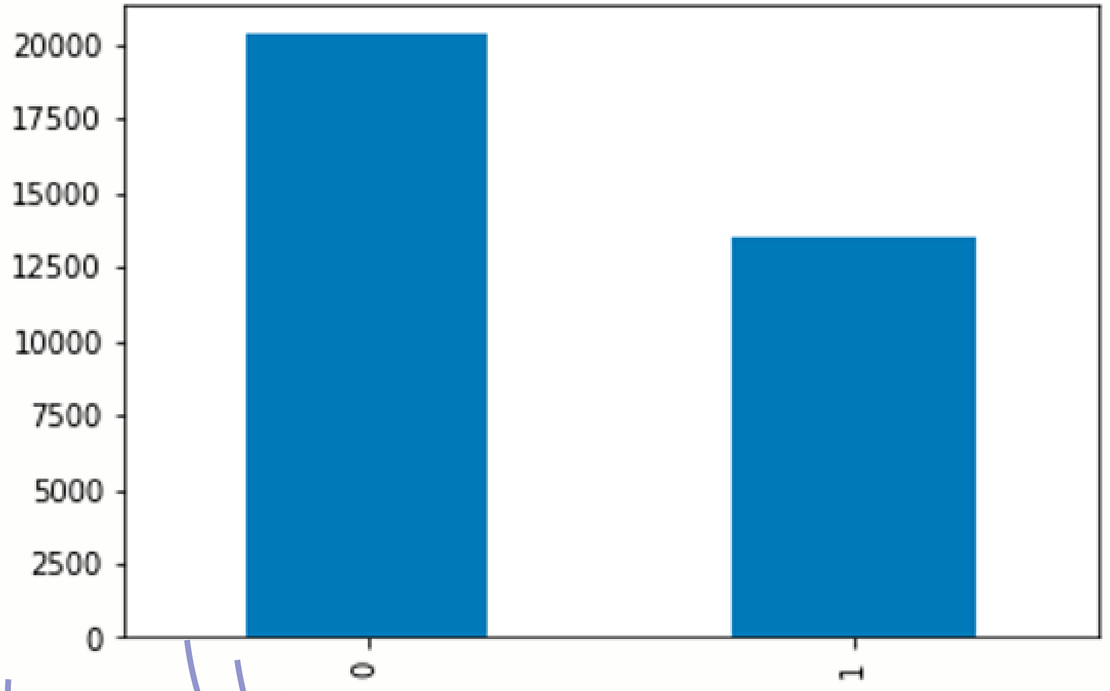
Not Spam Email Message Word Cloud



Data: Before and After Preprocessing and Cleaning




```
1    66.560151
0    33.439849
Name: label, dtype: float64
```




```
0    60.201263
1    39.798737
Name: label, dtype: float64
```

Preprocessing Text Data for ML

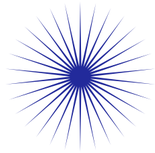
Steps taken to preprocess the message text data prior to model creation.

Replace all punctuation with a space

Differentiate between words easier and allow for proper stemming.

Remove common stop words from text

Stop words are frequently used but do not provide much information about class of email.



Stem all words in message body

Shorten words to their root to reduce corpus size and complexity.

Preprocessing Text Data for ML

Message body text after preprocessing.

```
dfml['message_clean_stem'] = dfml['message_clean'].apply(stemming)
dfml.head()
```

	label	message_clean	message_clean_stem
10844	0	on 4/16/2007 8:19 pm, jiho.han wrote: > dear r...	4 16 2007 8 19 pm jiho han wrote dear r expert...
22225	1	yo gnitpick!!. a genuine university degree in ...	yo gnitpick genuin univers degre notim ever th...
45933	0	hi, assume that we may model the nottingham t...	hi assum may model nottingham temperatur data ...
14313	1	oem software: throw packing case, leave cd/dvd...	oem softwar throw pack case leav cd dvd use el...
21792	0	hi, i'm using latex() from frank harrell's hm...	hi use latex frank harrel hmisc librari produc...

