COMP1816 - Machine Learning Coursework Report

Phakonekham Phichit - 001041931
Word Count: 1813

1. Introduction

This report discusses two different types of machine-learning implementations which are regression and classification. Each implementation provides two different models to predict the appropriate data sets. Regression is used to predict housing prices using the California Housing data set. Whereas, classification is applied to predict the survivability factors of the passenger onboard the Titanic based on the Titanic data set.

The regression implementation consists of using Linear Regression as the baseline model and Lasso Regression as the main model. The result of the R^2 score of both models is 66%. Due to the use of different and large quantities between the data points and the chosen variable, this makes a poor correlation for the R^2 score. Lasso Regression slightly performed better due to its ability to select and adjust fine-tuning parameters.

As for the classification implementation, the baseline model is Logistic Regression, and the main model is SVM (Support Vector Machine). Logistic Regression receives 81% accuracy and 71% recall. Whereas, SVM outputs better results with 83% accuracy and 74% recall. SVM features to experiment with different kernels output better performances than linear model kernels. The most optimum kernel for SVM that is applied in this implementation is Radial Basis Function(RBF) Kernel.

2. Regression

2.1. Pre-processing

The regression model implementation used the California Housing data set through the required CSV file. Within the data set, there are 1022 records. 801 records are applied for the training data set and 221 records for testing. The following table below shows the feature and details of each record.

Features	**Details**
No.	Number ID
longitude	Longitude of the area
latitude	Latitude of the area
`housing_median_age`	Median age of houses
`total_rooms`	Total number of the rooms per house
`total_bedrooms`	Total number of bedrooms per house
population	Population of the area
households	Number of household
`median_income`	Median income of the tenants
`median_house_value`	Median value of the house
`ocean_proximity`	The proximity levels to the ocean

Table 1. California Housing Dataset

There were missing records in the `total_bedrooms` column. These missing records can be deleted or ratio the total bedroom to total rooms and multiplied by the `total_rooms` records. However, the deletion method of the record was chosen as there was less correlation of the missing values between different data points.

Ocean Proximity	Encoded Value
INLAND	0
<1H OCEAN	1
NEAR OCEAN	2
NEAR BAY	3

Table 2. Numerical Encoding Ocean Proximity

There is also an additional pre-processing method that involves numerical encoding from the `ocean_proximity` column which only accepts numerical inputs. However, there was no correlation between both models within the data set in households and housing median age.

2.2. Methodology

2.3. Linear Regression

Linear regression is used for predicting a continuous numerical output based on one or more input features. It models the relationship between the input features and the output using a linear equation. This includes median income and medium house value. As well as total rooms and median house value which has a strong correlation.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{1}$$

where y is the output variable, x_1 to x_p are the input features, β_0 to β_p are the coefficients (slopes) that represent the effect of each input feature on the output, and ϵ is the error term. The goal of linear regression is to estimate the values of the coefficients that minimize the sum of squared errors between the predicted output and the actual output. Additionally fine-tuning is not required.

2.4. Lasso Regression

Lasso regression is a linear regression method that adds a penalty term to the sum of squared errors, in order to encourage the coefficients to be small and sparse (i.e., some of them are exactly zero). The penalty term is the L1 norm of the coefficients

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2}$$

where λ is the hyperparameter that controls the strength of the penalty (the higher λ, the more the coefficients are shrunk towards zero).

Compared to linear regression, Lasso regression has the advantage of automatically selecting a subset of the most important input features, which can lead to simpler and more interpretable models. However, Lasso regression can also have the disadvantage of being unstable when the input features are highly correlated, as it tends to select only one feature among a group of correlated features. In this case, there is no correlation with median house value which is appropriate for using this model.

