SPACESHIP TITANIC

- Shanthini Joshitha P (125018060)

TABLE OF CONTENTS

SNo	Topic	Page No
1	Abstract	2
2	Introduction	2
3	Related Work	5
4	Background	5
5	Methodology	6
6	Results	12
7	Discussion	13
8	Learning outcome	14
9	Conclusion	15

Abstract

The "Spaceship Titanic" Kaggle competition presents an intriguing challenge: predicting whether passengers aboard the Spaceship Titanic were transported to an alternate dimension. This task involves analysing a rich dataset that includes various features about the passengers, such as their demographics, travel details, and usage of onboard amenities.

By undertaking this project, I aim to explore various machine learning models to find the most effective one for accurately classifying each passenger as "Transported" or "Not Transported." This requires understanding the dataset's characteristics and identifying patterns and relationships that may influence a passenger's outcome.

The dataset consists of labelled data, which can be used for training the model, and unlabelled data for making final predictions. This project emphasises the importance of data preprocessing, feature engineering, and selecting the right algorithms to tackle this classification problem effectively. Through careful analysis and model development, I seek to uncover insights that explain the factors contributing to a passenger's transport status, adding depth to the understanding of this unique scenario.

Introduction

1. **Importance of dataset**

   The dataset plays a vital role in this project, offering a detailed snapshot of each passenger aboard the Spaceship Titanic. It contains a wide array of attributes that are key to developing a machine learning model capable of predicting whether passengers were transported to an alternate dimension.

   One important aspect of the dataset is the demographic information it provides, including age, gender, and home planet. These details help to identify patterns among different groups of passengers. For example, there could be noticeable trends showing that people from certain planets or specific age groups have different probabilities of being transported. Understanding these demographics is crucial as it sets a foundation for the model.

   Another significant part of the data is the cabin allocation and other travel details, such as group, deck, and specific cabin number. These elements might indicate certain sections or areas on the ship that have different chances of being linked to transportation events. By looking into these travel-related attributes, the model can pick up on any spatial or logistic patterns that may be influencing the outcomes.

   Additionally, the dataset offers insights into spending habits and the use of onboard amenities, like dining, entertainment, and other services. By studying this information, it is possible to spot behavioural trends, such as whether passengers who spent more on specific services had a higher or lower likelihood of being transported. These spending patterns can provide extra clues that help refine the model’s predictions.

Overall, this dataset, with its diverse range of features, allows for an in-depth analysis of different variables that might affect the outcome. Using this information, the project aims to identify key factors and patterns that improve the accuracy and reliability of the predictive model.

2. Project objectives

The primary goal of this project is to develop a machine learning model that can accurately predict whether a passenger aboard the Spaceship Titanic was transported to an alternate dimension. This involves tackling several complex challenges and requires a systematic approach to ensure the model's predictions are both precise and reliable.

One of the core tasks is to address data quality issues, particularly dealing with missing information. The dataset contains some gaps, which must be managed carefully to prevent them from affecting the model's performance. This may involve techniques such as imputation, where missing values are estimated based on available data, or strategies to handle the absence of information in a way that does not introduce bias.

Another important aspect is encoding categorical variables. The dataset includes features like passenger names, cabin locations, and home planets, which are non-numerical and need to be converted into a form that the machine learning model can interpret. Effective encoding ensures that the model understands the relationships between different categories without losing crucial information. Various methods can be explored, including one-hot encoding and ordinal encoding, depending on the nature of each variable.

The project also emphasises the need for feature selection. Not all attributes in the dataset may contribute equally to predicting whether a passenger was transported. Therefore, it is necessary to identify and focus on key features that have the most impact on the prediction outcome. This process involves careful analysis to determine which variables hold significant predictive power and which can be excluded without compromising the model's accuracy.

The overall objective is to create a robust and reliable machine learning model that can make accurate predictions on new, unseen data. This means the model should not only perform well on the training data but also be able to generalise effectively when applied to different datasets. The ultimate aim is to develop a solution that combines accuracy with generalisability, ensuring consistent performance across various scenarios and datasets.

