Machine Learning Report

Qiaotong Huang & Jay Shukla
College of Engineering
Northeastern University
Toronto, ON
huang.qiaot@northeastern.edu
Shukla.j@northeastern.edu

1 Definition of Problem(Jay)

Overview: The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according to ham (legitimate) or spam.

The dataset includes a collection of SMS messages in English, labeled as 'spam' or 'ham', and is often used in text classification and natural language processing (NLP) tasks to train models that can identify and filter out unwanted or spam messages from legitimate ones.

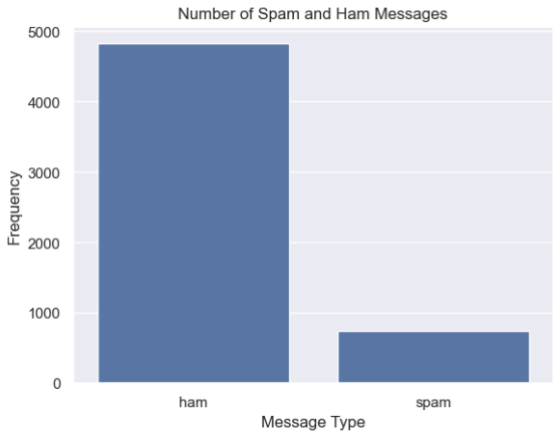
For a project using this dataset, the primary objective would be to construct a predictive model capable of distinguishing between spam and non-spam messages based on their text content. This involves preprocessing the text data, feature extraction, and then training a classifier using machine learning algorithms.

2 Data Collection(Huang)

I selected the SMS Spam dataset through Kaggle. It contains one set of SMS messages in English of 5,574 messages, tagged according to ham (legitimate) or spam.

Table 1: The basic feature of the dataset

	Data Set Characteristics	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attributes
SMS Spam	Multivariate	Real	Classification	5574	2



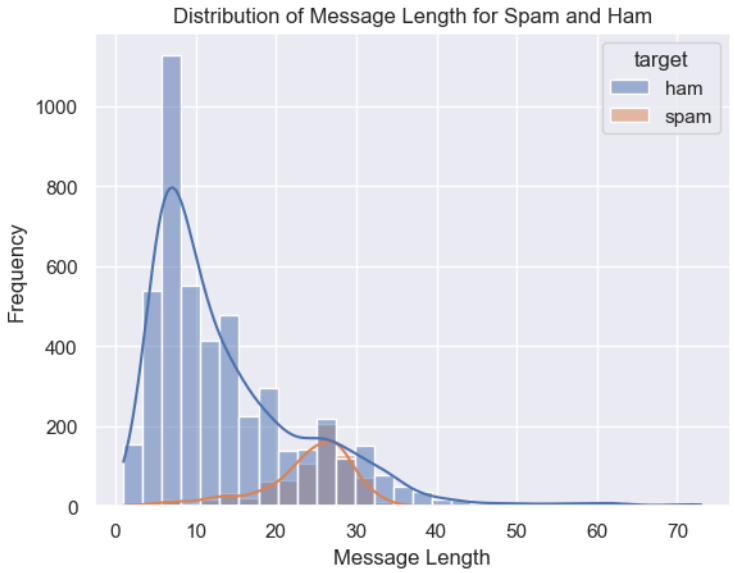
From this frequency plot we can see that this is an imbalanced data set. Ham's data is close to 5,000, and spam's data is less than 1,000.

```
df['message_len'] = df['message'].apply(lambda x: len(x.split(' ')))
df.head()
```

✓ 0.0s

	target	message	message_len
0	ham	Go until jurong point, crazy.. Available only ...	20
1	ham	Ok lar... Joking wif u oni...	6
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	28
3	ham	U dun say so early hor... U c already then say...	11
4	ham	Nah I don't think he goes to usf, he lives aro...	13

First, our data has only two attributes, named message and target. Then I added the length of each message as a third attribute. Because we want to analyze the relationship between the length of the text and the target.



It can be found that the spam message length is mainly concentrated in the 20-30 range. But Ham's information range is wider and the number within 20 accounts for the majority.

3 Data Preprocessing(Huang)

First, we clean up the corpus, mainly using regular matching. The purpose is to make text lowercase, remove text within square brackets, remove links, remove punctuation, and remove words containing numbers. The cleaned text is in the message_clean attribute.

	target	message	message_len	message_clean
0	ham	Go until jurong point, crazy.. Available only ...	20	go until jurong point crazy available only in ...
1	ham	Ok lar... Joking wif u oni...	6	ok lar joking wif u oni
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	28	free entry in a wkly comp to win fa cup final...
3	ham	U dun say so early hor... U c already then say...	11	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...	13	nah i dont think he goes to usf he lives aroun...

Then clean up the stop words. The stopwords library in nltk is used here. The stems are then extracted via the Snowball Stemmer library. We also encode the target with ham as 0 and spam as 1.

	target	message	message_len	message_clean	target_encoded
0	ham	Go until jurong point, crazy.. Available only ...	20	go jurong point crazi avail bugi n great world...	0
1	ham	Ok lar... Joking wif u oni...	6	ok lar joke wif oni	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	28	free entri wkli comp win fa cup final tkts m...	1
3	ham	U dun say so early hor... U c already then say...	11	dun say earli hor alreadi say	0
4	ham	Nah I don't think he goes to usf, he lives aro...	13	nah dont think goe usf live around though	0

We need to transform our messages, which are lists of tokens or lemmas, into numerical

vectors that can be processed by SciKitLearn's algorithms. So that we can prepare for Random Forests.