Model Development

- **Machine Learning Models**

  Logistic Regression, SVM, Naive Bayes

- **Artifical Neural Networks**

  A neural network was created for the purposes of classification.

- **Long-Short Term Memory**

  Variant of RNN for contextualizing words; more sophisticated and better suited for NLP tasks.

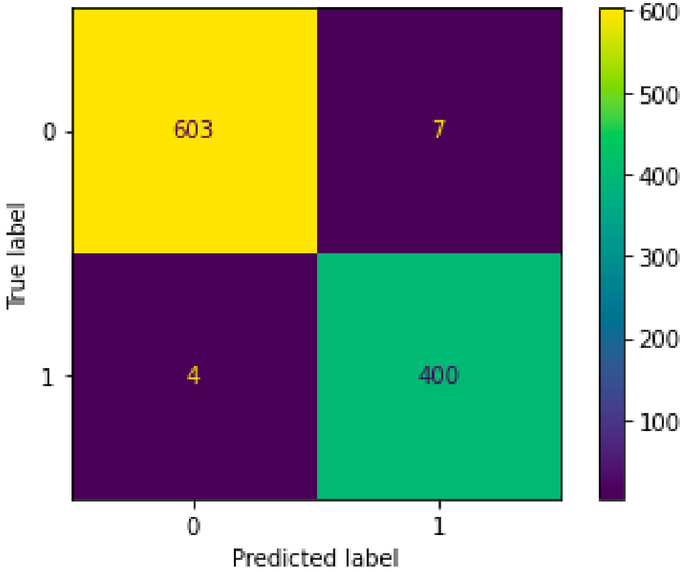
Vectorization

Tf-idf (Term Frequency–Inverse Document Frequency) Vectorizer

- Numerical statistic reflects how important a word is to a document in a collection or corpus. Calculates the term frequency, and the inverse document frequency. It then assigns different levels of importance to difference words.

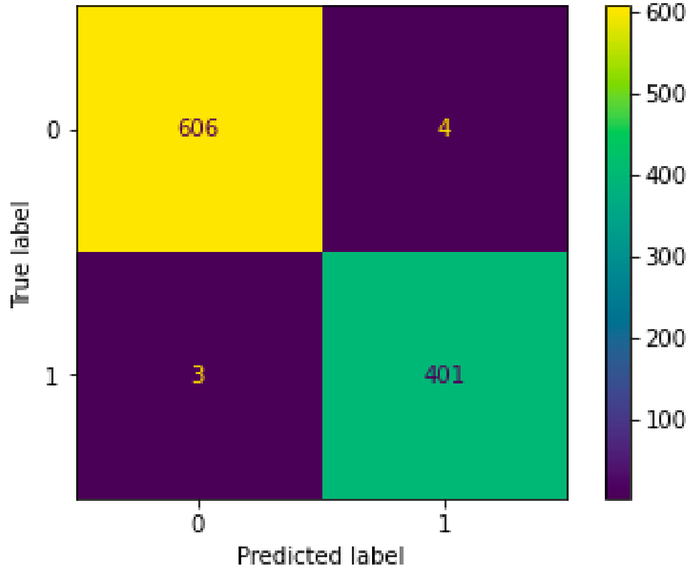
```
(0, 44471)    0.04969181837359509
(0, 23349)    0.07488079099705662
(0, 9865)     0.0738888598952085
(0, 19266)    0.09189293099242675
(0, 33724)    0.0576302855845526
(0, 35683)    0.12558118296575002
(0, 14205)    0.04484848726175382 6
```