3. Methodology

The project involves a comprehensive data preprocessing phase, which is essential for ensuring the machine learning models are trained on clean and meaningful data. This phase includes several critical steps, starting with data cleaning, where inconsistencies, errors, and irrelevant entries are identified and corrected. Proper cleaning helps to ensure that the dataset is accurate and reliable, reducing the risk of skewed or misleading results.

A significant part of the preprocessing is handling missing values. The dataset may have gaps in the information, which need to be addressed carefully. Various strategies can be employed, such

as imputing missing data using statistical methods, removing incomplete records, or using algorithms that can work with missing inputs. Choosing the right approach is crucial for maintaining the integrity of the dataset and ensuring accurate model predictions.

Feature engineering is another key step. This involves creating new variables or modifying existing ones to enhance the model's ability to capture patterns within the data. Through thoughtful feature engineering, it is possible to reveal hidden relationships and insights that can improve the overall performance of the model. Additionally, scaling the features ensures that numerical values are on a similar scale, preventing any single feature from disproportionately affecting the model.

After preprocessing, the next stage is to train and evaluate various machine learning models. Algorithms such as Logistic Regression, Random Forest, and Bayesian Networks will be implemented, each offering unique strengths for different aspects of the problem. These models will be rigorously tested and compared to determine which one provides the most accurate and reliable predictions.

To assess the performance of these models, evaluation metrics like accuracy, precision, recall, and F1 score will be used. Accuracy provides a general measure of how often the model's predictions are correct, while precision and recall offer more nuanced insights into its behaviour, particularly in identifying true positives and avoiding false positives. The F1 score balances precision and recall, giving a clearer picture of the model's overall effectiveness. By combining these metrics, the project can thoroughly evaluate each model's strengths and weaknesses, leading to the selection of the most suitable solution.

4. Results Obtained

The evaluation of the various machine learning models revealed that multiple algorithms demonstrated comparable accuracy, with performance metrics ranging between 74% and 79%. Among the models tested, Logistic Regression emerged as the most effective, achieving a notable test accuracy of 78.99%. This model's performance indicates its capability to effectively capture the underlying patterns in the dataset, translating to a high level of accuracy in predicting whether a passenger was transported to an alternate dimension.

The close performance among the models suggests that the dataset contains a rich set of features that various algorithms can leverage for accurate predictions. While Logistic Regression led the way, other models also performed admirably, providing a strong foundation for further exploration and potential enhancements. This outcome highlights the importance of selecting the right model based on specific project requirements and underscores the value of conducting thorough evaluations across different approaches to identify the most suitable solution.

Ultimately, the promising accuracy levels achieved indicate that the chosen models are well-equipped to generalise effectively to unseen data, paving the way for potential deployment in practical applications related to passenger transportation prediction.

5. **Document Structure**

The document is structured into several key sections to provide a comprehensive overview of the project and its findings. It begins with an introduction that sets the stage for the study, outlining the objectives and the importance of accurately predicting transportation outcomes.

Next, the document delves into the background on models and preprocessing techniques, detailing the various machine learning algorithms employed and the essential data preprocessing steps taken to ensure the integrity and usability of the dataset. This section provides insight into the methodologies applied and the rationale behind the chosen approaches.

Following this, the experimental design section describes the framework used to evaluate the models, including the selection of performance metrics and the procedures for training and testing the algorithms. This part of the document outlines the systematic approach taken to achieve reliable results.

The results section presents the findings from the model evaluations, highlighting the performance metrics achieved and comparing the efficacy of the different algorithms used in the study. This analysis is crucial for understanding the strengths and weaknesses of each model.

Related Work

1. **Sources Consulted**

The project was primarily informed by resources available on Kaggle, alongside valuable insights provided by ChatGPT. These tools played a crucial role in deepening the understanding of various aspects of machine learning, particularly in data preprocessing, feature engineering, and model evaluation.

Kaggle's extensive library of tutorials, datasets, and community discussions offered practical examples and methodologies that were instrumental in shaping the project's direction. The platform's hands-on approach facilitated the exploration of different techniques and best practices, ensuring a robust framework for tackling the challenges of the task.

