COMP1816 - Machine Learning Coursework Report

Sneha Naidu - 001193003
Word Count: 2058

1. Introduction

In this project, we explored the California Housing and Titanic Survival datasets to predict housing prices and passenger survival. The process began with data analysis and visualization, which was crucial for a comprehensive understanding of the factors at play. Data pre-processing followed, streamlining the datasets for the subsequent application of machine learning models. During this stage, the pivotal task was feature scaling, crucial for maintaining a balanced influence of all variables in the predictive process. We then separated target outcomes from features, with normalization to ensure data uniformity. After splitting the data into training and testing sets, the focus shifted to model tuning. In this stage, various hyperparameters were tuned through different methods, using specific performance metrics to evaluate and enhance the models' performance. Let's delve into the specifics in the report below.

2. Regression

2.1. Pre-processing

Starting with the California Housing Dataset, my first step was to get a good look at the data. I decided to drop the 'No.' column which held no predictive value for housing prices. For a closer look, I turned to visualization—graphs and bar charts for each feature. The distribution plot for the median house value showed a skew in the data, suggesting that while some houses were priced on the higher end, many more clustered in the mid-to-lower range, as shown in Figure[1].

The heatmap signals which features might influence house prices most Figure[2]. High-income levels correlate with pricier homes, and the further south you go (lower latitude), the higher the house values tend to be. The scatter plot with a regression line showcases a clear positive correlation, indicating that as median income increases, median house value tends to rise as well Figure[3]. To prepare for prediction, I filled in missing data using the median, maintaining a true representation of the dataset. Implementing one-hot encoding for 'ocean_proximity' translated this categorical data into a numerical form so that models can interpret. We perform feature normalization and scaling to achieve the same scale for all features. Then I remove the target feature to get 'X' for features and 'y' for the target, then split the dataset into an 80-20 train-test ratio.

2.2. Methodology

In determining the most effective approach to predict housing prices within the California Housing Dataset, the Random Forest Regressor emerged as the main model. This choice was influenced by the model's ensemble method, which leverages multiple decision trees to produce a more accurate and stable prediction by averaging the results, thus minimizing overfitting. Mathematically, if we consider D as our dataset and T as a decision tree, the Random Forest model prediction (RF) can be represented as:

Mathematical Definition:

$$RF(x) = \frac{1}{N} \sum_{i=1}^{N} T_i(x; \Theta_i, D)$$

where N is the number of trees, x is the input variable, and i represents the parameters of the i th tree learned from the dataset D.

For hyperparameter tuning, the GridSearchCV method was instrumental. It optimized parameters like 'max-depth' , 'n-

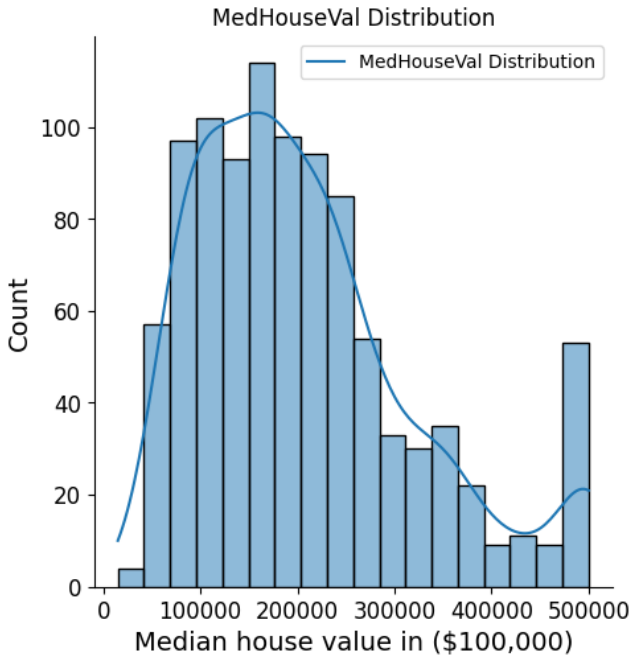

Figure 1.

