COMP1816 - Machine Learning Coursework Report

Iustin-Andrei Moisa-Tudor - 001228763
Word Count: 2600

1. Introduction

This report presents the analysis of two different machine learning tasks: regression and classification. For the regression task, the California Housing dataset provided is utilized to predict housing prices through the implementation of three models: Ridge Regression (main model), Linear Regression (baseline), and Lasso Regression(2nd baseline).

For the classification task, the Titanic dataset is analyzed to predict passenger survival using Decision Tree Classifier (main model), Support Vector Machine (SVM) as baseline, and Neural Network (2nd baseline). Both tasks involve preprocessing, methodological development, experimental validation, hyperparameter experimentation, and performance evaluations using metrics such as MSE and R2 and accuracy scores for classification.

The regression task achieved the best performance with the Ridge Regression Model with an R^2 score of 0.608. The classification task performed best with the Decision Tree Classifier, with a final an accuracy of 87.14% for the classification task out of the 3 chosen models.

The results demonstrate how effective certain regularization techniques are in regression tasks and decision tree-based methods in handling classification problems with mixed data types shown in the Titanic Dataset

2. Regression

2.1. Pre-processing

The California Housing dataset contains housing information across various districts in California, including features like longitude, latitude, housing median age, total rooms, bedrooms, population, households, median income, and ocean proximity.

The data was pre-processed by starting with handling missing values through dropna() function to ensure data quality and minimise possible errors during training. Categorical variables, found in the 'ocean_proximity' collumn in the dataset, were converted to numerical values (0-3) to make it easier for the model to understand and simplify the model creation.

A couple of extra features were created to potentially make the model performance more accurate:

- 'roomsHousehold' represents the total rooms divided by households, representing average rooms per house
- 'bedrooms_Room' represents the total bedrooms divided by total rooms, indicating bedroom proportion
- 'population_House' represents the population divided by households, showing average household size
- 'median_income_squared' represents the squared median income to capture non-linear relationships
- 'housing_median_age_squared' represents thesquared house age to account for non-linear age effects

Log transformations were applied to highly skewed distributions to normalize them, the main distributions being log_total_rooms, log_total_bedrooms, log_population and log_households.

The dataset was split into training (80%) and testing (20%) sets, with the last 190 data points reserved for testing as per the coursework requirements. Features were standardized using StandardScaler to ensure all features contributed equally to the model training process and were properly scaled, which is important when dealing with regularized regression models.

2.2. Methodology

Ridge Regression was selected as the main model for this task due to its effectiveness in handling multicollinearity and because it did perform the best out of all the models, which is common in housing datasets where features can be highly correlated. Unlike Linear Regression, Ridge Regression applies L2 regularization, which prevents overfitting of the data extracted from the dataset during training by penalizing large coefficients.

Ridge Regression minimizes the following function:

$$\mathbf{J}(\beta) = ||\mathbf{y} - \mathbf{X}\beta||^{\mathbf{2}} + \alpha||\beta||^{\mathbf{2}} \tag{1}$$

- Function Items:
  - y is the target variable also known as the Y or output (median house values)
  - X is the feature matrix
  - β is the coefficient vector
  - α is the regularization
  - $||\beta||^2$ is the L2 norm of the coefficient vector which is commonly used in ridge regresion models

The first term represents the ordinary least squares error, while the second term is the actual regularization component that penalizes the existing large coefficients. The hyperparameter α controls the strength of the L2 regularization applied. When $\alpha = 0$, Ridge Regression becomes equivalent to Linear Regression; as α increases, the model becomes more constrained, reducing the risk of overfitting the data to the model which will increase it's overall performance.

2.3. Experiments

2.3.1. Experimental Settings

The 3 regression models were evaluated using GridSearchCV which is a popular way of experimenting with different hyperparameter valuea and quite a couple of combinations of parameters to find the best values for the model. This was also done with with 5-fold cross-validation to optimize the hyperparameters effectively.

For the linear regression model:

- The hyperparameters tested are fit_intercept (True/False), positive (True/False)
- The best configuration is fit_intercept=True, positive=False

For the lasso regression:

- The hyperparameters tested are: alpha values [0.0001, 0.001, 0.01, 0.1, 0.9, 0.12, 1.0] - between 0 and 1 at different values.
- Maximum iterations of 10,000
- Tolerance of 0.001
- Best alpha found by the model and GridSearchCV is 1.0

For the ridge Regression:

- The hyperparameters tested are: alpha values [0.001, 0.01, 0.1, 1.0]
- Best alpha found is also 1.0

All models were trained on the standardized training data (X_train) and evaluated on the standardized test set to ensure fair comparison between the hyperparameter options and experiments.

2.3.2. RESULTS

The metrics that were used for evaluation of the models is Mean Squared Error which was chosen as the primary evaluation metric because it directly measures the average squared difference between predicted and actual house values in the dataset (y) and MSE is mostly suitable for regression tasks as it penalizes larger errors more heavily.

