Machine Learning Coursework Report

Fashion MNIST

The fashion MNIST image dataset, which includes a training set of 60,000 examples and a test set of 10,000 examples, will be used in the first and second tasks. Each example was a 28x28 grayscale image, which means that each example has a feature dimension of 784, and all examples were divided into 10 categories.

1. Analysis fashion-MNIST

1.1 PCA

Principal Component Analysis is a dimensionality reduction method based on an orthogonal transformation that transforms potentially linearly correlated data into linearly uncorrelated data, also known as principal components. The new reduced dimensional data can then be used to characterise all data in a smaller dimension. PCA will project the original data into a two-dimensional space, using the first and second principal components as axes (Figure 1).

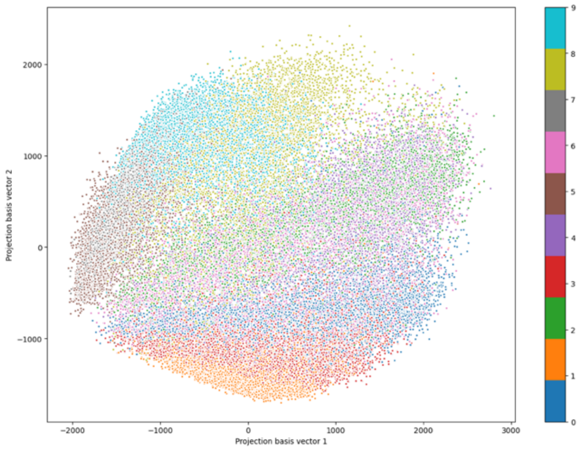


Figure 1: The scatter plot for first and second principal component

PCA is finding the projection with the highest variance. Formula 1 is used to calculate the covariance matrix of the original data. N is the number of data points, x_n is each data point and $\bar{x}$ is the mean of the data.

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T \quad (1)$$

The values on the diagonal of the covariance matrix are the variances of the individual data dimensions. Since $Su_1 = \lambda_1 u_1$ and $u_1^T S u_1 = \lambda_1$, the eigenvector and eigenvalue of the covariance matrix can be found. The first biggest eigenvector is the first principal component, and the second biggest eigenvector is the second principal component which is orthogonal to the first principal component.

Q1. The first principal component explained variance is about 12288132.61 and the second principal component explained variance is about 787596.49. And the variance ratio, which represents the ratio of the variance value of each principal component after dimensionality reduction to the total variance value, is approximately 29% and 17.8% for the first and second principal components respectively, indicating that these two principal components can be used as an important main component to describe the data feature, but the sum of their ratios is insufficient to represent all features in the data, which may lead to information loss.

Q2. Figure 1 illustrates the data projection into the first and second principal component spaces following dimension reduction, with each colour representing a distinct class. When examining the images, there are approximately 10 clusters, and the data points are concentrated, which minimizes the effect of extreme points on model learning. However, the boundaries between clusters are not particularly distinct. This also proves the last answer that, when the variance ratio of the selected principal components is insufficient, the attributes of original data points cannot be accurately preserved in the reduced-dimensionality data.

1.2 Gaussian Mixture Model

Q1. A Gaussian mixture model is a probabilistic model that assumes all data points are derived from a finite number of Gaussian distributions, as demonstrated by Equation 2.

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{2}$$

In the equation, K represents the number of Gaussian distributions and π_k represents the mixing coefficient for each Gaussian distribution, which is $\sum_{k=1}^{K} \pi_k = 1$. Moreover, every single Gaussian distribution has a mean and covariance matrix, which corresponds to a cluster. When the Gaussian mixture model is applied in the cluster task, the process of clustering data using the GMM is the inverse process of generating data samples using the GMM. Given the number of clusters K and the data set, the parameters of each Gaussian distribution are derived by using Maximum Likelihood Estimation (MLE), the log-likelihood equation is shown in equation 3.

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\right\} \tag{3}$$

When deriving the parameter for each Gaussian distribution, the MLE cannot be directly calculated. Therefore, the Expectation-Maximization (EM) algorithm will be used. This algorithm consists of two steps, E-step and M-step. E-step is used to estimate data from which distribution, which is the posterior probability of each sample being generated by each component, the responsibilities will be computed based on the current parameter for Gaussian distribution by using Bayes theory, the formula is defined by:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \tag{4}$$

After that, the M-step will estimate the parameter by using the calculated responsibilities from the last E-step. The formula for calculating each parameter is:

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{5}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \tag{6}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{7}$$

Repeat steps E and M until the likelihood function converges, or the maximum iterations are achieved. Figure 2 shows the clustering outcome of the Gaussian mixture model after PCA. The left is the Gaussian distribution computed from real data classification, while the right side

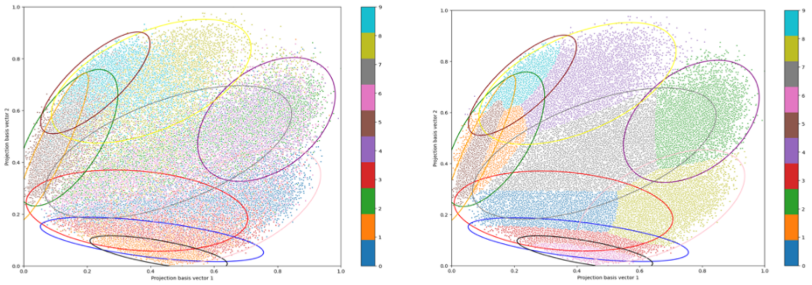


Figure 2: The Gaussian mixture model result

is the clustering obtained by the model according to the similarity of the data.

