**1 Motivation**

(Slide 1) Greetings TA and Profs, I am xxx, and my teammates are xxx, and xxx. Today we are going to share with you our SC1015 project. Our mini project is based on the fake news dataset sourced from kaggle, contributed by Aleksei Golovin.

(Slide 2) I present to you our table of contents. Firstly, I will explain our motivation behind choosing this project. Secondly, we will share our exploratory data analysis and visualization. Thirdly, we will elaborate on our ML models and finally, our conclusion.

(Slide 3) In today’s world, we are constantly bombarded with news from various sources. While some of this news is factual, there is also an alarming amount of fake news that is being spread. Fake news can cause significant harm by spreading false information, creating panic and distrust, and eventually leading to serious consequences. Especially in this current day and age, the issue is further compounded by the fact that fake news can spread extremely rapidly on social media platforms, reaching a vast audience in a matter of minutes.

(Slide 4) Undoubtedly, such fake news will raise a host of issues, not limited to monterey concerns, widespread panic and suspicion and loss of trust in journalism.

(Slide 5) Firstly, monetary concerns. For example, a hoax in 2017 declaring the founder of Ethereum had died caused the company to lose over $4 billion in market value and shares. This may cause many stakeholders and investors to lose their money and withdraw funds from the company. This may start a chain reaction, affecting many other companies too.

(Slide 6) Secondly, fake news causes widespread fear and panic. For example in the US Presidential Election, 64% of adults believe that the fake news proliferating will cause a lot of confusion and raise suspicions among the public. Furthermore, 23% of adults admitted to sharing or even fabricating fake news themselves, whether intentional or unintentionally.

(Slide 7) Thirdly, fake news may cause the public to lose trust in mainstream journalism as it is harder to discern fake from real news. This will erode trust between the media and the masses, and truthful news may go unreported, causing journalism to lose its integrity and goal of providing the truth.

(Slide 8) To address this issue, we decided to undertake a project focused on creating a machine learning algorithm that can effectively detect fake news in order to aid administrators/moderators in the elimination of fake news from their media platforms. This project aims to leverage the power of artificial intelligence to sift through news articles, identify patterns, and distinguish between genuine news and fake news.

Which leads us to our problem definition: To utilize machine learning algorithms that can accurately discern fake news from real news and analyze the factors that differentiate them.

**2 Exploratory Data Analysis and Visualization**

(Slide 9) First we began cleaning the data for our purposes.

From a precursory evaluation of the dataset, we identified redundant columns which we will not require for our analysis such as the ‘news\_url’ and ‘source\_domain’ column and removed them from our data set.

Next, we removed any duplicate messages to prevent them from affecting our findings.

After which, we used stopwords to remove words such as ‘to’ and ‘a’, common words which will appear in any English sentence in order to filter out the key words most commonly used in fake news. To further filter it, any common words not included in stopwords such as “&” was removed as well.

(Slide 10) After cleaning up the data, we can do our exploratory data analysis and visualization.

(Slide 11) The first thing we did was to compare the word count between real and fake news, in an attempt to find any patterns in distinguishing between real news and fake news. From the histogram, it can be seen that there is the same average number of words and similar dispersion in word count.

(Slide 12) Next, we plotted the 10 most common words, the 10 most common words in fake messages and the 10 most common words in real messages onto bar graphs to identify clearly the most important words to be used in our ML algorithm.

(Slide 13) More clearly, the differences can be more obviously seen when placed on a word cloud.

**3 Machine Learning**

(Slide 14) The machine learning models that we are planning to use is: Naive Bayes and Random Forest Regression

(Slide 15) In machine learning, naive Bayes classifiers are a family of simple “probabilistic classifiers” based on applying Bayes’ theorem with strong (or naive) independence assumptions between the features. This means that it classifies the data into different classes naively. Its origin is from the Bayes’ theorem, which states , but now transformed to . The posterior is calculated by multiplying the prior probability of the class with the conditional probabilities of each feature given the class, overall divided by the probability of observing that picture. As a result, we can see the probability of detecting any feature given the class. This model is closely related to linear regression which was covered in the lectures.