2.5. Experiments

2.5.1. Experimental Settings

Google Colab is used for this implementation with **Python 3.9**. The following imported libraries: **numpy, pandas, sklearn, matplotlib, sns** and **google.colab**. Both models are imported from **sklearn**. Linear Regression is not required to use any hyperparameters. Lasso regression used L1 Penalty regulariser which includes the **alpha** hyperparameter. **Lasso**, from **sklearn**, was used with the following experimental parameters that were to find the optimum value that was tuned for **alpha** : (0.001, 0.01, 0.1, 1, 10, 100, 1000). The best value for the alpha coefficient was 1.0

2.5.2. RESULTS

The main metric contributors to finding the appropriate scores are R-squared (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error(MAE). R^2 score is applied to determine the best-fitted data of the model. A higher score from the model predicts more accurate data. RMSE identifies the mean difference between the predicted values and the true values with an additional feature to scale the error margin of the model. Finally, MAE provides output regarding errors that are not affected by outliers with realistic datasets.

Table 3. The Results of Linear Regression

Linear Regression (baseline model)	
R^2	0.6591
RMSE	67292.51
MAE	51304.75

Table 4. The Results for Lasso Regression

Lasso Regression (main model)	
R^2	0.6590
RMSE	67307.61
MAE	51319.04

2.5.3. DISCUSSION

Lasso Regression slightly produced better results in RMSE by 15.1 and MAE metrics by 14.29. Whereas, R^2 has a similar score to Linear Regression. This is due to the results of Lasso regression to identify more useful data based on its features than Linear regression. However, there are no significant changes between both models. Both models are identical in results.

3. Classification

3.1. Pre-processing

The classification implementation of the Titanic data set consists of 893 records, 651 for training, and 242 for testing. The data set includes data from the passengers onboard the Titanic and the survivability rate of the passengers. The following table below contains the information on the dataset.

Features	**Details**
PassengerId	Number ID
Pclass	Ticket Class
Name	Name of the Passenger
Sex	Sex/Gender of the Passenger
Age	Age of the Passenger
SibSp	Number of Siblings/Spouse
Parch	Number of Parents/Children
Ticket	Ticket Number
Fare	Fare amount of the Passenger
Embarked	Embarked Port from S, Q, C
Target: Survived	Value (0 or 1) if the passenger survived

Table 5. Titanic Datset

In the data set of the Titanic, there was a significantly larger scale of missing records within the `Age`column. Instead of deleting the records from the previous model implementation. The missing age records are added based on the average age of the total passengers in the data set which is 29.

The irrelevant features such as PassengerID, Name, and Ticket will be ignored as it has no correlation within the dataset. Fare and Pclass also have no relevance as economic incentives or factors do not determine the passenger survivability rate abroad on the Titanic.

Additionally, similar to the previous task, there is also a numerical encoding data set which is the `Embarked` column.

Embarked	**Encoded Value**
S	0
Q	1
C	2

Table 6. Numerical Encoding Embarked Port

Finally, for the last required numerical encoding dataset, the `Sex` column is also needed to be encoded into numerical data in order to scale and normalized the data appropriately.

Sex	**Encoded Value**
Male	0
Female	1

Table 7. Numerical Encoding for Sex

3.2. Methodology

3.3. Logistic Regression

Logistic regression is an algorithm used for predicting a binary output (e.g., yes or no, true or false) based on one or more input features. It models the probability of the binary output using a logistic function that is suitable for the Titanic data set to determine the Targeted survivability of the passenger.

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}} \tag{3}$$

where y is the binary output, $\mathbf{x}$ is the input feature vector, $\mathbf{w}$ is the weight vector that represents the effect of each input feature on the log-odds of the binary output, and e is the mathematical constant approximately equal to 2.71828. The goal of logistic regression is to estimate the values of the weight vector that maximize the likelihood of the observed binary outputs, given the input features and the assumed distribution of the errors.

