Final Project Report

Name: Luke Roberts **Email: ltroberts@crimson.ua.edu**

Statement:

For this assignment's preparation, the author(s) have utilized ChatGPT, a language model created by OpenAI. Within this assignment, ChatGPT was used for purposes such as code commenting, debugging, lab code repurposing, visual enhancement, sentence structure, clarity in idea expressing, among other things.

1. Introduction

The sinking of the Titanic in 1912 remains a significant historical event, evoking enduring interest to this day. It continues to elicit exploration in data science and machine learning, particularly on platforms like Kaggle. Kaggle hosts a dataset containing details about Titanic passengers, serving as an entry point into the realm of machine learning. This exploration involves predictive analytics on passenger survivability, analyzing the intricate interplay of various factors. The scope extends beyond the model itself to encompass the critical aspect of deployment, aligning historical fascination with contemporary technological application.

2. Dataset Selection & Model Training

2.1 Dataset Description

The Titanic dataset, sourced from Kaggle, underwent rigorous cleaning to address issues like missing values and vague variable names. This process aimed to prepare the dataset for effective predictive modeling. The key variables selected for the machine learning model offer valuable insights into different aspects of a passenger's profile, shedding light on factors influencing survivability.

1. Pclass (Passenger Class): This variable indicates the class of the ticket (1st, 2nd, or 3rd), serving as a proxy for socio-economic status and highlighting how wealth impacted survival chances.

2. Sex: Reflecting the gender of the passenger, this fundamental factor uncovers societal dynamics during the Titanic disaster.

3. Age: A key determinant of vulnerability, the age variable illuminates how different age groups were affected during the tragedy.

4. SibSp (Siblings/Spouses Aboard): Showing the number of accompanying siblings or spouses, this variable hints at collaborative efforts for survival within families.

5. Parch (Parents/Children Aboard): Signifying the number of parents or children traveling together, this variable offers insights into family dynamics and potential altruistic behaviors during the crisis.

6. Fare: The amount paid for the ticket serves as a proxy for socio-economic status, correlating with passengers' financial means and their likelihood of survival.

7. Embarked: Indicating the port of embarkation, this variable reflects potential regional variations in survival, adding a geographical dimension to the analysis.

8. Title: Derived from passenger names, the Title variable provides information about their social or professional status, offering a nuanced perspective on the diverse backgrounds of Titanic passengers.

By excluding non-contributory variables like Name, Cabin, and Ticket, the model prioritizes the most relevant features, streamlining the analysis for a focused exploration of the factors influencing passenger survivability.

2.2 Model Training and Serialization

In the establishment of the predictive model, the incorporation of code from the in-class lab served to provide a foundational framework, ensuring adherence to best practices. The utilization of TPOT and Logistic Regression was undertaken with the objective of conducting a comprehensive exploration of various machine learning algorithms. TPOT, functioning as an automated machine learning tool, facilitated an exhaustive search for optimal model configurations, thereby simplifying the process of model selection. Concurrently, Logistic Regression, recognized for its classical yet robust attributes, contributed to the interpretability and clarity in discerning the impact of individual variables on survivability. The division of the dataset into training and testing sets was executed to serve a dual purpose. This bifurcation allowed the model to assimilate patterns from the training data while maintaining a distinct set for evaluation, ensuring a realistic assessment of predictive capabilities on unobserved data. The emphasis on accuracy as the primary metric was motivated by the necessity for a straightforward and interpretable gauge of overall model performance, aligning with the defined objectives of the project. The exploration of variable relationships within the Titanic dataset was undertaken to identify pivotal factors influencing survivability, as seen in Figure 1. This analytical endeavor was aimed at understanding the relationship among features to enhance model interpretability and potential refinement. Additionally, a detailed evaluation was conducted through the creation of a confusion matrix (Figure 2), elucidating the model's strengths and weaknesses via the categorization of true positives, true negatives, false positives, and false negatives. This evaluative process serves as a foundational step for subsequent phases involving model refinement, optimization, and the groundwork for deployment and serialization. The strategic integration of automation, interpretability, and rigorous evaluation in our chosen approach establishes a framework for the development of an effective predictive model.

