Predicting Fake News Using Bayes Classification

Chandler Campbell

Student ID: 011247597

Table of Contents

[Prompt A 4](#_Toc152943774)

[Letter of Transmittal 4](#_Toc152943775)

[Project Recommendation 6](#_Toc152943776)

[Application Benefits 6](#_Toc152943777)

[Application Description 7](#_Toc152943778)

[Data Description 8](#_Toc152943779)

[Objective and Hypothesis 8](#_Toc152943780)

[Methodology 9](#_Toc152943781)

[Funding Requirements 9](#_Toc152943782)

[Stakeholders Impact 10](#_Toc152943783)

[Data Precautions 10](#_Toc152943784)

[Developer Expertise 11](#_Toc152943785)

[Prompt B 12](#_Toc152943786)

[Project Proposal 12](#_Toc152943787)

[PROBLEM STATEMENT 12](#_Toc152943788)

[CUSTOMER SUMMARY 12](#_Toc152943789)

[EXISTING SYSTEM ANALYSIS 13](#_Toc152943790)

[DATA 13](#_Toc152943791)

[PROJECT METHODOLOGY 13](#_Toc152943792)

[PROJECT OUTCOMES 14](#_Toc152943793)

[IMPLEMENTATION PLAN 14](#_Toc152943794)

[EVALUATION PLAN 15](#_Toc152943795)

[RESOURCES AND COSTS 15](#_Toc152943796)

[TIMELINE AND MILESTONES 16](#_Toc152943797)

[Prompt C 17](#_Toc152943798)

[Application Files 17](#_Toc152943799)

[Prompt D 18](#_Toc152943800)

[Post-implementation Report 18](#_Toc152943801)

[Project purpose 18](#_Toc152943802)

[Datasets 18](#_Toc152943803)

[Data product code 19](#_Toc152943804)

[Hypothesis verification 20](#_Toc152943805)

[Effective visualizations and reporting 20](#_Toc152943806)

[Accuracy analysis 21](#_Toc152943807)

[Application testing 22](#_Toc152943808)

[A screenshot of a computer program

Description automatically generated 22](#_Toc152943809)

[Appendices 23](#_Toc152943810)

[Installation Guide 23](#_Toc152943811)

[User Guide 24](#_Toc152943812)

[Summation of Learning Experience 24](#_Toc152943813)

[References 25](#_Toc152943814)

# Prompt A

## Letter of Transmittal

December 1, 2023

John Smith

FriendFusion

123 Main St, Los Angeles, CA 12345

Dear John Smith,

I am writing to you because I want to suggest a new solution to the growing concern in the area of information on our platform. There is an abundance of fake news. In a time where the integrity of news is very important for decision-making, it is necessary that our organization takes steps to verify the accuracy of the information shared on our social media site.

The rise of social media has made information more accessible but it also led to the spread of misinformation. Fake news articles that are sensationalized or entirely fake content are a significant threat to the reputation of our company. They can influence public opinion, damage our users, and even impact global events. As an organization that values truth and accuracy, it is essential that we address this challenge at full force. During the last presidential election, we faced widespread criticism for failing to stop the spread of misinformation.

To fix this issue, I propose the development and implementation of a supervised machine learning model based on Naive Bayes classification. This model will analyze news titles and text to predict whether they are fake or not. By training the model on a dataset of verified real and fake news articles from Kaggle, it will learn to differentiate between the two with a high accuracy.

Implementing this solution will improve FriendFusion’s credibility and reliability. It will make sure that the news users share or like is accurate or has a warning label, facilitating trust among our stakeholders. Additionally, by leading the way in using machine learning, we demonstrate our commitment to innovation and integrity in the information war.

The implementation of this project will involve the following steps:

1. Collection and cleaning of a dataset from Kaggle comprising both real and fake news articles.
2. Development of the Bayes classification model, including training, testing, and validation.
3. Integration of the model into our existing platform.
4. Continuous monitoring and model refinement to maintain accuracy and relevance.

In conclusion, the proposed Bayes classification model offers a practical and forward-looking solution to the challenge of fake news on our site. The implementation will stabilize our position as the number one social media company in the word, and one of the biggest innovators of the tech industry. As a machine learning engineer with two decades of experience, I believe this project is not only feasible but will not be cost prohibitive. We should be able to get a working model within six months.

Thank you for considering this proposal. I am thrilled to discuss this project further and explore its potential impact on FriendFusion.

A black background with a black square

Description automatically generated with medium confidenceSincerely,

Chandler Campbell, Machine Learning Engineer

## Project Recommendation

FriendFusion is facing a difficult challenge with the surge of inaccurate news on their social media platform. This problem poses a big risk to the platform's reputation. The abundance of such content could lead to users making decisions based on false information, damaging the company’s image and potentially leading to legal and ethical dilemmas. A proactive strategy is essential to identify and stop the spread of this content.

At the heart of FriendFusion’s values is a commitment to maintaining a social media environment that is both credible and reliable. The current challenge is in the platform's limited ability to effectively differentiate between legitimate and fake news. Currently, it takes employees hours to independently verify news articles. Considering the middle management team, which comprises IT professionals with different degrees of computer science knowledge, there is a need for a solution that balances advanced machine learning with ease of use. Their primary concern is upholding the platform's integrity, making sure there is high user confidence and high standards of information accuracy.

The solution proposed includes the development of a Bayes Classification Application using Python. This tool will use the machine learning algorithm to examine news titles and content, predicting their chance of being fake. It will be integrated into the platform's back-end, adding capability to identify false news. In addition, a User Guide will be provided to the IT team, outlining how the application functions and its integration into existing systems. To further support the IT team, a training workshop will be created, making sure they are well-equipped to manage the system and effectively tackle any issues related to inaccurate news.

The introduction of this application has several key advantages. First, it will substantially boost the platform's trustworthiness by reducing the circulation of false news, which in turn will improve user trust, leading to greater engagement and loyalty. Second, it will also lower the risk of legal and ethical complications associated with the distribution of false information. Third, the application will streamline the news verification process, reducing the dependence on manual moderation and enhancing overall efficiency.

Therefore, the Bayes classification application we plan to develop is a innovative solution addressing the critical issue of false news on the social media platform. It meets the client’s requirements for a credible and user-friendly platform and offers a practical approach to a major challenge in the current information arena.

### Application Benefits

The introduction of the Bayes Classification-Based Application to FriendFusion’s social media platform is poised to deliver substantial benefits, mostly in the field of data-driven decision-making. This application stands as more than just a tool for filtering content. It as stands as a great example of informed and strategic decision-making within the company.

At its core, the application provides a more accurate and reliable platform by obtaining the ability to label false news. This improvement in quality is important for the company, as decisions based on inaccurate information can lead to massive issues, harming the company's reputation and user trust. By making sure that the that the platform labels fake news, the company can make informed choices regarding content management, user engagement strategies, and policy development.

The application's ability to classify news accurately creates opportunities for more data analytics. By analyzing trends in fake news, the company can anticipate and prepare for potential misinformation campaigns. This capability allows the company to stay ahead of trends and adapt its content moderation strategies as necessary.

In a time where digital platforms are increasingly criticized for their content management policies, the application helps risk management. By identifying and addressing the spread of fake news, the company can avoid legal and ethical challenges. This stance on content integrity makes sure we are compliant with changing regulations regarding online content, protecting the company against regulatory penalties.

From a user perspective, the application contributes to a more trustworthy platform. Users who trust the platform's content are more likely to engage with it, share it, and recommend it to others. This increased trust and engagement can translate into higher user retention rates, more time spent on the platform, and increased revenue from advertising.

By largely automating the process of news verification, the application also allows for more efficient resource allocation. Manual content moderation is not only time-consuming but also prone to human error. The application lowers the burden on human moderators, allowing them to focus on more difficult content issues that require human judgment. This optimization of resources leads to cost savings and increased efficiency.

In conclusion, the Bayes Classification Application offers FriendFusion an advantage over other social media companies. It improves decision-making, provides predictions, manages risks, increases user trust and engagement, and optimizes resource allocation. These benefits contribute to an excellent social media platform, able to handle the challenges of our time.

### Application Description

The application is an innovative tool to be used for FriendFusion’s social media platform. It uses the Naive Bayes classification algorithm to accurately analyze and classify news content as either real or fake. The core functionality of the application revolves around processing and evaluating news titles and text, making determinations based on learned patterns from a dataset composed of verified real and fake news articles.

The application will operate quietly in the background of the social media platform, scanning and assessing news content as it is posted. When it identifies a piece of news as potentially fake, it flags it for further review and labels it for users as problematic. Overall, this application aims to improve the integrity and trustworthiness of the content on the social media platform, making sure that users have access to reliable and accurate news.

### Data Description

The project will use a dataset from Kaggle that features a mix of around 21,000 real news articles and 18,000 fake ones. The dataset is organized into columns that include the article’s title, the main text, the type of news, and the date it was published. This robust collection of data is ideal for the project needs because it provides a diverse range of information that the Bayes classification algorithm can learn from to distinguish between real and fake news.

Throughout the development of the application, the dataset will first be reviewed for quality and consistency. I will clean it up, removing any duplicates and filling in gaps where information might be missing. During the development phase, I will simplify the content of the articles into a format the algorithm can learn from, picking out key features that can help identify the authenticity of a news piece. The data will then be split, with a portion used to teach the algorithm and the rest held back to test how well it has learned.

When the application is up and running, it will apply the same steps to new articles, ensuring that the model remains accurate in its predictions. FriendFusion should monitor the model and feed it new data regularly so it does not become outdated. This is the kind of maintenance that makes sure the application stays sharp and effective.

The dataset's size and variety make it a good fit for the application’s goal. It is important because it mirrors the real-world variety and volume of news that the application will eventually handle. If the company runs into any unusual data, such as an article that’s too short or long, or ones that seem to be out of place—the team will take a closer look to decide if they should be included. This way, we make sure that the model learns from the best possible examples, which will help FriendFusion maintain the accuracy and trustworthiness of the content on their social media platform.

### Objective and Hypothesis

The objective of the upcoming project is to create a tool that quickly and accurately spots fake news on FriendFusion's social media platform. The tool should be sharp enough to catch and label misinformation without slowing down the flow of real news. This tool, built around a Bayes classification algorithm, should blend into the platform's current setup, supporting a seamless user experience.

First, the tool needs to be accurate. It should spot the fake news without excessively flagging the real news. Second, it has to be fast. The company does not want to interrupt how users interact with the site. Third, it should be easy to add to the platform. The company is not looking to rebuild the system, just to improve it. Fourth, as the platform grows, the tool should be able to keep up with demand. Fifth, the company should make sure that the tool keeps learning and adapting.

I have three hypotheses. First, I hypothesize that the model will have an accuracy score of greater than 90%. Second, I hypothesize that the model will have a recall score of greater than 90%. Third, I hypothesize that the model will have a precision score of greater than 90%.

### Methodology

The project will use the Waterfall methodology. The Waterfall methodology is a sequential design process where progress flows in one direction downwards like a waterfall through the phases of conception, initiation, analysis, design, construction, testing, deployment, and maintenance. This traditional approach is structured and easy to understand, with each phase having specific deliverables and a review process. In the context of FriendFusion’s project to develop a fake news detection tool, Waterfall is advantageous due to its straightforward, methodical nature.

For this project, the Waterfall methodology makes sure that the company has a clear roadmap from the start. It begins with a thorough requirement analysis, which is important for understanding what the Bayes classification algorithm needs to achieve. Because the requirements are well understood and unlikely to change, I can proceed to design the system confidently. This design phase allows me to map out the entire algorithm, considering the dataset's structure.

Once the design is complete, I can build the system without the need for significant backtracking or revisions. This is essential for maintaining a clear focus and for efficient time management. The testing phase follows, providing an opportunity to work out any issues before the application is finished. Finally, the deployment and maintenance phases are well-defined in Waterfall, making sure that the tool is deployed effectively and supported over time.

Waterfall is great for this project because it allows for in-depth upfront research and planning. As this project included a proposal, this makes the most sense. With clear stages, I can ensure quality and accuracy, which is crucial for the application’s effectiveness and the platform's credibility.

### Funding Requirements

For the fake news detection project, there is an estimated total budget of approximately $240,000. This figure is inclusive of all phases, from initial research to long-term maintenance. The bulk of the budget, around $150,000, would be allocated to the development team's salaries over a six-month period. The necessary tech infrastructure, which includes servers and processing power, could cost about $20,000, with ongoing cloud services adding to $5,000 for half a year. The critical testing phase, making sure the tool is accurate and reliable, might require an investment of $15,000.

Deployment costs, accounting for integration, are projected at $10,000. After going live, the tool will need continuous updates and maintenance to keep up with the evolving nature of fake news. This could cost $20,000 annually. The company should also have a contingency fund of $20,000 to handle unexpected expenses. This budget plan builds a solid financial foundation for the project's success, protecting against potential shortfalls and making sure that the tool developed is effective for FriendFusion.

### Stakeholders Impact

The introduction of a fake news detection tool into FriendFusion's social media platform is expected to have a broad impact on various stakeholders. For the users of the platform the tool creates a more reliable and trustworthy environment. They can engage with content knowing that there's a system in place to protect them from misinformation, which will enhance user experience and satisfaction. For the company's employees involved in content moderation, the tool would reduce workload and stress by automating the detection process, allowing them to focus on more complex tasks that require human evaluation.

Shareholders and investors will benefit from the improved public perception and credibility of the platform, which could lead to an increase in user base and advertising revenue. Advertisers and partners will appreciate the association with a platform taking proactive steps against fake news, which aligns with corporate responsibility goals and can strengthen the brand's integrity. Moreover, content creators and news outlets will find the platform a better space, as their authentic content will not be lost in a sea of misinformation. This could attract higher-quality content contributions and collaborations.

On the regulatory side, implementing the tool demonstrates the company's commitment to addressing misinformation concerns, potentially loosening regulatory pressures. In essence, the deployment of this tool is expected to have a positive reaction for all of the company’s stakeholders.

### Data Precautions

Handling Kaggle's dataset for our project, the team strictly follow ethical guidelines and legal requirements. Privacy will be the top priority, ensuring no personal information is misused. The team will anonymize any sensitive data and make sure our use aligns with data protection laws. The team will only access the data they need, and stakeholders will be informed about how the team is using the information without sharing private details. Security measures, such as encryption and controlled access, will be in place to protect the data at all times.

### Developer Expertise

I have been a machine learning engineer for over twenty years. I have seen artificial intelligence grow from an emerging field into an essential element of social media. Presently, I am part of FriendFusion, a company at the nexus of social media innovation. My expertise is instrumental in creating projects that use data to improve user interaction engagement on the platform.

In my role at FriendFusion, I focus on applying machine learning to ensure the accuracy and authenticity of the content our users encounter. I guide a team that develops sophisticated algorithms capable of figuring out relevant patterns in large datasets. Our aim is to maintain the integrity of information, fostering an environment where truth and reliability are prioritized. The latest project, creating a tool to flag and filter fake news, is a challenge that captures the core of my professional career. By developing a Bayes classification model that learns from the Kaggle dataset, I am establishing a benchmark for content verification in the industry.

# Prompt B

## Project Proposal

### PROBLEM STATEMENT

In the social media arena, the rapid exchange of information is a defining characteristic. Unfortunately, the abundance of fake news has emerged as a major challenge. It undermines the credibility of platforms, distorts public discourse, and manipulates the decision-making process of users. At FriendFusion, the pressing issue we face is the need for a reliable method to quickly and accurately determine and label fake news content.

This challenge calls for a solution that addresses the problem of misinformation and does so in a way that is scalable, efficient, and maintains the user experience without introducing significant delays or restrictions. The development of a machine learning model capable of meeting these demands is a social responsibility that FriendFusion is committed to undertaking.

### CUSTOMER SUMMARY

The customers at FriendFusion are a diverse and dynamic group, ranging from young adults seeking to stay informed and connected with their peers, to professionals who rely on the platform for the latest updates in their industries, and to seniors who enjoy the platform's ability to keep them in touch with family and friends. They come to FriendFusion for a social media experience that is not only engaging and interactive but also trustworthy and reliable. They value the authenticity of the content they read and share, as it forms the basis of their knowledge, opinions, and online interactions.

This diverse customer base has common desire for the information they encounter to be accurate. The rise of fake news has lowered their trust in the content they come across, creating a gap that our product is positioned to fill. The Bayes classification-based fake news detection tool is designed to identify and flag fake content, protecting the integrity of the information on the platform.