To achieve this, we'll use the bag-of-words model in three steps:

1. Term Frequency (TF): Count the occurrences of each word in a message.

2. Inverse Document Frequency (IDF): Adjust these counts by giving less weight to common words.

3. L2 Normalization: Scale the vectors so their length is 1, making them independent of the text's original length.

Finally, after such processing, our data has been converted into vector data that can be trained with the model, and this data retains information to the greatest extent possible.

4 Model Selection and Development(Huang)

In the project, we chose Random Forests and RNN (LSMT).

```
pipe = Pipeline([('bow', CountVectorizer()),
                 ('tfid', TfidfTransformer()),
                 ('model', RandomForestClassifier())])
```

Why Random Forests? Random forests are an effective alternative, especially when the data set is smaller or has fewer features. Random forests splits through self-service sampling and randomly selected features, and has a certain ability to resist over-fitting. Random forests can provide a ranking of feature importance, thereby providing model interpretability.

```
def RNN():
    inputs = Input(name='inputs',shape=[max_len])
    layer = Embedding(max_words,100,input_length=max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256,name='FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1,name='out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs=inputs,outputs=layer)
    return model
```

In the RNN model, we have an input layer and an embedding layer. We use the Embedding function of Keras to define the embedding layer and take the input layer as input.

An LSTM layer (Long Short-Term Memory) is used to process the time dependence of the input sequence. The dimension is 64.

A Dense Layer used to linearly transform the output of the LSTM layer. 256 specifies the output dimension of the fully connected layer.

A ReLU activation function layer to introduce nonlinearity.

A Dropout layer to randomly discard some neurons during training to prevent overfitting.

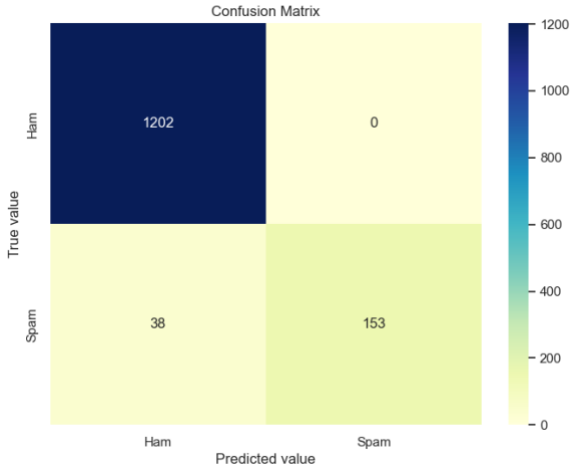
An output layer is used to output the prediction results of the model. 1 specifies the dimension of the output layer to be 1 because this is a binary classification task.

A Sigmoid activation function layer is used to map the output value to the [0, 1] interval, representing the predicted probability.

In this way we define a simple RNN model using LSTM.

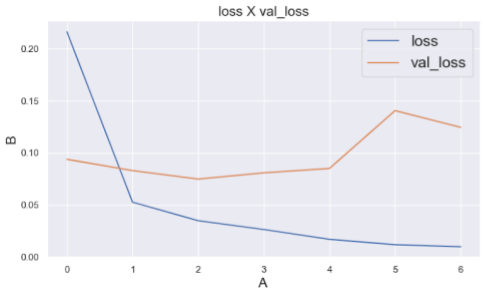
5 Model Evaluation(Jay)

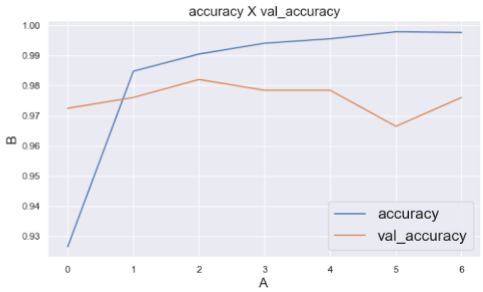
For this project we have used Random Forests Classifier. So, first of all we calculated accuracy that is 0.9727207465900939. Given below is the image of confusion matrix.



Evaluation of the model:

Test Loss:0.12
Test Accuracy:0.98
Test F1 Score:0.87
Test Precision:0.89





6 Conclusion(Jay)

Model Performance: How well the predictive model distinguishes between spam and ham messages, usually quantified by metrics such as accuracy, precision, recall, and F1 score.

Feature Importance: Insights into which words or phrases are most indicative of spam, highlighting the effectiveness of different preprocessing and feature extraction techniques like tokenization, stemming, and vectorization.

Algorithm Comparison: Evaluation of different machine learning algorithms (e.g., Naive Bayes, SVM, decision trees, neural networks) on their ability to classify messages accurately, including discussions on model complexity, overfitting, or underfitting issues.

Challenges and Limitations: Identification of any challenges faced during the analysis, such as handling imbalanced data, dealing with abbreviations and slang in SMS, and differentiating between spam and ham messages with similar wording.

Practical Implications: The real-world applicability of the spam detection model in filtering unwanted messages, improving user experience, and maintaining communication security.

Future Work: Suggestions for further research or improvements, such as incorporating more contextual information, using more advanced NLP techniques like deep learning, or expanding the dataset to include messages from various sources for a more robust model.

The conclusion will synthesize these elements to reflect on the dataset's utility in building effective spam detection systems and the potential for future enhancements in the field of text classification and NLP.

References

[1] Kaggle.com “SMS Spam Collection Dataset” https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset