COMP1816 - Machine Learning Coursework Report

Rifath Ahmed-001187671
Word Count: 1992

1. Introduction

Regression and classification techniques are used to predict house prices and Titanic passenger survival. Classification algorithms included Logistic Regression, Gradient Boosting, and Support Vector Machine. Gradient Boosting had a competitive 0.82 accuracy and 0.89 precision. SVM performed well with 0.85 accuracy and 0.85 precision. Linear Regression, Decision Tree, and Random Forest were used for regression. Random Forest again outperformed the others with an R-squared value of 0.7237, demonstrating enhanced prediction. These findings demonstrate Random Forest's classification and regression efficacy.

2. Regression

2.1. Pre-processing

The housing dataset includes longitude, latitude, median age, rooms, bedrooms, population, households, median income, and ocean vicinity. The goal variable is the median house value. The models' datasets were split into the first 800 train points and the last 200 test points.

Preprocessing Steps:

Handling Missing Values: We used median values to hold missing values in numerical features. This strategy resists outliers and replaces missing values with a data distribution representative value.

- Imputation of missing values using the median involves replacing missing values (NaN) in numerical features with the median (median(X)) of the respective feature:

$$\text{Imputed Value} = \text{median}(X)$$

Feature Scaling (Standardization):

Standardisation gives numerical features a mean of 0 and a standard deviation of 1. Linear regression and support vector machines, which use distance metrics or gradient descent optimisation, must be standardised.

- Standardization transforms the numerical features X by subtracting the mean (μ) and then dividing by the standard deviation (σ) to ensure that the features have a mean of 0 and a standard deviation of 1:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

One-Hot Encoding:

This transformation is needed since most machine learning algorithms only accept numerical input and cannot handle categorical variables. Ocean proximity is one-hot.

- One-hot encoding converts categorical variables into a binary format where a binary feature represents each category. If there are N categories, each category is transformed into a binary vector of length N with all zeros except for one at the index corresponding to the category:

$$\text{Category A} \rightarrow [1,0,0,\ldots,0]$$

$$\text{Category B} \rightarrow [0,1,0,\ldots,0]$$

$$\text{Category C} \rightarrow [0,0,1,\ldots,0]$$

...

2.2. Methodology

Main Model: RandomForest

RandomForest leverages multiple decision trees to mitigate overfitting and enhance generalisation. Each tree, denoted as T_i, is constructed through:

1. Bootstrap Sampling: Random selection with replacement creates bootstrap samples D_i from the original dataset D, ensuring diversity.
2. Random Feature Selection: At each node of T_i, a random subset of m features out of p total features is considered for splitting.
3. Splitting Criterion: The mean squared error (MSE) is typically used to determine the best split at each node, promoting tree growth.
4. Aggregation of Predictions: The final prediction $\hat{y}$ is obtained by averaging predictions from all trees:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i(x)$$

Where N is the number of trees, RandomForest's ensemble approach enhances robustness and predictive performance.

These equations explain Random Forest Regression's bootstrap sampling, random feature selection, splitting criterion, and prediction aggregation. They show how Random Forest Regression uses several decision trees to create reliable regression predictions.

Reasons for Choosing Random Forest over Decision Tree and Linear Regression

Robustness: Random Forest is robust to outliers and noisy data due to its ensemble nature.

Automatic Feature Selection: Random Forest performs feature selection by considering random feature subsets at each node.

Reduced Overfitting: Random Forest mitigates overfitting by combining multiple weak learners (decision trees).

Handling High-Dimensional Data: Random Forest efficiently handles high-dimensional data, unlike Decision Tree and Linear Regression, which may suffer from the curse of dimensionality.

2.3. Experiments

2.3.1. Experimental Settings

Model Architecture and Hyperparameters:

Baseline Models: A main model was previously mentioned, and two baseline regression models were chosen: Linear Regression (no architecture parameters) and Decision Tree Regression (max depth and min split are hyperparameters).

A linear regression model models the connection between a dependent variable (target) and one or more independent variables. It implies features and targets are linearly related.

Equation:

For a simple linear regression with one independent variable, the equation can be represented as:

$$y = \beta_0 + \beta_1 x + \epsilon$$

Where:

- y is the predicted value (dependent variable),
- x is the independent variable (feature),
- β_0 is the y-intercept (bias term),
- β_1 is the coefficient of the independent variable,
- ϵ is the error term.