For our customers, this tool represents a shield against the misinformation that can lead to confusion and bad decisions. By providing labeling fake content, we are fulfilling their need for reliable information and also enhancing their overall experience on the platform. This commitment to credibility will lead to deeper trust in FriendFusion, encouraging greater engagement and longer-term loyalty among the users. The introduction of this tool is a testament to our dedication to serving our customers.

### EXISTING SYSTEM ANALYSIS

At FriendFusion, the current approach to mitigating misinformation is primarily manual, involving content moderators who inefficiently review and assess the accuracy of news widely shared on the platform. It is inherently slow, labor-intensive, and struggles to keep up with the volume of content generated daily. Moreover, this manual process can be inconsistent, subject to the biases and varying expertise of individual moderators. As a result, there is a special need for an automated system that can help the human moderators, enhance accuracy, and make sure that the moderation of news keeps pace with the spread of it.

### DATA

The dataset is from Kaggle, comprising approximately 21,000 real news articles and approximately 18,000 fake news articles, is essential for the development of our fake news detection tool at FriendFusion. Possessing features such as news title, text, category, and posting date, it provides a solid foundation for training and the Bayes classification algorithm. In the initial stages of the project, this dataset will be pivotal in helping identify and understand the distinct characteristics and patterns that differentiate genuine news from fabricated stories. This will be explored through word clouds of real and fake news articles.

### PROJECT METHODOLOGY

The project's implementation plan, centered on developing a Bayes classification-based machine learning tool for fake news detection, adheres to the structured phases of the Waterfall methodology. Initially, in the requirements gathering phase, the team will define the specific needs and objectives for the tool, making sure there is a clear understanding of the features it must possess to effectively distinguish between real and fake news. This phase will be informed by an analysis of the Kaggle dataset and insights from experts at FriendFusion.

Subsequently, the project moves into the design phase, where I will architect the Bayes classification model, outlining the algorithm's structure and data processing methods. This phase involves detailed planning of how the model will process and analyze the news titles and text from the dataset to make accurate classifications.

The development phase follows, where the actual coding of the model takes place. Here, the Kaggle dataset serves as the primary source for training and testing the algorithm. This stage will focus on the iterative development of the model, with continuous integration and testing against a subset of the data to ensure accuracy and performance.

After development, the model enters the testing phase, where it undergoes validation against a separate set of data. This phase is crucial for ensuring the model's reliability and effectiveness in a real-world environment. It is also the stage where any adjustments or refinements are made based on testing outcomes.

Once testing confirms the model's efficacy, we proceed to the deployment phase, where the tool is integrated into FriendFusion's platform. Lastly, the maintenance phase will see ongoing monitoring and updating of the model to adapt to new patterns in fake news, ensuring the tool remains effective over time. Throughout each of these phases, the Waterfall methodology provides a systematic approach, going in order one phase at a time.

### PROJECT OUTCOMES

The successful completion of our project to develop a Bayes classifier for fake news detection at FriendFusion will result in several deliverables, each important in the project's overall effectiveness. The main deliverable is the fully developed Bayes classifier model, trained using the Kaggle dataset to accurately identify fake news. This model will incorporate advanced language processing techniques, enabling it to analyze news titles and text effectively.

With this model will be detailed training and testing reports, providing insights into the model's performance metrics like accuracy, recall, precision, and the F1-score.

A crucial part of these outcomes is the integration plan, which will outline the technical steps and strategies for incorporating the classifier into FriendFusion's existing platform. To facilitate ease of use and understanding, a user guide will be supplied. These documents will serve as a resource for platform moderators and administrators, guiding them on how to use and interpret the tool's outputs.

Additionally, a maintenance and update strategy will be provided, detailing how the classifier will be kept updated in the rapidly evolving field of fake news. This strategy will include procedures for updating and retraining the model with new data. Lastly, the project will conclude with an evaluation report, summarizing the effectiveness of the Bayes classifier. These deliverables mark the successful implementation of practical tool for enhancing the credibility and reliability of content on the FriendFusion platform.