Machine Learning Models



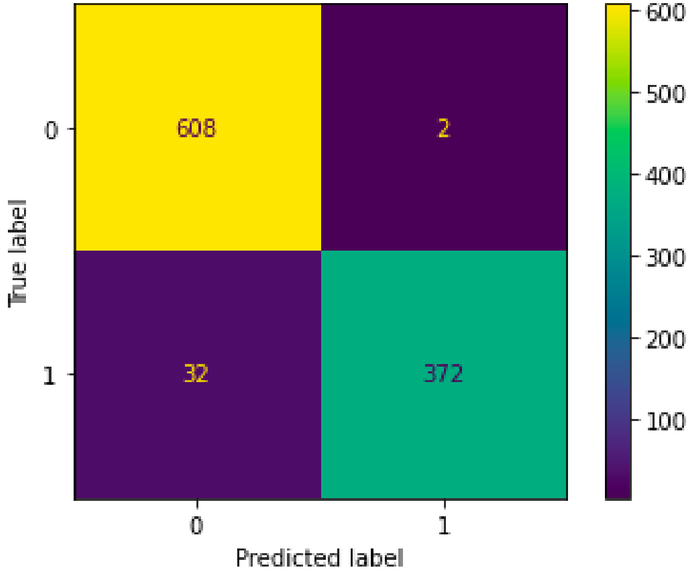
Logistic Regression

- Accuracy: 98.92%
- Precision: 99%
- Recall: 99%



Support Vector Machines

- Accuracy: 99.31%
- Precision: 99%
- Recall: 99%



Naive Bayes

- Accuracy: 96.65%
- Precision: 97%
- Recall: 97%

Overfitting analysis: Logistic regression

```
1 #cross-validation of logistic regression
2 score_LR_model=cross_val_score(LR_model, X_train, y_train,cv=10, scoring='accuracy')
3 print(score_LR_model)
4 print("Avg :",np.average(score_LR_model))
5
```

```
0.99630086 0.98890259 0.99260173 0.99753391 0.99012346 0.9962963
0.99012346 0.99382716 0.98765432 0.99506173]
vg : 0.992842550729932 7
```

Overfitting analysis: Naive Bayes Classification Model

```
1 #cross-validation of Naive Bayes Classification Model
2 score_NB_model=cross_val_score(NB_model, X_train, y_train,cv=10, scoring='accuracy')
3 print(score_NB_model)
4 print("Avg :",np.average(score_NB_model))
```

Loading...

```
[0.9864365  0.97410604 0.96300863 0.97657213 0.97777778 0.97777778
 0.9654321  0.9691358  0.97407407 0.97283951]
Avg : 0.973716034159305
```

Overfitting analysis: Support Vector Machine Classification Model

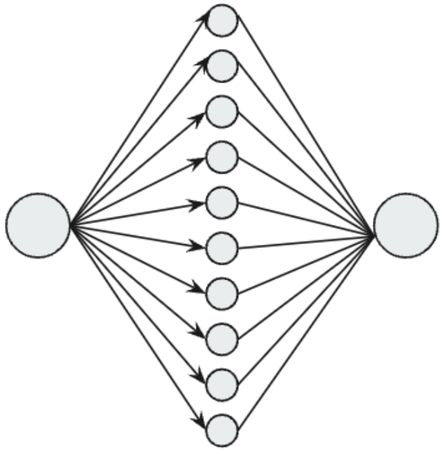
```
1 #cross-validation of Support Vector Machine Classification Model
2 score_SVC_model=cross_val_score(SVC_model, X_train, y_train,cv=10, scoring='accuracy')
3 print(score_SVC_model)
4 print("Avg :",np.average(score_SVC_model))
```

```
[0.99630086 0.99260173 0.99506782 0.99630086 0.99506173 0.99753086
 0.99259259 0.9962963  0.99012346 0.99753086]
Avg : 0.9949407072506127
```