In addition, the insights gained from ChatGPT provided further clarity and guidance, assisting in the formulation of strategies for effective model development and testing. The combination of these resources fostered a structured approach to the project, enabling the systematic building and evaluation of machine learning models. This synergy of tools not only enhanced the understanding of the technical aspects but also contributed to the overall success of the project.
Kaggle

Background

1. **Models Used**

1.1. **Logistic Regression**: A simple yet effective linear model for binary classification, Logistic Regression estimates the probability of an outcome based on predictor variables. Its interpretability and efficiency make it a popular choice for many tasks, especially when relationships are approximately linear.

1.2. **Decision Tree**: A model that represents decisions in a tree-like structure, Decision Trees split data based on feature values to capture both linear and non-linear relationships. They are easy to interpret and can handle both categorical and numerical data, though they can be prone to overfitting.

1.3. **Random Forest**: An ensemble method that combines multiple decision trees, Random Forest improves accuracy by averaging predictions from diverse trees trained on different data subsets. This approach captures complex patterns and interactions while reducing overfitting, making it robust against noise.

1.4. **Bayesian Network**: A probabilistic model that represents variables and their dependencies using a directed acyclic graph, Bayesian Networks facilitate reasoning under uncertainty. They effectively capture relationships among variables and can incorporate prior knowledge, allowing for updates with new evidence.

2. **Preprocessing Techniques**

2.1. **Data Cleaning**: Missing values were addressed using mean imputation for numerical features, ensuring that the data remains robust while maintaining overall trends. For categorical data, the most frequent strategy was employed, allowing for a straightforward way to fill gaps without introducing bias.

2.2. **Scaling and Encoding**: Standard scaling was applied to numerical features to ensure that they have a mean of zero and a standard deviation of one, which is essential for many machine learning algorithms. One-hot encoding was used for categorical variables, transforming them into a format suitable for model training while avoiding ordinal relationships.

2.3. **Dimensionality Reduction**: Principal Component Analysis (PCA) was utilised to reduce the dataset to 10 principal components. This technique helped retain significant variance in the data while eliminating noise, allowing for improved model performance and faster computation times.

Methodology

1. **Experimental Design**

The dataset was systematically divided into three subsets: training (70%), validation (15%), and testing (15%). This allocation ensured that the models had sufficient data to learn from while also allowing for robust evaluation. Machine learning models were trained and evaluated on these subsets, utilising performance metrics such as accuracy, precision, recall, and F1 score to measure their effectiveness.

To enhance the reliability of the results and ensure the models could generalise well to unseen data, cross-validation was employed. This technique involved partitioning the training data into multiple subsets, enabling the models to be trained and validated on different combinations of data. By doing so, it helped to mitigate the risk of overfitting, ensuring that the models perform well not just on the training set but also on new, unseen data.

2. **Environment and Tools Used**

The project was developed using Python, a versatile programming language well-suited for data analysis and machine learning. Key libraries were employed to facilitate various tasks, including

pandas for data manipulation and analysis, scikit-learn for implementing machine learning algorithms and evaluating model performance, and matplotlib for visualising results and data distributions.

Experiments and code execution were conducted on Google Colab, which provided a cloud-based environment with access to powerful computational resources. This platform enabled efficient processing of large datasets and facilitated collaboration, allowing for seamless sharing and iteration on the project. By leveraging Google Colab's capabilities, the project benefited from increased computational power and flexibility, making it easier to explore different models and techniques.

3. Code Location

All code files related to the project, including preprocessing scripts and model training notebooks, are stored in a dedicated GitHub repository. This repository serves as a centralised location, ensuring easy access for anyone interested in reviewing or replicating the work. By maintaining the code on GitHub, reproducibility is enhanced, allowing other researchers and practitioners to follow the methodologies employed and build upon the findings of this project. The structured organisation of the repository also facilitates collaboration and version control, making it an ideal platform for ongoing development and sharing within the data science community.

4. Preprocessing Steps

4.1. **Dataset Size and Feature Details**: The dataset consists of records for 14,000 passengers, encompassing 12 features that provide crucial information for analysis. After undergoing thorough data cleaning, which included addressing missing values, as well as scaling and encoding, the final feature set was meticulously prepared for modeling. This preparation ensured that the data was in an optimal state for the subsequent machine learning processes.