Linear regression aims to find the best-fitting line that minimises the sum of squared differences between the observed and predicted values.

Decision Tree Regression:

Decision Rule: Splits data at each node based on a chosen feature and threshold. Splitting Criterion: Minimizes impurity, e.g., Gini impurity for classification, MSE for regression. Recursive Partitioning: Continues splitting recursively until a stopping criterion is met. Leaf Node Prediction: Predicts using majority class for classification and mean/median for regression. Overall Prediction: Traverses nodes to reach a leaf for making predictions.

Hyperparameter Tuning: GridSearchCV was used to find optimal values for max depth, min samples split, and n estimators for Decision Tree and Random Forest models. No Hyperparameter tune was used for Linear Regression.

2.3.2. Results

Evaluation Metrics:

R-squared (coefficient of determination) and Root Mean Squared Error (RMSE) are used to assess model performance. These metrics provide insights into accuracy and generalisation capability, aiding in model comparison and selecting the best-performing model.

Model	R-squared	RMSE
Linear Regression	0.6721	63969.21
Decision Tree	0.3963	86799.52
Random Forest	0.7237	58717.79

Table 1. Performance Metrics of Regression Models

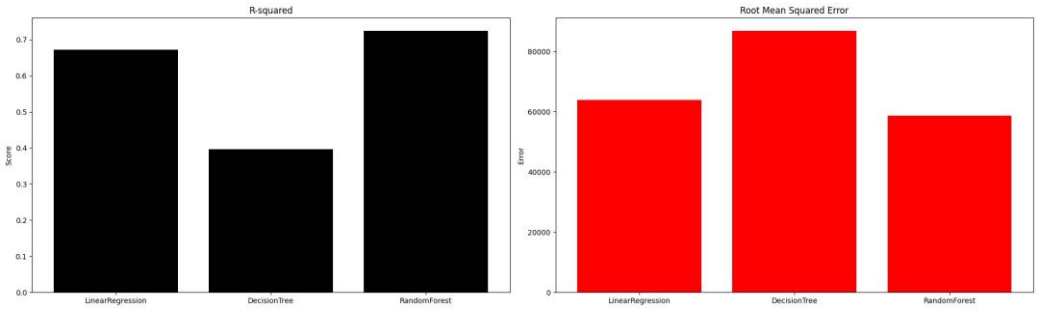

Figure 1. Bar plot of the evaluation matrices

Evaluation Metrics with Equations and Explanations for choosing

R-squared:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where: y_i is the actual value for sample i $\hat{y}_i$ is the predicted value for sample i by the model, $\bar{y}$ is the mean of the actual values, n is the number of samples

R-squared measures the proportion of variance in the target variable (house price) explained by the model. A value of 1 indicates perfect fit, while 0 signifies no explanation by the model. It is suitable for comparing models on the same dataset, but its interpretation needs to be more accurate in some instances.

R-squared: Offers a comprehensive measure of model adequacy, aiding in comparing models' performance on identical datasets.

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE squares the differences between actual and predicted values before averaging, giving more weight to more significant errors. It is also interpretable in the same units as the target variable and penalises models for significant prediction errors more heavily than MAE.

RMSE strikes a balance between the interpretability of MAE and the sensitivity to significant errors, providing a well-rounded evaluation of model efficacy.

2.3.3. Discussion

Analysis of regression models shows performance differences. Linear Regression has a moderate R-squared score of 0.6721 but a large Root Mean Squared Error (RMSE), showing forecast disparities. Decision Tree, with a lower R-squared value of 0.3963, makes even more errors, signifying less accurate predictions than Linear Regression. Random Forest performs best, with the highest R-squared score of 0.7237. This indicates higher predictive power, with Random Forest having the lowest RMSE of the three models. Compared to Linear Regression and Decision Tree models, Random Forest performs best in accuracy and prediction.

3. Classification

3.1. Pre-processing

The dataset details Titanic passengers: age, gender, ticket fare, embarkation port, family size, and survival rates. Features taken from the original data include passengers travelling alone or with family, age, and fare classes. The dataset is extensively used for predictive modelling and machine learning, especially for forecasting survival outcomes using passenger features. The first 700 train data points and the last 190 test data points were split.