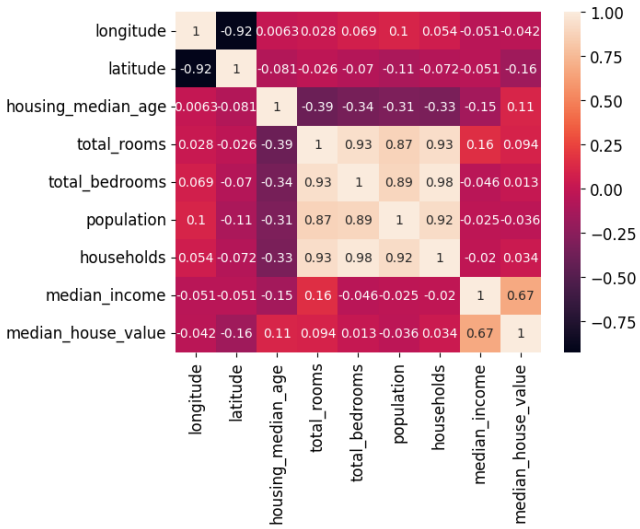

Figure 2. Co-relation matrix of features

Figure 3.

estimators', and 'min-samples-split' ensuring the model was neither too simple nor too complex that is ensuring a perfect balance to avoid overfitting while maximizing accuracy. This fine-tuning led to a significant enhancement in the model's performance, evidenced by a high R-squared value. I also explored other models for a comprehensive analysis. The Decision Tree Regressor, while simpler, offered baseline insights. Linear Regression, enhanced with polynomial features and scaling, addressed the dataset's linear aspects. Lastly, the SVR model, after extensive tuning, provided a different perspective, although it didn't outperform the Random Forest. The Random Forest Regressor, optimized with its balanced approach to bias and variance, emerged as the superior model for predicting California housing prices due to its comprehensive performance and ability to capture the complexity of the data, forming the basis of our approach.

2.3. Experiments

2.3.1. Experimental Settings

Initially, we began with Linear Regression as our baseline model. It offered an initial glimpse into the dataset's behaviour but lacked the depth needed for complex analysis since it didn't support hyperparameter tuning. Seeking more detailed insights, I explored the DecisionTreeRegressor. Where I adjusted settings like 'max-depth' to better capture the nuances without overfitting, with the help of GridSearchCV, which identifies the best combination that maximizes model accuracy and robustness.

To tackle dataset complexity, I shifted to RandomForestRegressor and SVR, fine-tuning parameters like 'n-estimators' and 'min-samples-split' for enhanced prediction through diverse decision trees. SVR adjustments included 'C', 'gamma', and 'epsilon' for handling non-linearities. Among tested models, RandomForestRegressor excelled, not merely for its R-squared score but for its optimal bias-variance balance, marking it as our top choice due to its adaptability and depth of insight.

2.3.2. Results

In the analysis phase of this report, the evaluation metric chosen was R-squared. To measure how good our models are it tells us in a percentage how much of the changes in house prices can be predicted by the features we use. For Example we have R-square of a model as 70% that means model can explain 70% of the variation in house prices". This straightforward interpretation advantageously contrasts with metrics like RMSE, which, despite its utility in quantifying prediction errors, does not offer an immediate grasp of model effectiveness in variance explanation.

Comprehensive Results:

Model	R-squared	RMSE
DecisionTreeRegressor	0.5578	71786.87
RandomForestRegressor	0.6917	59942.09
LinearRegression	0.5663	71095.20
SVR	0.5063	75853.10

Table 1. Model comparison with R-squared values and additional metric.

2.3.3. Discussion

In my modeling process, I utilized grid search with cross-validation to identify the best-performing model. This method assesses different parameter sets and validates model accuracy across varied data segments. Here's a brief overview:

Decision Tree Regressor: Using 3-fold cross-validation, tested six configurations, finding the best at a max-depth of 15, achieving an R-squared of 0.5578 and RMSE of 71786.87

RandomForestRegressor: Applied 5-fold cross-validation on 36 setups. Optimal settings included a max-depth of 20, min-samples-split of 2, and 120 estimators, resulting in an R-squared of 0.6917 and RMSE of 59942.09. Linear Regression: Evaluated three polynomial degrees; degree 1 was best, with an R-squared of 0.5663 and RMSE of 71095.20.

SVR: Conducted 3-fold cross-validation across 48 configurations, selecting C=100, epsilon=0.2, gamma='scale', and kernel='linear', but scored lower in performance with an R-squared of 0.5063 and RMSE of 75853.10. This methodical

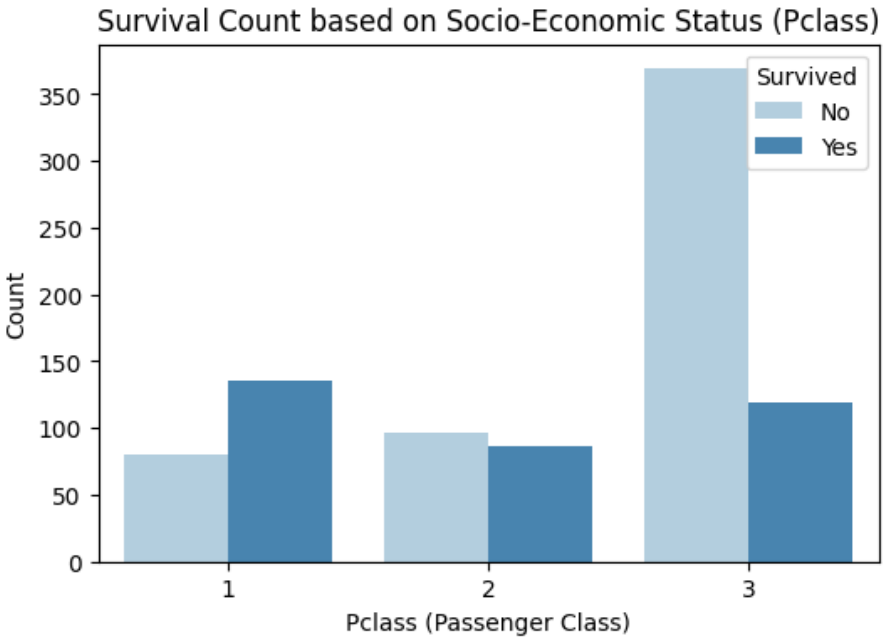

Figure 4.

evaluation pinpointed the RandomForestRegressor as the top model, due to its adeptness in navigating complex data patterns.

3. Classification

3.1. Pre-processing

Starting with the Titanic dataset, I first loaded it to understand the data. This helped me identify which columns weren't going to be useful for predicting who survived. For instance, names and ticket details didn't seem helpful, so I decided to drop them to decrease the models load.

Then I moved to Data Analysis and Visualization. I plotted various graphs to understand the patterns in the data and which factors affected the Survival rate. Like, passengers in first class had a better chance of surviving than others as in Figure[**??**]. And, depending on their age and sex, some passengers were more likely to survive than others as shown in Figure[**??**]. The graphs also showed that men and women had different survival rate, with women with higher rate of survival. Figure[6].

The heatmap of all the data pointed out which factors might influence survival. For example, it showed that age and ticket class were big factors. Additionally, the number of siblings/spouses aboard (SibSp) also hinted at survival probabilities, indicating that those with fewer family members aboard were more likely to survive. After spotting some missing information in important areas like age and passenger class, I filled those gaps with median values or placeholders to keep the data consistent.

Next I turned the 'Sex' column from words into numbers using Label Encoding so that our models could understand it better. Then I normalized the data using Min-Max Scaling. After that I split the data into Two sets Target and Feature further diving them into Training and Testing data 650 datapoints for training and the rest 240 for testing . With the data now ready and set up nicely, we're all prepped to start using models.

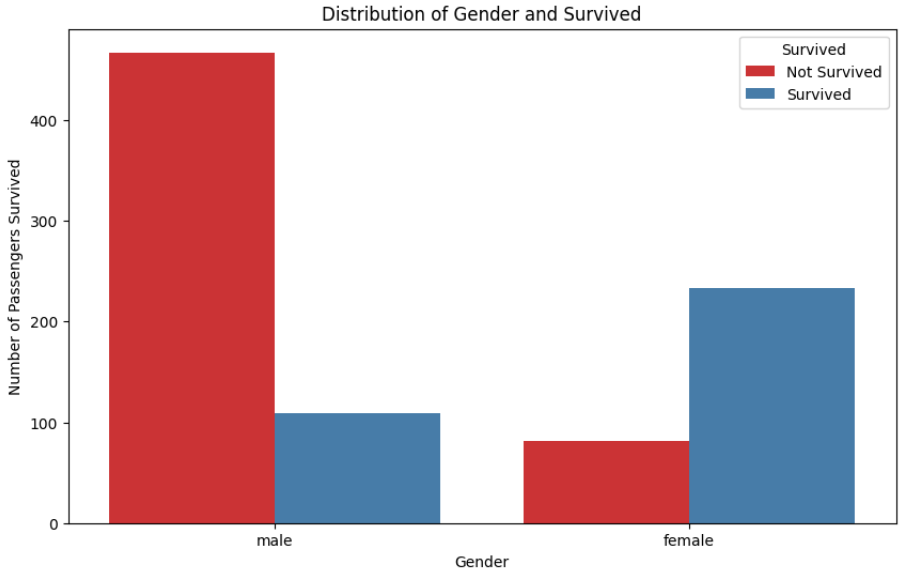

Figure 5.