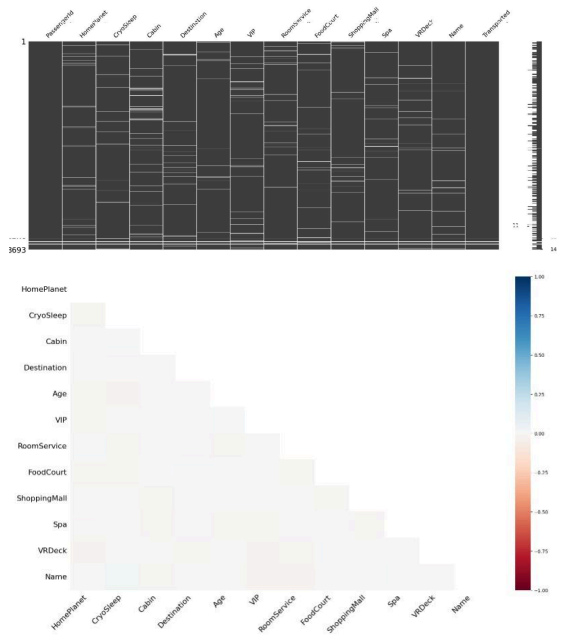
4.2. **Outlier Analysis and Feature Reduction**: To enhance the model's performance and interpretability, Principal Component Analysis (PCA) was employed to reduce the dataset to 10 principal components. This technique effectively eliminated noise while preserving critical information, allowing the model to focus on the most informative aspects of the data. Additionally, an outlier analysis was conducted; however, no significant outliers were detected that could potentially impact model performance. This thorough preprocessing step ensured a robust foundation for the training and evaluation of machine learning models.

5. Exploratory Data Analysis

Topic	Graph

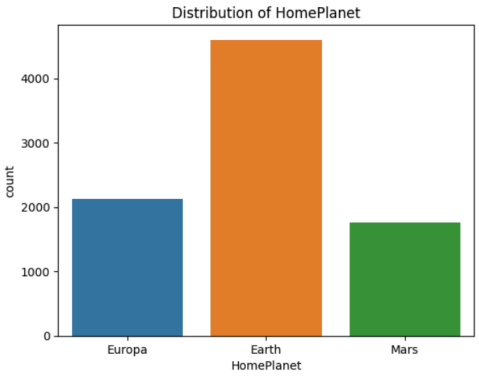
Missing Values Visualization

1. *msno.matrix(train)*: Visualises missing values across the dataset, with gaps indicating absent data, helping identify concentrated missingness.
2. *msno.heatmap(train)*: Displays correlations in missing values between columns, highlighting patterns of missingness.