Deep Learning

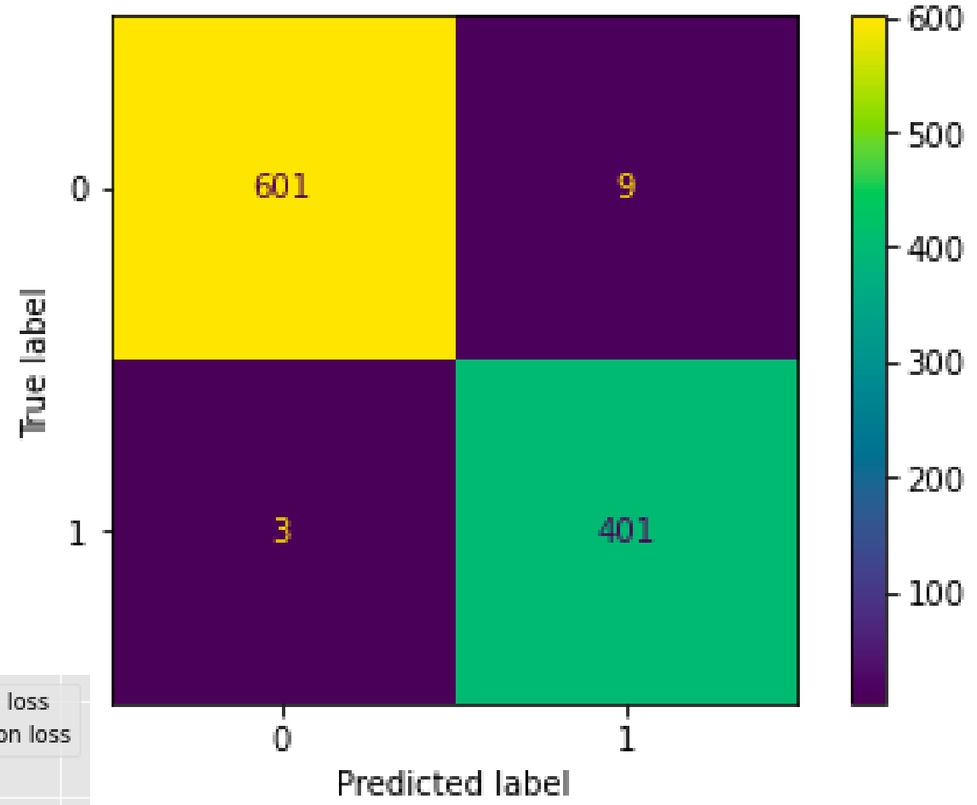
Artificial Neural Networks

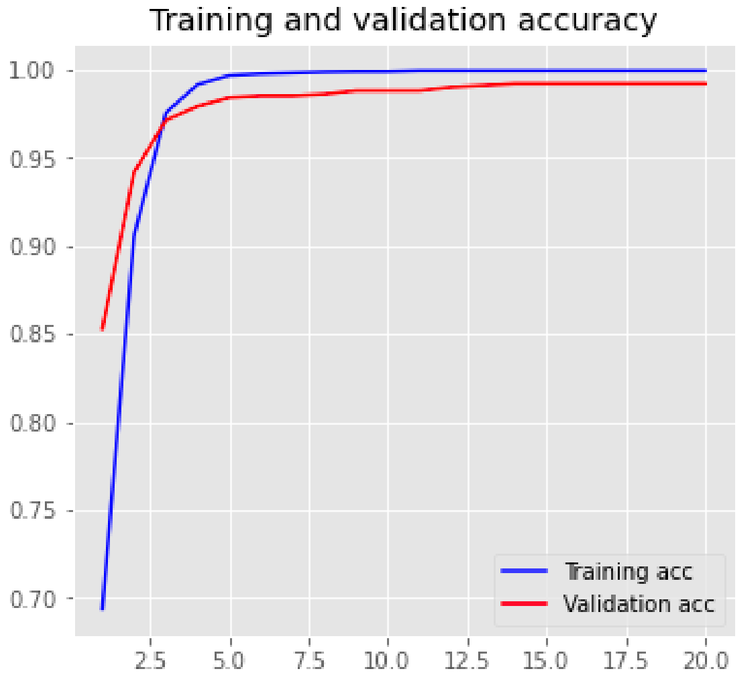
- In addition to the ML models created, a basic neural network was created for the purposes of classification. It consists of the input layer, and one dense layer with 10 neurons, followed by an output layer. Trained for 20 epochs.