### IMPLEMENTATION PLAN

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected start date | Anticipated end date |
| Requirements Gathering | 2 weeks | December 8, 2023 | December 22, 2023 |
| System Design | 3 weeks | December 23, 2023 | January 12, 2024 |
| Development | 8 weeks | January 14, 2024 | March 10, 2024 |
| Testing | 4 weeks | March 11, 2024 | April 8, 2024 |
| Deployment | 1 week | April 9, 2024 | April 16, 2024 |
| Training and Documentation | 2 weeks | April 17, 2024 | May 1, 2024 |
| Maintenance | 2 weeks | May 2, 2024 | May 16, 2024 |
| Project Evaluation | 1 week | May 17, 2024 | May 24, 2024 |

### EVALUATION PLAN

Our evaluation strategy for the Bayes classifier data product uses precise verification and validation methods to make sure it fulfills the project specifications and our customers' requirements. Throughout the development stages, the team will employ specific checks: stakeholders will review the requirements to confirm their accuracy, the design will be subject to peer assessments to ensure it aligns with the criteria, and the code will be examined through static analysis to guarantee quality. Testing will consist of a series of automated and manual procedures to verify the functionality of the classifier in a range of conditions.

Upon the project's completion, the team will initiate validation methods. The classifier's efficacy will be measured against distinct news articles it has not previously encountered, using metrics such as accuracy, recall, precision, and the F1-score to assess its performance. User acceptance testing will be used to confirm that the classifier meets the operational needs of the users on the FriendFusion platform. The team will also validate the classifier's performance in an operational environment, monitoring its accuracy and scalability live. Additionally, ongoing customer feedback will establish a loop for validation, making sure the classifier adapts to user needs and the evolving nature of news content. This comprehensive approach to verification and validation will make sure that the final product is efficient.

### RESOURCES AND COSTS

Hardware and Software Costs:

* Development workstations (5 units at $2,000 each): $10,000
* PyCharm Professional Edition licenses for the development team (5 licenses at $200 each): $1,000
* Server setup for development and testing, including CPUs, memory, and storage: $10,000
* Cloud-based servers for deployment and live environment testing (6 months at $500/month): $3,000

Labor Costs:

* Data Scientists (2 at $100,000/year, prorated for 6 months): $100,000
* Machine Learning Engineers (2 at $100,000/year, prorated for 6 months): $100,000
* Project Manager (1 at $100,000/year, prorated for 6 months): $50,000
* Quality Assurance Specialists for the testing phase (2 months at $10,000/month): $20,000

Environment Costs:

* Deployment and integration: $10,000
* Hosting costs for the application post-launch (annual): $5,000
* Ongoing maintenance and updates post-launch (annual): $20,000
* Contingency fund to cover unexpected costs: $20,000

### TIMELINE AND MILESTONES

The projected timeline for the fake news detection project using a Bayes classifier begins with Requirements Gathering, to start on December 8, 2023, and run for two weeks until December 22, 2023. This phase will involve our project manager and analysts who will define the project scope and specifications. Immediately following, the System Design phase will take place for three weeks, starting on December 23, 2023, and concluding on January 12, 2024, using our system architects and UX designers to create a blueprint for the classifier system.

Development is scheduled to begin on January 14, 2024, with an eight-week duration, ending on March 10, 2024. Our data scientists and machine learning engineers will be the primary resources during this phase, turning the system design into a working prototype. Next is the Testing phase, which will take four weeks from March 11 to April 8, 2024. Quality assurance teams will be responsible for verifying the classifier's reliability and performance.

Deployment is set for one week from April 9 to April 16, 2024, requiring the collaboration of FriendFusion’s IT and DevOps teams to integrate the classifier into the existing platform. Subsequently, Training and Documentation will last two weeks from April 17 to May 1, 2024, during which the trainers and technical writers will develop comprehensive user guides and conduct training sessions for staff.

Maintenance will be conducted over the following two weeks, from May 2 to May 16, 2024, with system administrators monitoring and refining the system. Project Evaluation will last one week, from May 17 to May 24, 2024. This will involve senior management and the project team reviewing the project's success and gathering insights for future improvements.

# Prompt C

## Application Files

\main.py The Python code that contains the machine learning functionality

\True.csv The dataset containing the real news articles

\Fake.csv The dataset containing the fake news articles

\user\_predictions.log The log file containing the user inputs and predictions

\real\_news\_wordcloud.png Word cloud of most common words in real news

\fake\_news\_wordcloud.png Word cloud of most common words in real news

\word\_count\_bar\_chart.png Bar chart showing average number of words in real vs fake news

# Prompt D

## Post-implementation Report

### Project purpose

The purpose of the capstone project within FriendFusion was to develop a machine learning-based solution using a Bayes classifier to identify fake news on the social media platform. The company's vision was to have a digital environment where information integrity and user trust is important. By integrating this solution, the company lowered the spread of misinformation.