Handling Missing Values: Imputing missing values with median age and fare preserves data distribution while replacing missing embarked ports with the most common ensures reasonable estimation.

- Replace missing values in the ”Age”, ”Fare”, and ”Embarked” columns with the median age, median fare, and most common embarked port, respectively.

Encoding Categorical Variables:

Converting categorical variables like ”Sex” and ”Embarked” into numerical representations allows machine learning algorithms to process them effectively.

- Convert categorical variables like ”Sex” and ”Embarked” into numerical representations, such as mapping ”male” to 0 and ”female” to 1 for ”Sex” and ”C” to 0, ”Q” to 1, and ”S” to 2 for ”Embarked”.

Feature Engineering:

Dropping irrelevant columns (”Name,” PassengerId,” Ticket,” etc.) simplifies the dataset, mitigating overfitting. Creating ”FamilySize” and ”IsAlone” features enriches insights, enhancing predictive capability. Binning ”Age” and ”Fare” aids in capturing non-linear relationships and diminishes outlier effects.

- Dropping irrelevant columns (”Name”, ”PassengerId”, ”Ticket”) simplifies the dataset, mitigating overfitting.
- Creating ”FamilySize” and ”IsAlone” features enriches insights, enhancing predictive capability.
- Binning ”Age” and ”Fare” aids in capturing non-linear relationships and diminishes outlier effects.

3.2. Methodology

Gradient Boosting as Main Model:

Gradient Boosting is an ensemble learning method that combines the predictions of multiple weak learners (typically decision trees) to create a robust predictive model. It builds the model sequentially, where each new weak learner is trained to correct the errors made by the existing ensemble.

Suppose we have a training dataset with input features $\mathbf{X}$ and corresponding target labels $\mathbf{y}$. The goal is to learn a function $F(\mathbf{x})$ approximating the true relationship between $\mathbf{X}$ and $\mathbf{y}$.

Objective Function

The objective is to minimise a loss function $L(\mathbf{y}, F(\mathbf{x}))$, which measures the difference between the true labels $\mathbf{y}$ and the predicted values $F(\mathbf{x})$.

$$\text{Objective:} \quad \min_{F} \sum_{i=1}^{N} L(y_i, F(\mathbf{x}_i))$$

Gradient Descent

Gradient Boosting uses gradient descent to minimise the loss function. At each iteration, it fits a weak learner to the negative gradient of the loss function concerning the predictions of the existing ensemble.

$$\text{Residuals:} \quad r_i = -\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}$$

Weak Learners (Decision Trees)

Gradient Boosting typically uses decision trees as weak learners. Each tree is trained to predict the residuals from the previous iteration.

$$\text{Tree Model:} \quad T(\mathbf{x};\Theta) \approx r$$

Model Update

The predictions of the weak learner are added to the ensemble with a small learning rate (shrinkage parameter) ν, which controls the contribution of each tree to the final prediction.

$$\text{Update:} \qquad F^{(m)}(\mathbf{x}) = F^{(m-1)}(\mathbf{x}) + \nu T(\mathbf{x};\Theta_m)$$

.

Advantages of Gradient Boosting over the Baseline models

Gradient Boosting outperforms Linear Regression and Support Vector Machine due to its flexibility in capturing complex relationships, robustness to overfitting, and ensemble learning approach. Its ability to handle intricate patterns in the data while mitigating the risk of overfitting results in higher predictive accuracy. Additionally, the ensemble nature of Gradient Boosting allows it to leverage the strengths of multiple weak learners, enhancing its overall performance compared to the other models.

3.3. Experiments

3.3.1. Experimental Settings

Baseline Models: We used two Baseline models, Logistic Regression and SVM, for our experimental settings.

Logistic Regression:

Logistic Regression models the probability of a binary outcome using the logistic function:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Where $h_\theta(x)$ is the predicted probability, x is the input features, and θ are the model parameters. It minimises the logistic loss function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log\Big(h_\theta(x^{(i)})\Big) + (1-y^{(i)})\log\Big(1-h_\theta(x^{(i)})\Big)]$$

By adjusting θ using optimisation techniques like gradient descent. The decision boundary is defined by $\theta^T x = 0$, separating instances of different classes.