In conjunction with the model training phase, the serialization process assumed a crucial role in ensuring the effective storage and deployment readiness of the trained model. Serialization entails the conversion of the model's internal state into a format suitable for convenient storage and subsequent reconstruction. This procedure streamlines the storage of intricate model configurations and establishes a foundation for seamless integration into deployment frameworks. Utilizing the `joblib` library in Python, the serialized model is preserved, enabling swift retrieval and application without necessitating retraining. The compact size of the serialized model enhances its portability, a critical factor for deployment across diverse platforms. Serialization aligns with the project's objectives by facilitating the preservation of the trained model's integrity while enabling straightforward implementation in both local and cloud-based deployment environments. This meticulous approach to serialization reinforces the project's commitment to scalability, reproducibility, and accessibility in deploying the predictive model.

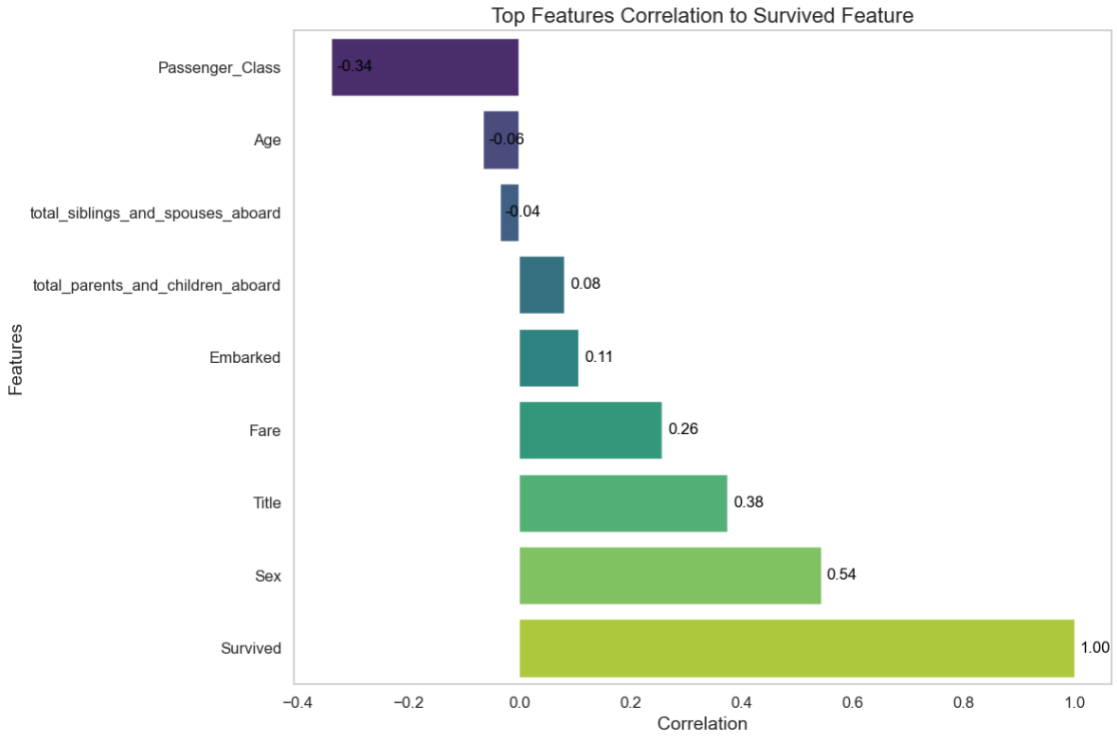

Figure 1: Correction of a Variable to the Survived Variable

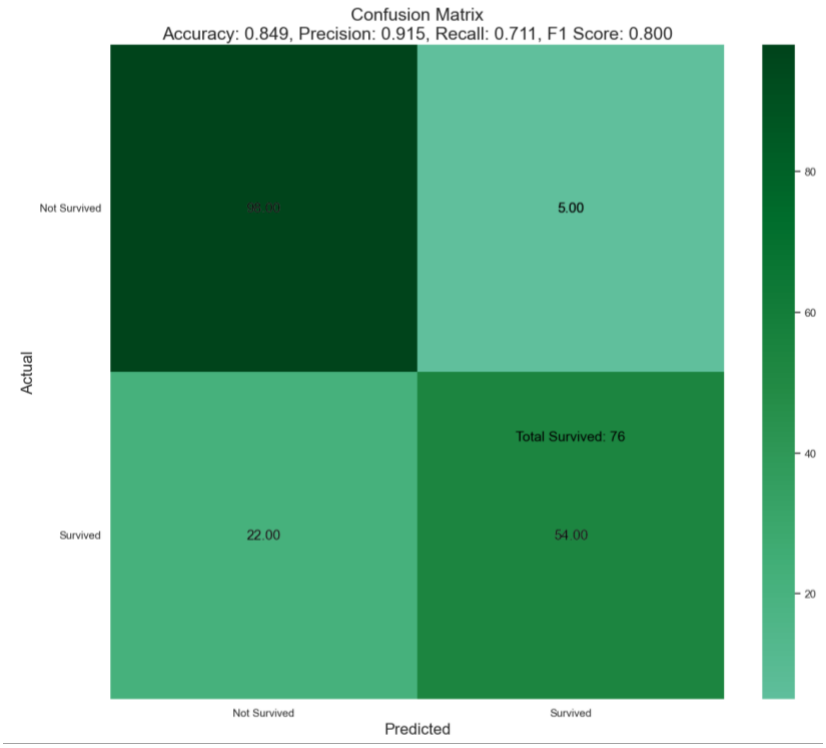

Figure 2: Confusion Matrix

3. Deployment Preparation

3.1 Selection of Deployment Tool/Platform

In the deployment phase, I chose Streamlit as the hosting platform for its user-friendly interface and interactive capabilities, facilitating an engaging user experience. The decision to deploy the application on the web via Heroku was guided by its scalability and reliability, as well as my in class experience. This combined approach, utilizing Streamlit for the application interface and Heroku for web hosting, streamlined deployment. Additionally, Streamlit's versatility allowed for local deployment on my computer, providing a convenient testing environment before the global launch on Heroku.