The Bayes classifier model was at the center of this project. It was designed to analyze the textual data of news articles to predict their authenticity with a high degree of accuracy. This meant a significant reduction in the manual labor required to moderate content. The project employed a methodical approach using the Waterfall model of software development. This made sure that each phase from conception and data collection to model training and deployment was executed with attention to detail and a clear focus on the end goal.

A reader with a computer science background would recognize the complexity of training a machine learning model to navigate natural language and the challenges in distinguishing genuine news from counterfeit content. The successful implementation the project reflects a significant achievement in applying machine learning techniques to real-world problems in social media.

### Datasets

The datasets came from Kaggle. There was one for fake and one for real news. It contains approximately 21,000 real news articles and approximately 18,000 fake news articles, structured with columns for the title, text, category, and date of publication. This dataset supported the development of the Bayes classifier.

The datasets were first read using the pandas library. They were then combined into one dataset with an additional 0 or 1 label indicating if the news article was real or fake. The features that were used for prediction were the article title and the article text combined together. The dataset was split into 80% for training and 20% for testing. In order to effectively train the model, the dataset was cleaned by stripping whitespace, removing special characters, and lowering the text.

### Data product code

Initially, the code focused on data loading and preprocessing. Through the load\_and\_preprocess\_data function, it imported real and fake news datasets from CSV files and assigned labels to them for identification (0 for real news and 1 for fake news). The code merged the title and text fields for preprocessing.

The subsequent phase involved creating word clouds for both real and fake news, using the generate\_and\_save\_wordcloud function. This step was under descriptive analysis, providing a visual representation of the most frequent words in each category of news. The code saved these word clouds as image files, giving a visual distinction between the two types of news based on their most common words.

In addition to word clouds, the code also performed an average word count analysis for both news types using the create\_bar\_chart function. This analysis resulted in a bar chart that visually compared the average word count in real and fake news, contributing to the descriptive analysis. The code ensured data security and integrity through the validate\_input function, which cleaned user inputs by removing special characters and making the text lowercase.

The main part of the predictive process involved the feature extraction and model training. The code utilized CountVectorizer to transform the combined text data into a format suitable for machine learning, followed by training a Naive Bayes classifier (MultinomialNB). This trained model became the main part of the predictive methodology, allowing for the prediction of news articles as real or fake.

The code evaluated model performance using metrics such as accuracy, recall, precision, and the F1 score. These metrics were important for understanding the effectiveness and reliability of the predictive model. Furthermore, the code used a graphical user interface (GUI) developed using Tkinter. This GUI allowed users to input news titles and receive predictions on their authenticity. The interface also displayed the calculated model metrics and visualizations such as word clouds and bar charts.

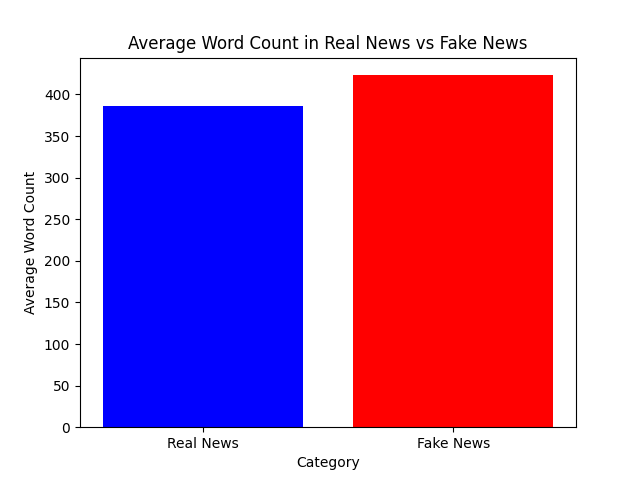
Lastly, the code included a logging mechanism to record user inputs and model predictions. This feature was important for monitoring the tool's performance and maintaining a record of its usage. In conclusion, the code combined data preprocessing, visualization, machine learning, and user interface development to create a tool for analyzing and predicting the authenticity of news articles, using descriptive and predictive methods.