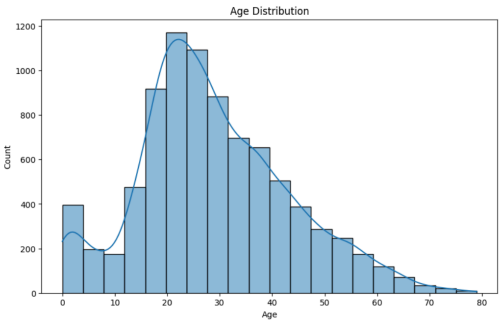
Categorical Variables Analysis

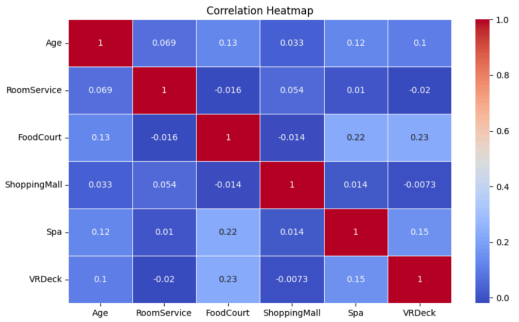
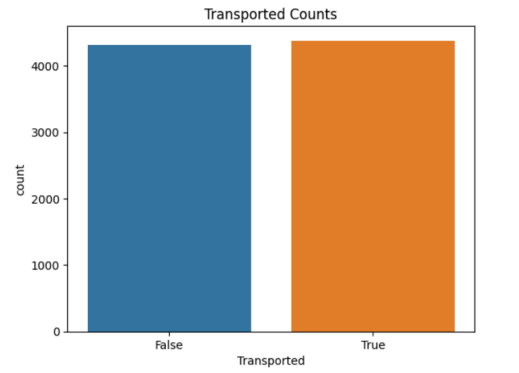
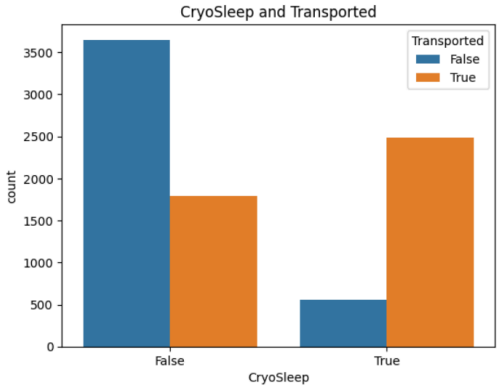
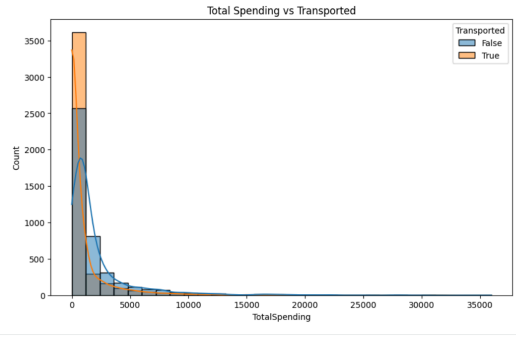
HomePlanet Distribution:
sns.countplot(data=train, x='HomePlanet'): Shows the distribution of passengers from different home planets, revealing common origins.

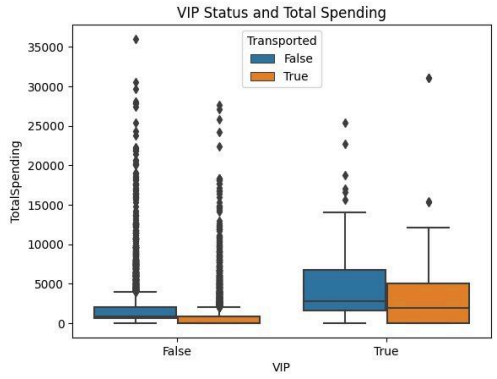
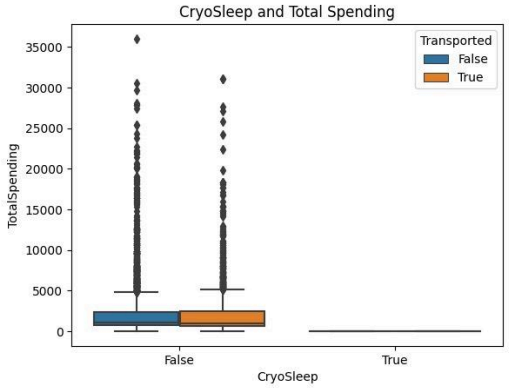
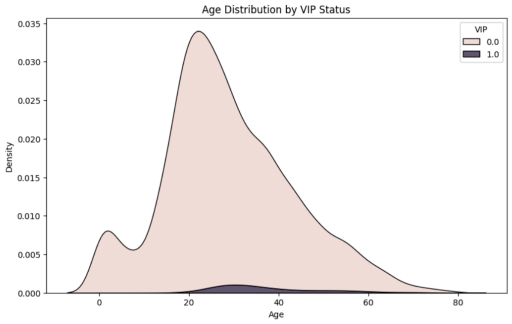
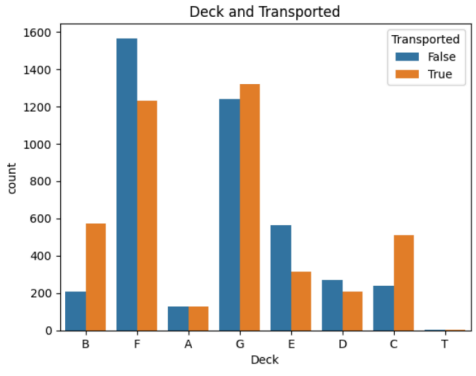