Support Vector Machine (SVM):

SVM finds the optimal hyperplane to separate data classes, maximising the margin between them. The hyperplane, defined as $\mathbf{w^T x} + b = 0$, is determined by $\mathbf{w}$ (average vector), $\mathbf{x}$ (input features), and b (bias term). Using the kernel trick, SVM handles non-linearly separable data by mapping features into a higher-dimensional space. The optimisation problem aims to maximise the margin while minimising classification error, and it is often solved using quadratic programming.

Hyperparameter Tuning with GridSearch: C balances data fit and model simplicity in logistic regression. For gradient boosting, key hyperparameters are $n_estimators$, $learning_rate$, and max_depth. SVM's C and γ manage margin, error, and boundary complexity. We optimised these using grid search, exploring values like $[0.001, 0.01, 0.1, 1, 10, 100]$ for C and others accordingly. Employing k-fold cross-validation ensured robust hyperparameter selection, aiming to maximise accuracy on the training data for enhanced predictive performance.

3.3.2. Results

Evaluation Metrics:

The performance of each classifier is evaluated using various metrics, including Accuracy and Precision.

Classifier	Accuracy	Precision
Logistic Regression	0.82	0.84
Gradient Boosting	0.82	0.89
Support Vector Machine	0.85	0.85

Table 2. Classifier Performance Metrics

Comparing the graph of the classification models

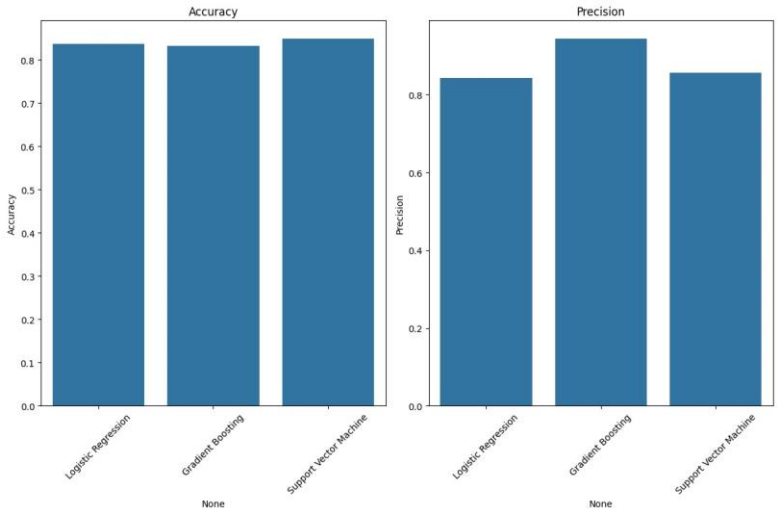

Figure 2. Comparison of Classification evaluation matrices for the three classifiers

We considered precision as the main evaluation metric for our model. So, Precision score is essential here.

Accuracy:

Mathematical Expression:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy measures the proportion of correct predictions (both true positives and negatives) of the total number of predictions. It provides an overall assessment of the model's performance.

Precision:

Mathematical Expression:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision measures the proportion of true predictions out of all positive predictions made by the model. It indicates the model's ability to avoid false positives.

Accuracy and precision are crucial for evaluating the Titanic survivor prediction model. Precision evaluates the model's optimistic forecasts' reliability when false positives have profound effects, while accuracy measures their correctness. Considering these indicators, we can assess the model's performance and capacity to produce accurate and dependable predictions, enabling informed decision-making.

3.3.3. Discussion

Gradient Boosting had the highest Titanic survival prediction precision (0.89), reducing false positives. With precision matching accuracy (0.82), logistic regression performed well. SVM has the highest accuracy but poorer precision (0.85), suggesting higher false positives. Gradient Boosting was chosen for its accuracy and precision.

4. Conclusion

The study used regression to predict house prices and Titanic survival classification. Ensemble approaches like Random Forest, and Gradient Boosting outperforming baseline models highlighted the relevance of feature engineering and model selection in predicting accuracy. Data quality, quantity, feature incompleteness, and computing resources are restrictions. Obtaining more datasets, adding features, and enhancing model interpretability and generalisation are possible future goals.