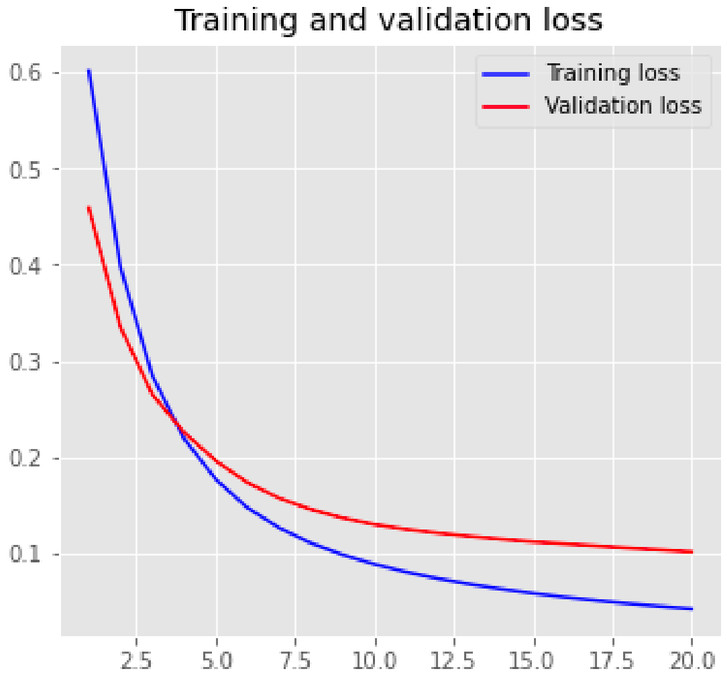


ANN Performance

- Training accuracy: 99.90%
- Validation accuracy: 98.82%
- Recall: 99%
- Precision: 99%







Overfitting analysis: ANN classifier

```
from keras.callbacks import EarlyStopping, TensorBoard

# create subdirectory to visualize EarlyStopping  in tensorboard
logdir = os.path.join("logs", "EarlyStopping-Loss")
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

# Adding EarlyStopping call back
stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0.01, patience=3)
```

```
history = model.fit(X_train, y_train,
                    epochs = 20,
                    verbose = False,
                    validation_data = (X_test, y_test),
                    batch_size = 1000,
                    callbacks=[tensorboard_callback, stop])
```

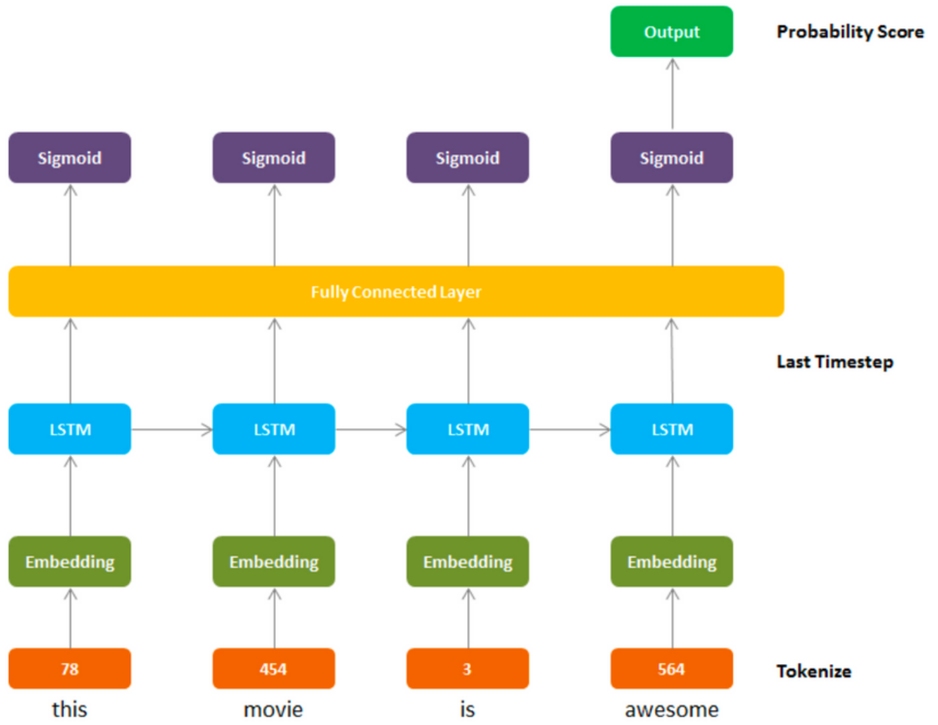
Overfitting analysis: ANN classifier

```
model = Sequential()
model.add(layers.Dense(10, kernel_regularizer=regularizers.l2(l2=0.01),input_dim=input_dim, a
model.add(layers.Dense(1, activation = 'sigmoid'))
```

Deep Learning

Long Short-Term Memory

- Variant of RNN for contextualizing words
- Bidirectional parameter
  - Check the context early and later down the text
  - Choose the best ones
  - Merge them



LSTM Process

01

Use a tokenizer and fit it on the training data messages.