The second metric used is the R^2 score which was used to assess the proportion of variance in the dependent variable that is predictable from the independent variables.

The results in the table below represent the values that were obrained for the MSE and R2 score for each of the regression models created:

Model	MSE	R^2 Score
Linear Regression	5.77×10^9	0.5505
Lasso Regression	5.76×10^9	0.5515
Ridge Regression	5.04×10^9	0.6077

Table 1. Model Performance Comparison

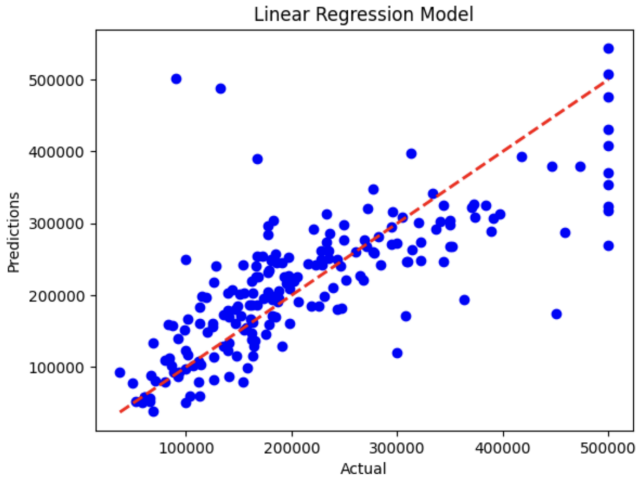

Figure 1. Linear Regression Model

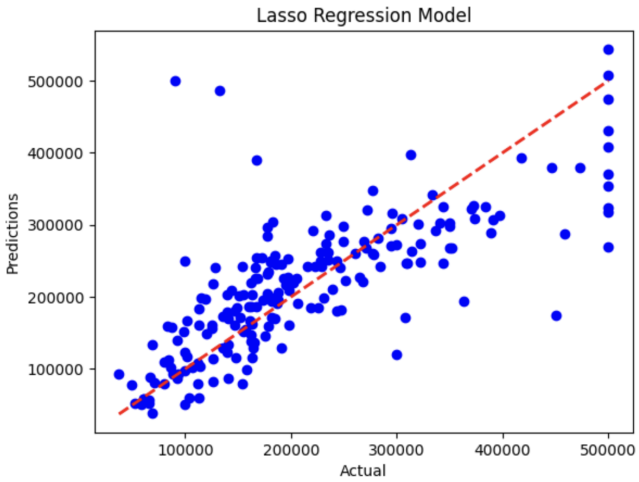

Figure 2. Lasso Regression Model

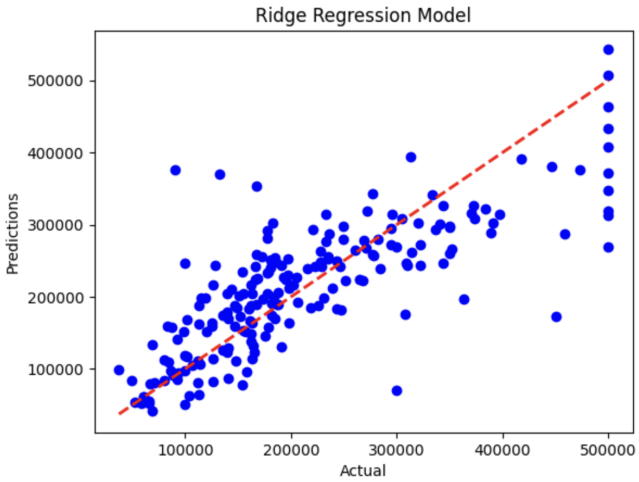

Figure 3. Ridge Regression Model

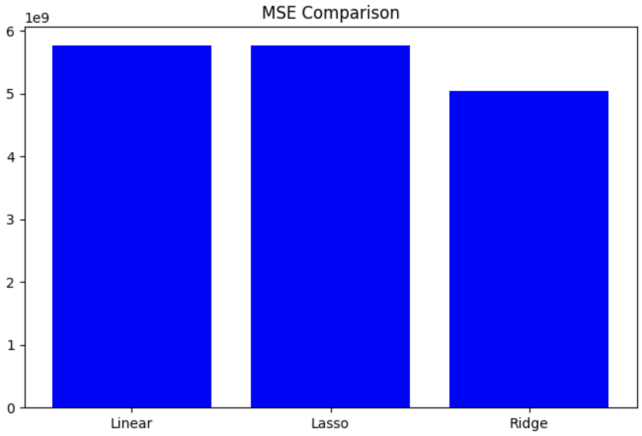

Figure 4. MSE Comparison

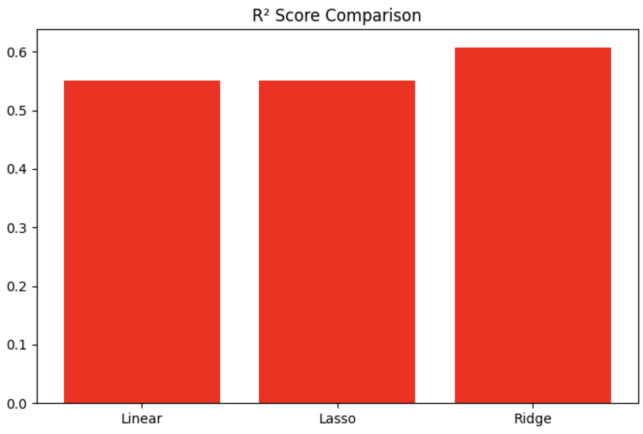

Figure 5. R2 Comparison

The scatter plots of actual versus predicted values for each model are shown in Figures 1-3, and the MSE and R2 comparison is visualized in Figure 4 and 5 respectively.

2.3.3. DISCUSSION

As seen in the results above, the main Ridge Regression model outperformed both baseline models with the lowest MSE and highest R^2 score (0.6077). This was influenced by the following factors: 1. The California Housing dataset likely contains correlated features (e.g., total rooms and total bedrooms). Ridge Regression gets rid of this issue by shrinking correlated coefficients toward each other. 2. While Lasso Regression can completely eliminate features by setting their coefficients to zero, Ridge maintains all features with reduced weights. For this housing dataset, retaining all features with appropriate weighting proved more effective than feature selection. 3. Ridge Regression tends to be more stable than Linear Regression when dealing with high-dimensional data

The Lasso model performed slightly better than the linear model but still worse than Ridge, suggesting that even if regularization is good for the model the L1 penalty of lasso regression might have been too aggressive. The optimal alpha value of 1.0 for both Ridge and Lasso indicates that regularization was necessary to improve prediction accuracy.

The scatter plots (Figures 1-3) show similar patterns across all models, with predictions aligning reasonably well. But Ridge Regression shows tighter clustering around the line.

3. Classification

3.1. Pre-processing

The Titanic dataset provided contains passenger information from the Titanic ship, including categories/features such as passenger class, sex, age, number of siblings/spouses aboard, number of parents/children aboard, fare, embarkation, ticket number, and survival status with the target variable being survival 0 meaning they did not survive or 1 meaning they survived.

The first thing that was done during the preprocessing was to deal with any missing values:

- Age and Fare were filled with their column means
- Pclass was filled with the mode (most frequent value)
- SibSp which represents siblings/spouded, Parch which represents parents/children, and Survival was filling the missing values with 0
- Embarked or not were filled with the mode

Like the regression models, some extra feature engineering steps were implemented so that the model's accuracy will increase and offer a better performance, the features are:

- FamilySize - which was created by summing SibSp and Parch plus 1 , where the 1 meant the actual passenger itself
- FarePerPerson - which was calculated by dividing the Fare by FamilySize
- Sex - was simply converted to binary (0, 1) using LabelEncoder so either 0 or 1 represent male or female
- Embarked collumn was transformed using one-hot encoding to create binary columns for each port (C, Q, S)

The feature set created included the following new features: 'Pclass', 'Sex', 'Age', 'Fare', 'FamilySize', 'FarePerPerson', 'Embarked_C', 'Embarked_Q', and 'Embarked_S'.

The last 140 data points were reserved for testing, with the remainder used for training as requested in hte requirements.

StandardScaler was applied to standardize the features, ensuring they all had a mean of 0 and standard deviation of 1. Which is beneficial in developing SVMs or Neural Network models in general.

3.2. Methodology

Out of all 3 chosen model types, the **Decision Tree** was selected as the main model due to its ability to handle both numerical and categorical data, and it's effective way of capturing non-linear relationships.

The Decision Tree model works by recursively partitioning the feature space to create regions with homogeneous target values.

For each node, the algorithm selects the feature and threshold that maximizes information gain. Specifically for the classification, this is typically measured using Gini or entropy. Gini is measung the impurity.

The general entropy formula which was also used for the node is:

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2}$$

Formula Items:

- S is the set of samples at the node
- c is the number of classes
- p_i is the proportion of samples belonging to class i

The first baseline model for the classification task is **Support Vector Machine** SVM, it's a type of classifier that finds an optimal hyperplane to separate classes. The mathematical function for an SVM is:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{3}$$

The 2nd baseline model for the classification task is a **neural network classifier**, which in a multi-layer perceptron with two hidden Dense layers of 32 and 16 neurons and dropout regularization. The activation function for hidden layers was ReLU which is best for classification as it the most popular used and it turns negative values to 0s so the model will work with positive values only and increase performance during training. The formula for the ReLu activation is:

$$f(\mathbf{x}) = \max(0, x) \tag{4}$$

And the output Dense layer containing a single neuron with the information passed from the larger neurons used the sigmoid activation as generally, ReLu isn't the optimum activation function for the last layer. The formula for the sigmoid function is as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

3.3. Experiments

3.3.1. Experimental Settings

Like the regression models, the 3 classifier models went through hyperparameter tuning using GridSearchCV with 5-fold cross-validation to experiment with different parameter values and mix them up to check which one is the best to use for this specific model.