3.2 Web/API Endpoint Development

Streamlit significantly simplified the development of the web/API endpoint. As a high-level app framework, Streamlit inherently manages user inputs, eliminating the need for an additional framework like Flask. The seamless integration of user interface design and backend logic within Streamlit streamlined the development process, allowing for a cohesive and efficient creation of the web and API endpoint. This choice aligns with the goal of minimizing complexity in deployment while ensuring a user-friendly and responsive interface.

4. Deployment and Testing

4.1 Deployment Process

The local deployment of the app using Streamlit proved to be a straightforward and efficient process. Executing a simple command, 'streamlit run app.py,' within the code folder initiated the app seamlessly. The app displayed responsiveness to changes in input and accuracy in predictions, reflecting the modeled impact of each attribute on survivability, as illustrated in the accompanying graph (Figure 3). Alongside the predictive graph, the app was enhanced with additional features to improve user experience. These included a visually appealing background featuring ‘waves’, clear instructions on app usage in app, and definitions of key terms. However, challenges arose during attempts to deploy the app on the web, specifically on Heroku. Despite similarities in code between my model and the lab's, the deployment encountered a persistent error. This led to the deduction that the issue lay with the model itself rather than the code. The error persisted even when attempting web deployment on Streamlit, indicating a deeper model-related challenge. However, I updated my dependencies in my ‘requirements.txt’ file and that was able to resolve my issue. In the process, I discovered that since I could deploy my app on Streamlit without the need for Heroku, I deployed my app on Streamlit. It can be found at https://cs451app.streamlit.app/ .