02

Pad the sequences so they are all the same length, up to a maximum of 525 words.

03

Use Label Encoder to encode the y variables.

04

Create an embedding index and matrix, use it to create an embedding layer.

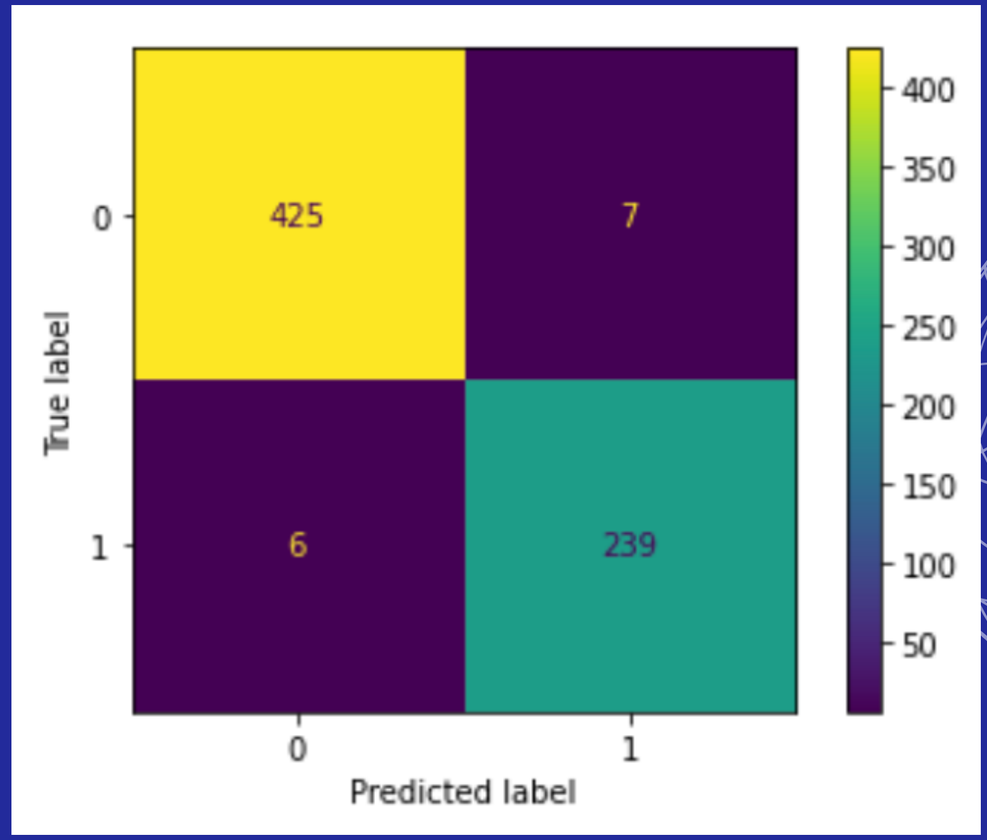
05

Create the LSTM model, train and test.