Decision Tree Classifier:

- Hyperparameters tested:

  - max_depth: 2, 4, 6, 8, 10
  - min_samples_split: 2, 5, 10
  - criterion: ['entropy', 'gini']
- Best hyperparameters found by the model for Decision Tree are criterion='entropy', max_depth=4, min_samples_split=2

Support Vector Machine:

- Hyperparameters tested:
  - C: [0.1, 1, 10]
  - kernel: ['rbf', 'linear']
  - gamma: ['scale', 'auto']
- Best hyperparameters found by the model for SVM are C=10, gamma='scale', kernel='rbf'

Neural Network:

- Sequential with layers of 32 neurons and 16 neurons with dropout 0.2 from each neuron to prevent overfitting
- Optimizer: Adam with learning rate=0.001
- Loss function: Binary cross-entropy
- Early stopping was implemented with patience=10 epochs, monitoring validation loss
- Batch size: 32

3.3.2. RESULTS

The evaluation metric for all 3 of the classification models was accuracy because it provides a straightforward measure of the proportion of correct predictions. Apart from accuracy, precision, recall, and F1-score were also implemented to gain more insights on the class performance (0 or 1) in binary classification.

0 represents passengers who did not survive, while 1 represents survivors.

The performance metrics for each of the 3 models are shown in the table below:

Model	**Accuracy**	**Precision (Class 0)**	**Recall (Class 0)**	**Precision (Class 1)**	**Recall (Class 1)**
Decision Tree	0.8714	0.90	0.90	0.82	0.82
SVM	0.8357	0.82	0.96	0.89	0.62
Neural Network	0.8286	0.84	0.90	0.80	0.70

Table 2. Classification Models Performance

Out of all 3 models, the Decision tree classifier has received the highest accuracy score of 87.14 percent while SVM has a lower accuracy of 83 percent and the neural network 82 percent.

3.3.3. DISCUSSION

Out of all the models for the classification task, the Decision Tree classifier model outperformed both baseline models with an accuracy of 87.14%, compared to 83.57% for SVM and 82.86% for the Neural Network.

This might be because of several factors of which some are:

1. It achieved high and balanced precision and recall for both classes (0.90/0.90 for non-survivors) and (0.82/0.82 for survivors), showing it performed fairly well across both categories without bias toward either class

2. Decision Trees naturally model feature interactions. In this case, the interaction between passenger, gender, and age classes was probably crucial in determining the chances of survival for the passengers.
3. The model's optimal depth of 4 suggests that a relatively simple tree structure was sufficient to capture the key patterns in the data

This way it avoided overfitting of the training data and achieved a higher accuracy in the end.

The baseline SVM model showed an accuracy of 83.57% with good recall for non-survivors (0.96) but poor recall for survivors (0.62) which means the model was more conservative in predicting survival.

The 2nd baseline Neural Network model performed slightly worse than the SVM, with an accuracy of 82.86%. Its performance was more balanced between classes than the SVM but still showed some preference for correctly identifying non-survivors. This might also be because of the low number of layers it was given at the start.

The best hyperparameters for the Decision Tree chosen by GridSearchCV (eg: entropy criterion, max_depth=4, min_samples_split=2) indicate that the model benefited from moderate complexity constraints.

4. Conclusion

In this report, 3 regression models and 3 classification models were analysed which used the California Housing and Titanic datasets provided.

For the regression task, Ridge Regression turned out to be the best performind model out of the 3, outperforming Linear and Lasso Regression.

This demonstrates the effectiveness of L2 regularization in handling and preventing overfitting in housing price prediction.

For the classification task, the Decision Tree Classifier achieved the highest accuracy (87.14%) with balanced performance across both survival classes. This generally shown the effectiveness of decision tree-based methods in capturing complex relationships in mixed data.

A couple of limitations were identified throughout the development of these models. In the regression task, all models showed reduced accuracy for extremely high-priced houses, suggesting that additional features or non-linear transformations might be needed to better capture luxury housing data or higher priced ones too.

Potential improvements could include:

1. Experimenting with Random Forest or Gradient Boosting for both regression and classification tasks
2. Exploring more deep learning architectures with transfer learning specifically for the classification task.

Overall, the tests and development show insights on the development of the 6 models, how the models compare and what their performance is.