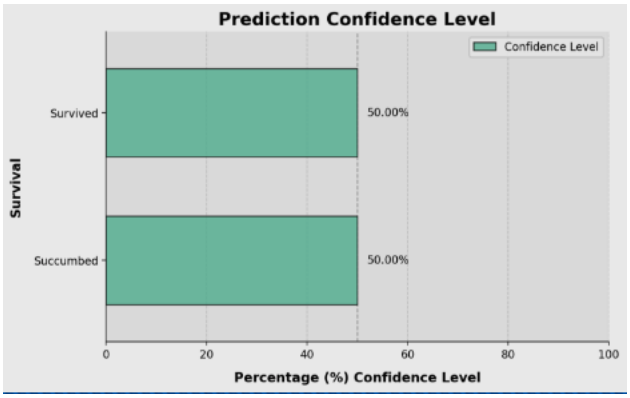

Figure 3: Confidence of Survival

4.2 Testing Methodology

The testing methodology focused on assessing the app's functionality, accuracy, and user experience. During the local deployment, rigorous testing involved scrutinizing the responsiveness of the app to input changes, validating the accuracy of predictions, speed of the response, and evaluating the visuals. For web deployment, the testing methodology shifted towards identifying and resolving errors hindering successful deployment. Diagnostic measures included scrutinizing error logs, debugging the code, and exploring potential discrepancies in the model compared to the lab's model.

4.3 Test Results and Observations

After extensive testing and manipulation of the model, the results aligned with expectations, underscoring the significant roles that sex, socioeconomics, and gendered characteristics played in determining the likelihood of survival. These reinforce historical patterns observed on the Titanic. The impact of these variables on survivability was graphically represented in Figure 1, providing a visual interpretation of the model's predictions. Notably, the variable concerning the number of parents or children onboard initially posed a challenge, displaying a higher succumbing rate. However, a deeper analysis revealed a plausible explanation: individuals with family onboard might have prioritized them, inadvertently leading to a higher risk of demise. This understanding enriched the interpretability of the model's predictions, contributing to a more comprehensive evaluation of its efficacy and real-world implications.

5. Reflection

5.1 Challenges and Solutions

Deployment, particularly on web platforms like Heroku, presented a notable challenge throughout this project. The challenges faced during deployment became opportunities for a thorough code analysis, resulting in a deeper understanding of the codebase. This insight, gained through troubleshooting, might not have been as comprehensive without the encountered deployment error. This also allowed myself to save some money not having to rely on Heroku, as even though the basic plan is up to seven dollars a month, where I have not started a full time job yet, that was troubling my budget. Another challenge confronted in the initial stages was the selection of a suitable dataset or project. The vast array of available datasets and the need for a unique project posed a dilemma. Given the previous exploration of the wine dataset, the decision to delve into the Titanic dataset emerged as an engaging challenge. Emulating and predicting survival outcomes on the Titanic proved to be a captivating and informative endeavor.

5.2 Lessons Learned

The project underscored the significance of persistence as a critical lesson, particularly evident in the commitment to resolving deployment challenges amid technical difficulties. The exploration of diverse hosting services, though unanticipated, yielded valuable insights into various deployment platforms. Despite the project's constraint in hosting the application online, the acquired proficiency in troubleshooting and navigating hosting services establishes a foundation for future tasks. Additionally, the project marked my initiation into developing a model with adjustable inputs, enhancing comprehension of dynamic model interaction. The associated challenge of synchronizing user inputs with the graphical representation contributed to a refined understanding. The newfound capability to manipulate inputs and observe real-time graphical changes augments the versatility of my skill set. Anticipating potential applications, web game development emerges as a prominent consideration for this acquired proficiency. This experience has instilled confidence in my ability to construct and visualize interactive machine learning models. This newfound skill is a valuable

asset that I can carry into future classes and my post-graduation career. In hindsight, I wish I had encountered this class earlier in my career, but the lessons learned will undoubtedly enhance my professional contributions moving forward.

6. Conclusion

In working towards the goal of training a machine learning model and attempting to deploy it as a web service, the Titanic survival prediction project has been a valuable journey in data science. Despite facing challenges, successfully deploying it using Streamlit showcased the model's effectiveness. Dealing with these challenges improved my technical proficiency and problem-solving skills. Applying machine learning techniques to a real-world dataset and developing an intuitive user interface broadened my insights into data science. As I wrap up this project, the positive experiences highlight the importance of overcoming obstacles and provide a solid foundation for future endeavors in the dynamic realm of data science.

7. References

Zhu, E. (2022). wine-example-app. GitHub. https://github.com/EthanZhu23/wine-example-app

Will Cukierski. (2012). Titanic - Machine Learning from Disaster. Kaggle. https://kaggle.com/competitions/titanic