3.4. Support Vector Machine (SVM)

Support Vector Machine (SVM) is an algorithm that can be used for both binary and multi-class classification tasks. SVMs work by finding the hyperplane that maximizes the margin between the closest data points of different classes. The decision boundary of an SVM is determined by a subset of the training data points called support vectors, which lie closest to the hyperplane. The hyperplane can be linear or non-linear, depending on the choice of kernel function.

3.5. Experiments

3.5.1. Experimental Settings

Google Colab is used for this implementation with **Python 3.9**. The following imported libraries: **numpy, pandas, sklearn, matplotlib** and **google.colab**. Similarly with the same environment as the previous task.

Table 8. Hyperparameter settings for Logistic Regression

Hyperparameter	Settings
C	0.01,0.1,1.0,10,100
solver	'newton'-cg, 'ibfgs', "liblinear'
penalty	'l2'

Logistic Regression hyperparameters are used within the build-in function of sklearn. The optimum parameters are: solver='liblinear', C=1.0, and penalty='l2'. There are different solvers such as 'cg' and 'ibfgs'. However, it does not result in better accuracy or recall scores.

Table 9. Hyperparameter settings for the support vector machine (SVM)

Hyperparameter	Settings
Kernel coefficient γ	0.01
Kernel	RBF
C	0.01,0.1,1.0,10,100,1000

Similarly with SVM hyperparameters are used within the build-in function of sklearn. The optimum parameters are: kernel='rbf', C=100, gamma=0.01

3.5.2. RESULTS

Accuracy and Recall are the metrics for the classification implementation. A measure of accuracy compares the percentage of correctly classified instances to all instances in the dataset. It is a simple metric that offers a general assessment of a classifier's performance. However, when the dataset is unbalanced and one class is significantly more prevalent than the other, accuracy can be deceiving. In these circumstances, even though it does not offer useful predictions for the minority class, a classifier that consistently predicts the majority class can still achieve high accuracy.

Recall, on the other hand, is a metric that measures the proportion of correctly classified instances of a specific class over the total number of instances that belong to that class. It is a useful metric for imbalanced datasets, as it captures the ability of a classifier to correctly identify instances of the minority class. In the context of SVMs, recall can be used to evaluate the performance of the classifier on positive instances that are important to the problem at hand.

Table 10. Accuracy and Recall Results

Model	Accuracy	Recal
Logistic Regression	0.8101	0.7101
Support Vector Machine (SVM)	0.8324	0.7391

3.5.3. DISCUSSION

Based on the results, SVM algorithm outperformed logistic regression by 2% in accuracy and 3% by recall. This is by SVM ability to collect nonlinear relationships within the data set where it requires more complexity. SVM hyperplanes are more intelligent to identify different dimensions of features in a large dataset as it is able to visualize non-linear and linear kernel functions. Whereas, Logistic Regression is not as flexible as SVM and only could determine data linearly. Logistic regression is a probabilistic model that can output the confidence or certainty of its predictions, while SVMs are binary classifiers that only output the predicted class.

4. Conclusion

In the regression task, there were no significant changes in both model. There were missing records in both training and testing datasets. All the metrics scores of R^2, RMSE, and MAE were identical in Linear Regression and Lasso Regression. Overall, Lasso Regression performed better slightly than Linear Regression. For further implementation, instead of deleting

records within the dataset, it is more advised to identify an average or ratio and filled in the data depending on the dataset and correlation of the features within the dataset. In addition, experimenting with more models such as Ridge Regression is beneficial to determine on what is the best-fitted model for the dataset.

As for the classification task, there were more missing datasets in the Titanic file. However, it is solved by including the average in the missing column. It is clear SVM algorithm outperformed Logistic Regression by 2% in accuracy and 3% in recall scores due to the SVM layers of complexity of the model. SVM had multiple aspects that made it more complex, whereas Logistic Regression was easier to interpret but less flexible. The SVM algorithm is much suited for small datasets such as the Titanic which could be problematic for neutral networks but it is a possible implementation. For further recommendation, use the decision tree for classification to identify features and decision rules within the Titanic dataset.