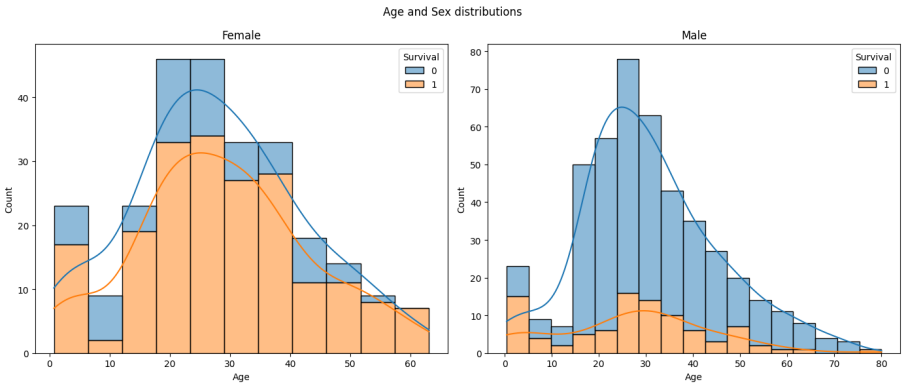

Figure 6.

3.2. Methodology

We expiremented with various classification models Decision Trees and Logistic Regression and Random Forest Regression. The results leading to Decision Tree being our Main Model. The Decision Tree Classifier excelled in our analysis for its clarity and adaptability. Its interpretability is a major advantage, letting us follow the thought process behind each prediction. This is especially valuable for historical data like ours, where understanding the reasons behind survival is as important as knowing who survived. Mathematically, the operation of a Decision Tree can be summarized with a formula representing the decision-making process from the root to the leaves:

Mathematical Definition:

$$DT(x) = \sum_{i=1}^{N} c_i \cdot I(x \in R_i)$$

In this formula:

- $DT(x)$ denotes the Decision Tree's prediction for an input x.
- N is the number of leaf nodes, each corresponding to a decision outcome.
- c_i is the outcome predicted within the i^{th} leaf node.
- I is an indicator function that returns 1 if x falls within the region R_i associated with leaf node i, and 0 otherwise.

Through this equation, we can see how the Decision Tree classifies each individual by following a series of decisions (I(xRi)) until reaching a conclusion (ci) at a leaf node. This clarity and sequential breakdown of decisions underscore why the Decision Tree is especially suited to our analysis of the Titanic dataset.

3.3. Experiments

3.3.1. Experimental Settings

In our experiment, we explored several models to analyze the Titanic dataset, beginning with Logistic Regression, advancing to models such as Random Forest and Decision tree. For hyperparameter I used Grid Search CV a powerful tool that searches through a range of parameter values to find the most effective settings for each model.

RandomForestClassifier: Tuned 'max-depth' and 'n-estimators', achieving the best performance with a depth of 5 and 200 trees, which significantly improved accuracy and precision.

DecisionTreeClassifier: Adjusted 'max-depth' and 'min-samples-split', finding optimal settings at a depth of 5 and a minimum of 2 samples to split a node. This improved the recall and F1 Score, indicating a good precision-recall balance.

LogisticRegression: Optimized 'C' for regularization strength and 'penalty' type, with the best results at 'C': 1.623776739188721 and 'penalty': 'l2', enhancing balanced accuracy and precision.

Through GridSearchCV, we pinpointed the best settings for each model on the Titanic data, enhancing accuracy and offering insights into optimal model behavior for survival prediction.

3.3.2. Results

In evaluating the performance of our models on the Titanic dataset, we utilized several classification metrics: accuracy, precision, recall, and F1 Score. Among these, we chose to focus primarily on the F1 Score for its comprehensive ability to balance the trade-offs between precision and recall. Precision evaluates how well the model identifies true positives without inflating survival rates, while recall ensures all actual positives are captured, minimizing overlooked survivals.

F1 Score, combining precision and recall, serves as a unified metric reflecting their equilibrium, crucial for accurately identifying survivors without overlooking any. This balance is vital in predicting Titanic survival, ensuring precise survivor identification and minimal misclassification.