Long Short Term Memory Model

- Accuracy: 98.07%
- Precision: 99%
- Recall: 98%

- 7 ham emails predicted as spam
- 6 spam emails predicted as ham



Overfitting analysis: Bidirectional LSTM

```
from keras.callbacks import EarlyStopping, TensorBoard

# create subdirectory to visualize EarlyStopping  in tensorboard
logdir = os.path.join("logs", "EarlyStopping-Loss")
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

# Adding EarlyStopping call back
stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0.01, patience=3)
```

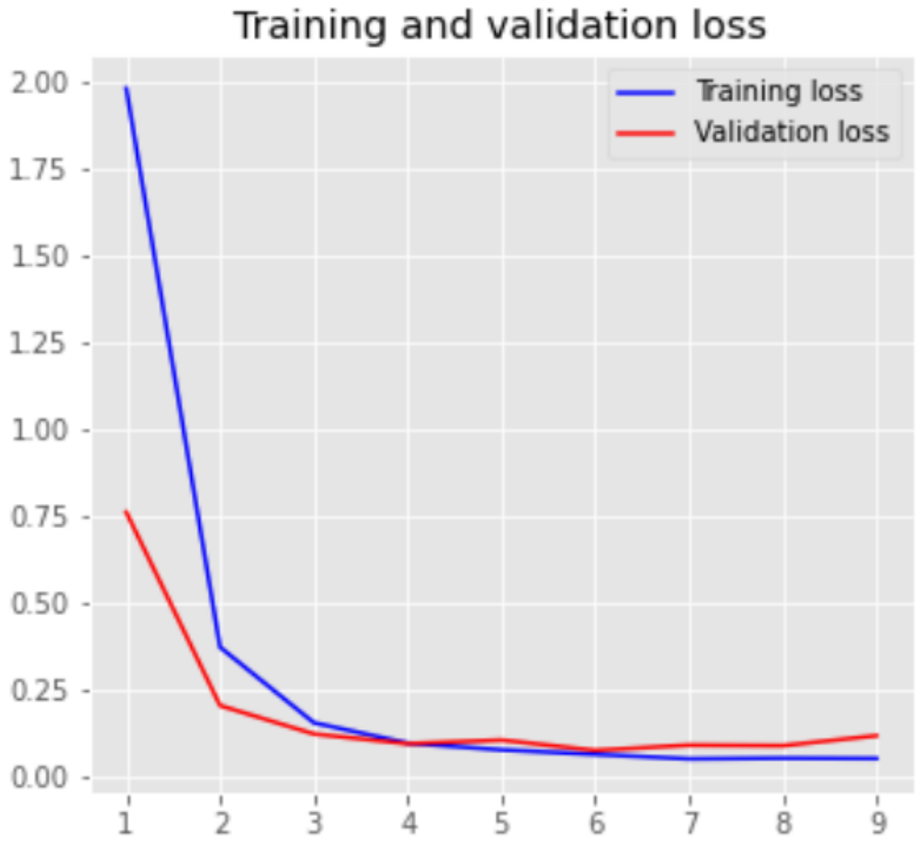
```
embedding_sequences = embedding_layer(sequence_input)
x = SpatialDropout1D(0.2)(embedding_sequences)
x = Conv1D(64, 5, activation='relu')(x)
x = Bidirectional(LSTM(64, dropout=0.2, recurrent_dropout=0.2))(x)
x = Dense(512, kernel_regularizer=regularizers.l2(l2=0.01), activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512,kernel_regularizer=regularizers.l2(l2=0.01), activation='relu')(x)
outputs = Dense(1, activation='sigmoid')(x)
model = tf.keras.Model(sequence_input, outputs)
```