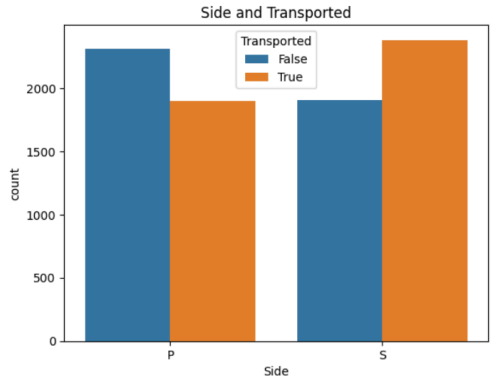
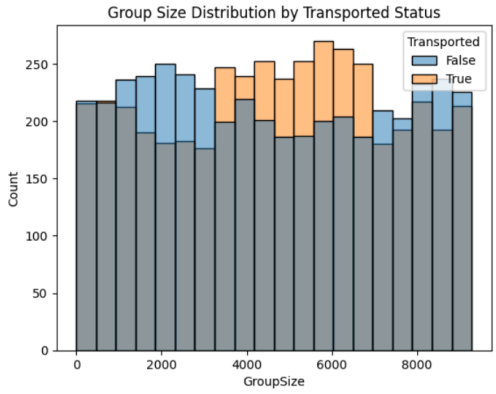
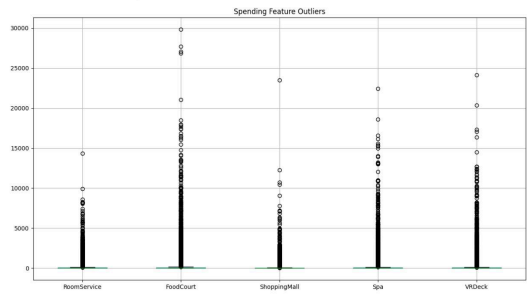
Age Distribution

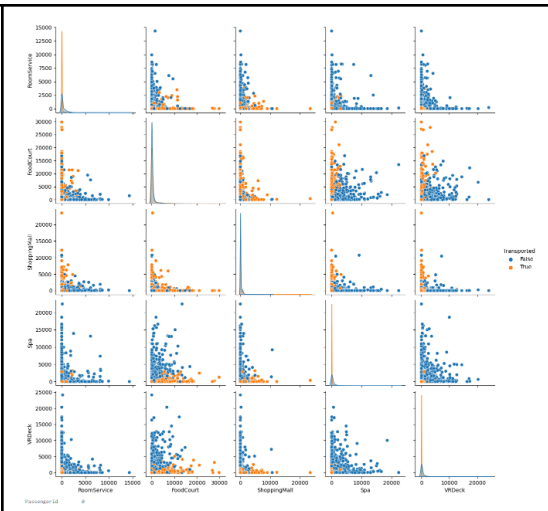
sns.histplot(train['Age'], bins=20, kde=True): Illustrates age distribution with a histogram, enhanced by a Kernel Density Estimation curve.



Correlation Heatmap *sns.heatmap(train[numerical_features].corr())*: Highlights correlations among numerical features, with warmer colours indicating stronger relationships.	
Transported Count *sns.countplot(data=train, x='Transported')*: Displays the count of transported versus non-transported passengers, assessing class balance in the target variable.	
CryoSleep and Transported *sns.countplot(data=train, x='CryoSleep', hue='Transported')*: Analyses how CryoSleep status correlates with transport outcomes.	
Total Spending vs Transported *sns.histplot(data=train, x='TotalSpending', hue='Transported', kde=True)*: Examines spending patterns based on transport status, indicating if higher spenders were more likely to be transported.	