The results table includes the F1 Score and other metrics for a holistic performance overview.

Model	Best CV Accuracy	Test Accuracy	Precision	Recall	F1 Score	Balanced Accuracy
RandomForest	0.8046	0.8319	0.8197	0.8129	0.8160	0.8129
DecisionTree	0.7831	0.8361	0.8219	0.8389	0.8276	0.8389
LogisticRegression	0.7846	0.7899	0.7727	0.7699	0.7712	0.7699

Table 2. Comparative Performance Metrics of Classification Models

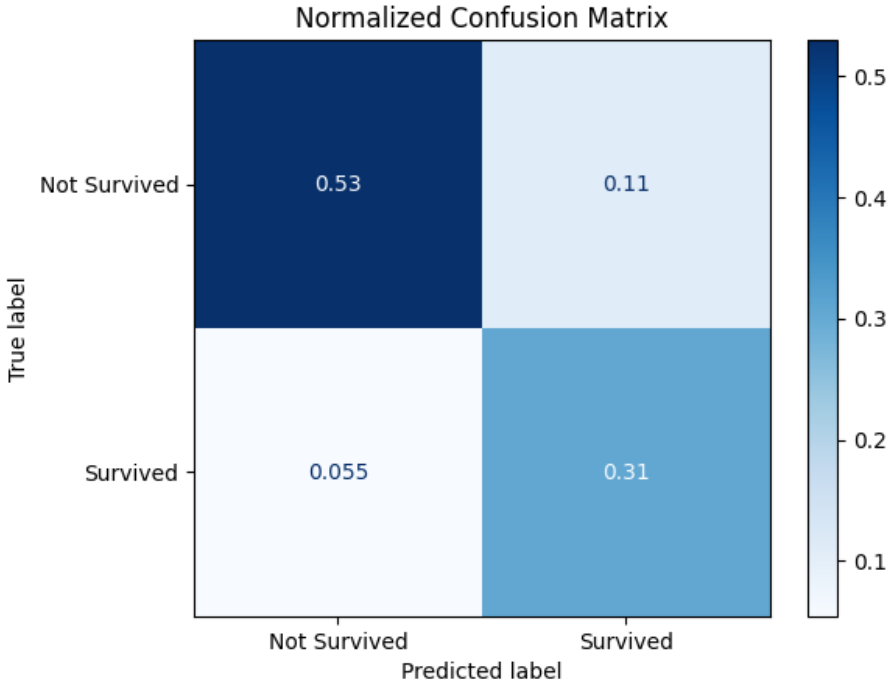

Figure 7. Confusion Matrix

3.3.3. Discussion

In the analysis of the Titanic survival predictions three models were used namely - RandomForestClassifier, DecisionTreeClassifier, and LogisticRegression. When comparing these models, the RandomForestClassifier achieved a good balance between accuracy (0.8319) and F1 Score (0.8160) for the 'survived' class but the DecisionTreeClassifier outperformed it with a slightly higher F1 Score (0.8276) and better precision. The LogisticRegression model lagged slightly behind, with lower overall accuracy (0.7899) and F1 Score (0.7712).

The DecisionTreeClassifier, with its best F1 Score, was the most successful at navigating the complexities of the Titanic dataset. F1 metric was particularly appropriate for the dataset since it accounts for the balance between not only correctly predicting survivors (precision) but also not missing out on actual survivors (recall), which is vital when assessing survival outcomes.

The DecisionTreeClassifier's ability to parse through the data, considering the unique interactions between features like age, class, and familial ties, likely gave it the edge in generating a more nuanced predictive model, as indicated by its superior F1 Score and balanced accuracy.

This confusion matrix for DecisionTreeClassifier shows that the model correctly predicted 'Not Survived' 53% of the time and 'Survived' 31% of the time, with some misclassifications evident.[**??**]

4. Conclusion

In our project, we aimed to predict California housing prices and Titanic passenger survival, using machine learning and data preprocessing for insightful predictions. Initial dataset analysis, cleansing, and feature selection ensured high-quality inputs. The Random Forest Regressor excelled in housing price prediction with its bias-variance balance, while Decision Trees were chosen for Titanic survival for their clarity. Hyperparameter tuning greatly improved performance, with R-squared and F1 Score metrics highlighting our models' effectiveness. Despite successes, we recognize potential enhancements, such as incorporating more diverse data and advanced techniques for sharper predictions in future work.