### Hypothesis verification

The three hypotheses are all assessed as correct and accepted. The first hypothesis was that the model accuracy would be above 90%. It ended up being approximately 96%. The second hypothesis was that the model would have a recall score above 90%. It ended up being approximately 95%. The third hypothesis was that the model would have a precision score above 90%. It ended up being approximately 96%.

### Effective visualizations and reporting

The visualizations are in the submitted files and the GUI. They are included below as well.



Fake news articles had a higher average number of words.

A close up of words

Description automatically generated

The figure above is the fake news word cloud.

A close-up of words

Description automatically generated

The figure above is the real news word cloud.

### Accuracy analysis

The model’s metrics suggested that the Bayes classifier was very accurate. The accuracy score was 96%, meaning that of the articles it had not seen before, it was able to predict the authenticity correctly 96% of the time. The recall score was 95%. The precision score was 96%. The F1-score was 96%. Recall is about trying to find all the fake news stories, so in this case, the model found 95% in the testing dataset. Precision is about false positives. So in this case, 96% of news stories labeled as fake were actually fake, but 4% were not. The F1-score combines the precision and recall scores into one number. Therefore, the high scores of the accuracy, recall, precision, and F1 metrics suggest that the model is very good at predicting fake news. However, the data is from 2015 to 2018, so it would probably be less accurate on recent news articles.

### Application testing

# A screenshot of a computer program Description automatically generated

This code here is where most of the testing and optimization took place. I experimented with different ranges, max features, testing sizes, and alpha values. Using a higher number of max features and a bigger range allowed me to get the model to 97% accuracy. However, I had to make a tradeoff decision between speed and accuracy. I determined that the small increase in accuracy was not worth the additional time to train the model. It was taking close to five minutes to execute the code when the range was 1 to 5 and the max features was 15,000. By lowering those parameters, I was able to get the execution time down to around two minutes, which was still long.

Following the Waterfall methodology, I waited to test my code until all of it was written. I inputted various real and fake news articles that were posted in 2023. That is when I noticed that the model was not as good at predicting the authenticity of news articles from this year. I also discovered a trend that news articles that were more shocking were much more likely to be marked as fake even when they were true. Conversely, news articles that I considered boring were much more likely to be marked as real. Below is a screenshot of the GUI that I used to test news articles.

A screenshot of a computer

Description automatically generated

# Appendices

## Installation Guide

1. Download and install PyCharm from [JetBrains.com](http://jetbrains.com/).
2. Download Python from [python.org](http://python.org/).
3. During installation, ensure to check the box "Add Python to PATH".
4. Open PyCharm and navigate to "File" > "Settings" (or "PyCharm" > "Preferences" on macOS).
5. Go to "Project: [Capstone\_Code]" > "Python Interpreter".
6. Click on the gear icon, then select "Add".
7. Choose "System Interpreter" if Python is installed on your system or create a new virtual environment.
8. In the Python Interpreter settings, click the '+' button to add new packages.
9. Install the following packages: pandas, matplotlib, wordcloud, scikit-learn, Pillow, and regex.
10. Tkinter should be included with Python's standard library. If not, ensure your Python installation includes Tkinter.
11. Unzip the project.
12. Open the project in IntelliJ by selecting "Open" and navigating to your project directory.

## User Guide

1. Open up main.py.
2. Click the run icon in the top right corner.
3. Wait for several minutes for the graphical user interface (GUI) to display.
4. Enter a news title, text, or both into the text field.
5. Click the check button.
6. The model’s prediction will be displayed.
7. If desired, enter more news articles.
8. Click the red stop button in the top right corner.

## Summation of Learning Experience

My previous courses at Western Governor’s University strongly prepared me for the Computer Science Capstone. The only exception was the creating the graphical user interface in Python. I had to research that. I believe the most useful course for the capstone was Introduction to Artificial Intelligence. Discrete Mathematics II was surprisingly helpful for understanding the Bayesian classifier and probability. Data Structures and Algorithms II was useful for building my Python skills. Software Engineering as well as Software Design and Quality Assurance were useful for planning my application and testing it. Finally, Version Control was helpful for keeping track of my revisions.

## References

This page is left intentionally blank.