VIP Status and Total Spending *sns.boxplot(data=train, x='VIP', y='TotalSpending', hue='Transported')*: Compares spending habits between VIP and non-VIP passengers in relation to transport status.	
CryoSleep and Total Spending *sns.boxplot(data=train, x='CryoSleep', y='TotalSpending', hue='Transported')*: Investigates spending variations among CryoSleep passengers concerning transport status.	
Age Distribution by VIP Status *sns.kdeplot(data=train, x='Age', hue='VIP', multiple='stack')*: Displays age distributions for VIP and non-VIP passengers, facilitating comparison.	
Deck and Side Analysis 1. *sns.countplot(data=train, x='Deck', hue='Transported')*: Visualises passenger distribution across cabin decks regarding transport status. 2. *sns.countplot(data=train, x='Side', hue='Transported')*: Examines the effect of ship side (port or starboard) on transportation outcomes.	

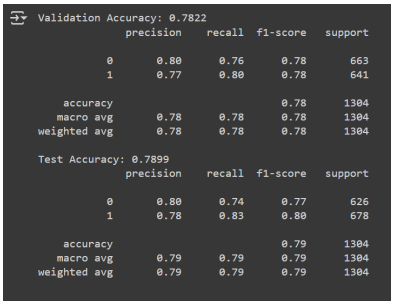
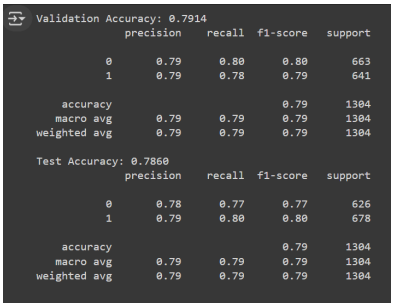
	
Group Size Distribution *sns.histplot(data=train, x='GroupSize', hue='Transported', bins=20)*: Analyses group sizes to see if larger groups were more likely to be transported.	
Spending Feature Outliers *train[spending_columns].boxplot():* Identifies outliers in spending features, indicating unusual spending patterns.	

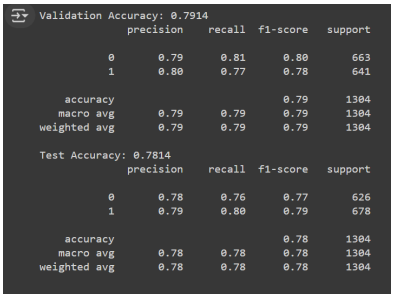
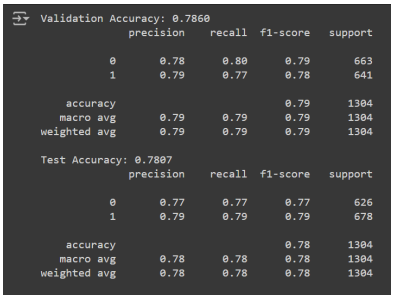
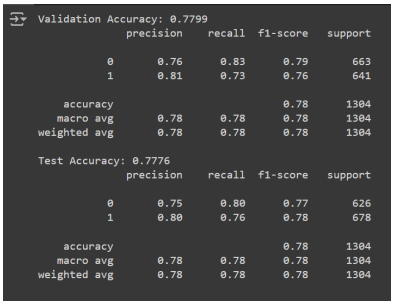
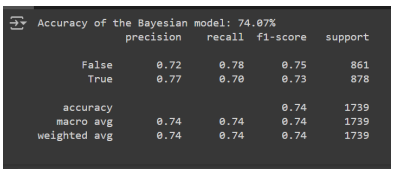
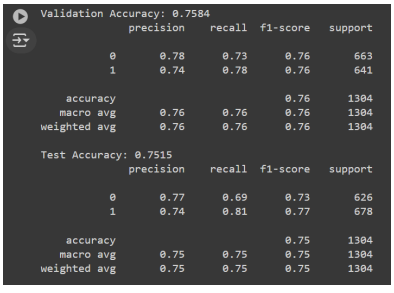
Pairplot for Spending Features *sns.pairplot(train[spending_columns + ['Transported']], hue='Transported', diag_kind='kde'):* Visualises pairwise relationships among spending features in relation to transport status.	