(Slide 16) This Is a code snippet from the Naive-Bayes Model to find the confusion matrix. Through this, we can see the respective True and False Positive rate and True and False Negative Rate.

(Slide 17) From the same model, we can see that the accuracy is 0.81312, which is a ratio of number of correctly classified instances to the total number of instances in the dataset, hence being a good result. In addition, the F1 score is 0.88883. The formula for F1 score is the harmonic mean of precision and recall, where precision measures the percentage of true positives out of all positive predictions, while recall measures the percentage of true positives out of all actual positives. Hence, this high F1 score means that the model is fairly good at correctly classifying instances.

(Slide 18) Random forest regression involves creating a multitude of decision trees, where each tree is trained on a different subset of the data and a different set of input variables. In other words, it involves creating an ensemble of decision trees where each decision tree is trained on a random subset of the data, which helps to reduce overfitting and improve the generalization performance of the model.

During the training phase, each decision tree is built by selecting the best split at each node of the tree based on a randomly chosen subset of input features. The final prediction of the random forest is then obtained by taking the average of the predictions of all the decision trees.

(Slide 19) This is a code snippet from the Random Forest Classifier to find the confusion matrix. Through this, we can see the respective True and False Positive rate and True and False Negative Rate.

(Slide 20) From the same model, we can see that the accuracy is 0.82601, which is a ratio of number of correctly classified instances to the total number of instances in the dataset, hence being a good result. In addition, the F1 score is 0.89147 meaning that the model is also fairly good at correctly classifying instances.

(Slide 21) K-fold cross-validation is a technique used to evaluate the performance of a machine learning model. It involves dividing the original data into K equally-sized subsamples or "folds".

One of the folds is used as the test set, and the remaining K-1 folds are used as the training set. The model is then trained on the training set and evaluated on the test set. This process is repeated K times, with each fold serving as the test set once.

The final performance of the model is then obtained by averaging the performance over the K folds. This technique allows for a more reliable estimate of the model's performance, as it uses all the data for both training and testing purposes.

From the table, the random forest classifier is better than the naive-bayes model in terms of K-fold accuracy and accuracy. Hence, it is a better model compared to naive-bayes. However, the accuracy is not 1 so we can explore different models to achieve even more accurate results.

(Slide 22) ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model. It is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at different classification thresholds. The ROC curve is a useful tool for evaluating and comparing the performance of different classification models, as it provides a graphical representation of the tradeoff between TPR and FPR at different classification thresholds. A model with a higher AUC (Area Under the Curve) value is generally considered to be better at distinguishing between the two classes.

From the graph, the random forest classifier occupies a larger area compared to naive-bayes model. The ROC curve helps to visualize how well it can distinguish between the two classes by plotting the TPR against the FPR. The ratio of F1-score to accuracy is also higher for random forest classifiers that’s why it is a preferred mode in this case. However, naive bayes model may work better on different types of dataset.

**4 Conclusion**

(Slide 23) In conclusion, we found out that it is possible to predict and discern fake news from real news using ML models. Furthermore, we can do to a large extent as seen from the high accuracy and F1 score for both models. While the number may not be perfectly 1, they are still very acceptable and accurate.

However, there are some possible improvements. The dataset has limited factors. It would be better if we have a more comprehensive and detailed dataset with more factors such as not limited to just words in the title and number of upvotes. For example, the number of comments, the number of reshares and demographic statistics. More factors can lead to an accurate dataset and reduce biases due to certain factors having more correlation with the fake and real news.

(Slide 24) We can also explore different models such as Support Vector Machines (SVMs) and Neural networks, known for their powerful ability to draw out relationships and predict to a larger extent as they work well on non-linear relationships between predictors and responses.

(Slide 25) Additionally, we can also implement a feedback system to improve the model’s learning. For example, people can rate ChatGPT answers to give their satisfaction based on the answers given. Users can provide feedback on the responses generated by the model. When a user submits feedback, it is sent to a database where it is stored along with the corresponding response from the model. The feedback is then used to improve the model's performance. The data is analyzed to identify patterns and areas where the model may need to be improved. The feedback is then used to retrain the model, so it can generate better responses in the future. This helps the model to learn from its mistakes and become better at generating responses over time.