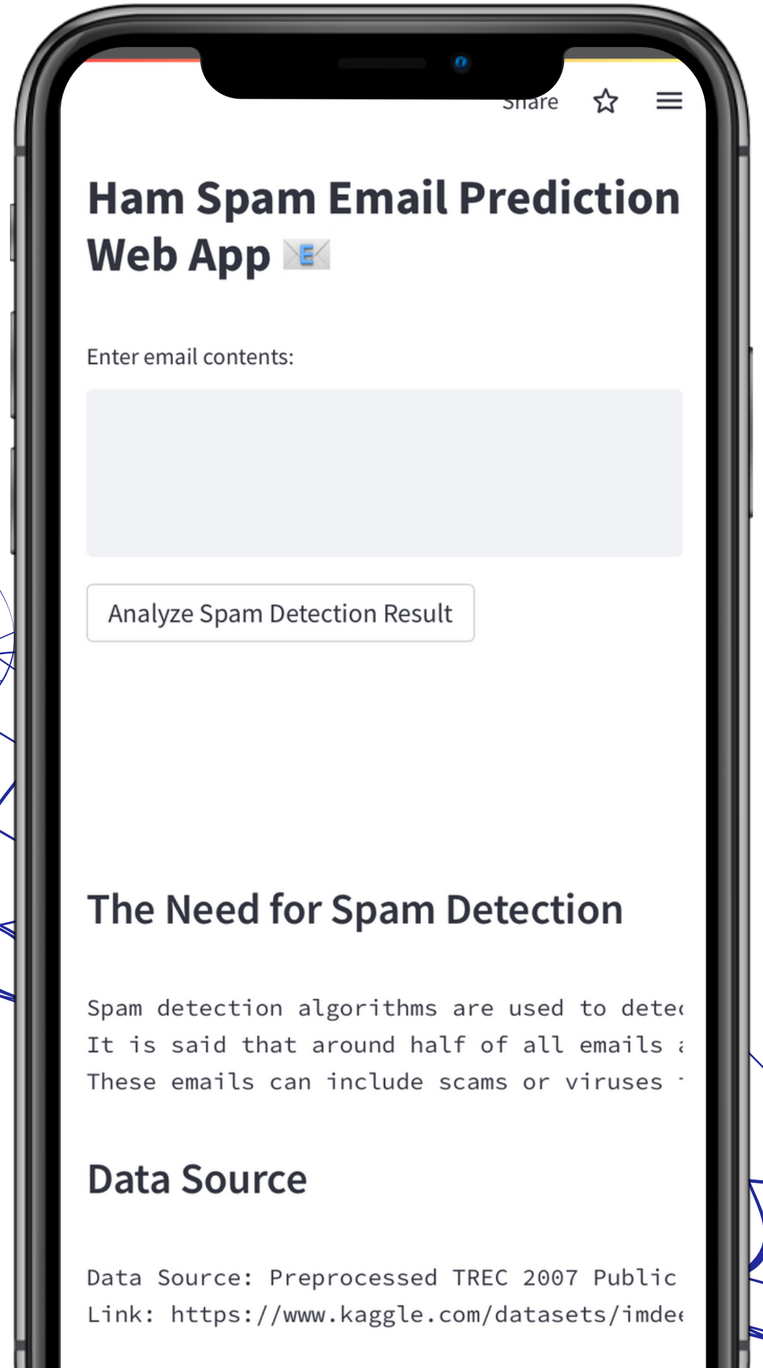
Overfitting analysis: Bidirectional LSTM

```
ReduceLROnPlateau = ReduceLROnPlateau(factor=0.1,
                                      min_lr = 0.01,
                                      monitor = 'val_loss',
                                      verbose = 1)
```

```
history = model.fit(X_train, y_train, batch_size=BATCH_SIZE, epochs=EPOCHS,
                    validation_data=(X_test, y_test), callbacks=[ReduceLROnPlateau, stop])
```

Overfitting analysis: Bidirectional LSTM





Streamlit Application

Please scan the above QR code to access the webpage

Streamlit: Steps

01

Save the word tokenizer as a pickle file, and LSTM model as a .h5 file

02

Code Streamlit python app for webpage

03

Upload app and model files to GitHub repository

04

Create Streamlit webpage

05

Deploy the webpage

Challenges

- **Unable to train entire dataset:**

  Computational Limitations; Only used a 20% sample in the model deployed on Streamlit

- **Faced errors during model deployment on Streamlit**

  Conflicting files and misunderstood file paths created some confusion in deployment; we learned to use GitHub for version control.

- **Difficulties in setting model parameters**

  LSTMs are heavy computationally, but performance of the model can be influenced negatively without enough data or correct parameters.

Further Development

01

Train model on the entire dataset for improved accuracy

02

Testing the prediction app with additional emails and enhance the model

03

Test if LSTM model can be optimized more

04

Conduct hyperparamter optimisation for LSTM

05

Use additional emails from web app to improve model and prevent data drifting

Thank You!

Do you have any questions?

Send it to us! We hope you learned something new.