Results

Results Obtained

Various models, including Logistic Regression, Random Forest, and Bayesian Network, were evaluated. Logistic Regression achieved the highest accuracy at 78.99%, while other models showed similar results, ranging between 74% and 79%. Random Forest models with 20 and 50 estimators, as well as ensemble approaches, performed similarly to Logistic Regression.

Model	Accuracy (%)	Screenshot
Logistic Regression	78.99%	
Random Forest (100 estimators)	78.60%	

Random Forest (50 estimators)	78.14%	
Random Forest (20 estimators)	78.07%	
Ensemble of Logistic Regression and Random Forest (20 estimators)	77.76%	
Bayesian Network	74.07%	
Decision Tree	75.15%	

Discussion

1. **Overall Results**

The results indicate that **Logistic Regression** outperformed all other models with an accuracy of **78.99%**, making it the most effective choice for predicting passenger transport status in this project. Notably, its performance is closely matched by **Random Forest models**, particularly the one with 100 estimators, which achieved an accuracy of **78.60%**. The accuracies of the various Random Forest configurations are all relatively similar, ranging from **78.07% to 78.60%**. In contrast, the remaining models, including the Bayesian Network and Decision Tree, showed lower accuracy rates. Overall, while Logistic Regression stands out as the best-performing model, the closeness of the results among Logistic Regression and Random Forest models suggests that either could be a viable option for this binary classification task, emphasising the effectiveness of these algorithms in this dataset.

2. **Hyperparameter tuning**

   Adjustments in hyperparameters, especially for the Random Forest model, resulted in slight improvements in performance. Tweaking parameters like the number of estimators and depth of trees helped optimise the model's accuracy and generalisation. Further tuning and exploration of various parameter settings are planned to see if additional gains in accuracy can be achieved, ensuring the model is well-suited for diverse scenarios.

3. **Model comparison and selection**

   Despite experimenting with various models, Logistic Regression was ultimately chosen for its simplicity, interpretability, and slightly higher accuracy compared to the others. While ensemble methods, such as combining Random Forests and Logistic Regression, showed potential, they did not outperform Logistic Regression by a significant margin. However, these methods remain promising and may be revisited in future evaluations.

Learning Outcome

1. **Links:** [Spaceship-Titanic]
2. **Skills and Tools Used**
   - 2.1. Skills
     - 2.1.1. Data cleaning and preprocessing
     - 2.1.2. Feature engineering and selection
     - 2.1.3. Statistical analysis and visualisation
     - 2.1.4. Development and optimisation of machine learning models
   - 2.2. Tools
     - 2.2.1. Python for scripting and model development
     - 2.2.2. Pandas for data manipulation and analysis
     - 2.2.3. Scikit-learn for building and evaluating machine learning models
     - 2.2.4. Google Colab for code execution and resource management
     - 2.2.5. GitHub for version control and code sharing
3. **Dataset Used**

   The dataset originates from the Spaceship Titanic competition on Kaggle. It includes comprehensive passenger information, such as demographic details, cabin assignments, and their final transport outcomes. This data provides a foundation for analysing patterns and building predictive models to determine which passengers were transported to an alternate dimension.

4. **Learnings**
   The project provided insights into effective data preprocessing, handling missing values, and implementing machine learning pipelines. Understanding model selection and hyperparameter tuning was also a key takeaway.

Conclusion

1. **Concluding remarks**
   The project successfully developed a predictive model for the Spaceship Titanic competition. Results indicated that demographic and spending features played a role in predicting transport outcomes, with logistic regression achieving the highest accuracy among the models tested. However, the performance was not significantly different across models, suggesting room for improvement through more refined feature engineering and hyperparameter tuning.
2. **Accomplishment of objectives**
   The project achieved its primary objective of building an accurate model, effectively framed the problem as a binary classification task, and employed systematic preprocessing and model evaluation strategies
3. **Advantages and Limitations**
   3.1. Advantages: Simple models like Logistic Regression provided interpretable results. The preprocessing pipeline ensured robust data handling, and PCA helped reduce data complexity.
   3.2. Limitations: The dataset's feature complexity might limit model performance, and slight improvements in accuracy indicate a need for further tuning and feature engineering.