Q2. Figure 2 shows 10 Gaussian distributions that separate the data into ten clusters. When training a GMM model, the true classification of the data is not imported into the model, but the model score on the training data set and test data set are 0.696 and 0.607, indicating a correlation between the true classic label and the cluster. In the bottom left corner of the graph, the true classification of the data is roughly consistent with the cluster, whereas the top right corner is not precisely represented by the clustering, possibly because the distinction between those classes is not immediately obvious, and the PCA could lose information.

2. Classifiers

2.1 Artificial Neural Network

Artificial Neural Network (ANN) has three different types of layers, which are the input layer, output layer and hidden layer. The input layer receives and transmits the data to the hidden layer. The hidden layer is located between the input and output layers and is responsible for data processing and learning. The output layer outputs the results of the hidden layer as the result for the whole model. An ANN neural network can have only one input and output layer, but it can contain one or more hidden layers, known as the multiple-layer perceptron. In this task, a neural network model with 64 hidden layers will be utilised to classify the MNIST dataset. In addition, the MNIST dataset is normalised using the min-max scaler function, which can make the training model more precise and efficient.

Q1. The learning curves for training and validation are shown in Figure 3. Red represents the accuracy of the model on the training set and the green represents the accuracy of the model on the test set. Overall, the model is useful for analysing this dataset and does not overfit because the scores on the validation and training sets are at least 0.8 and do not differ significantly from one another, and the scores improve as the size of the sample pool increases.

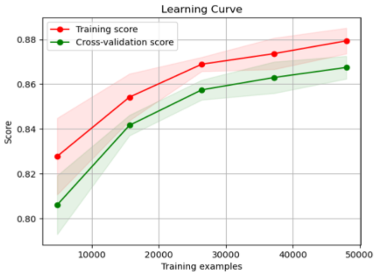
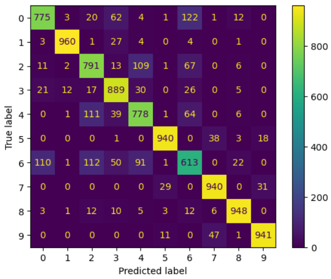


Figure 3: The learning curve for ANN

Q2. After training the ANN model, the test dataset will be used to test the accuracy of the model. The score for the test data set is 0.86 and the confusion matrix is shown in Figure 4. Each column of the confusion matrix represents the predicted category, and each row represents the true attribution category of the data. The numeric values in each column represent the number of real data that were predicted to be of that class. Figure 4 shows that the model overall classification is accurate because the number on the diagonal line is the greatest. Except for the diagonal, the rest of the region represents the number of samples confounded by the model



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	775	3	20	62	4	1	122	1	12	0
1	3	960	1	27	4	0	4	0	1	0
2	11	2	791	13	109	1	67	0	6	0
3	21	12	17	889	30	0	26	0	5	0
4	0	1	111	39	778	1	64	0	6	0
5	0	0	0	1	0	940	0	38	3	18
6	110	1	112	50	91	1	613	0	22	0
7	0	0	0	0	0	29	0	940	0	31
8	3	1	12	10	5	3	12	6	948	0
9	0	0	0	0	0	11	0	47	1	941

Figure 4: Confusion matrix for test

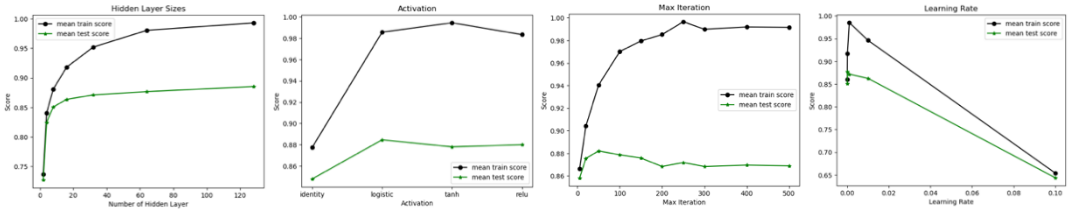


Figure 5: The model score for different hyperparameters

between the class, which shows that some categories are not well defined.

Q3. The size of the hidden layers, the activation function, the maximum number of iterations, and the learning rate typically influence ANN models. Figure 5 shows the correlation between these parameters and the score on the train and test sets. When the number of hidden layers increases, the performance of model classification gradually improved, but when it reaches a certain amount, the growth will slowly slow down. The activation function is used to incorporate nonlinear factors, as the linear model if not sufficiently, so that the data can be better classified. Compared with setting the parameters of the model to identity, which means that do not use the activation function, the model with activation function has higher accuracy on the test and train set. Setting the max iteration can control the complexity of the model. With Figure 5 discovery, more iterations will make the model has higher accuracy. However, if the max iteration is not set appropriately and the model is trained more complicated, the performance on the test set will not have much improvement or even decrease, which means that the model is easily overfitted. The learning rate controls the step size of the model learning, the curve in Figure 5 demonstrates that if the learning rate is set excessively large, the accuracy of the model will be negatively impacted because the model might miss the optimal solution.

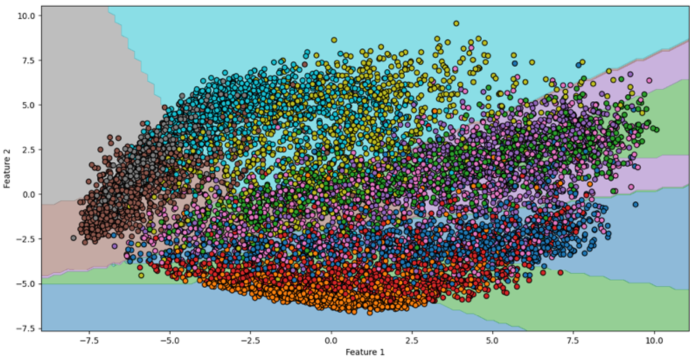
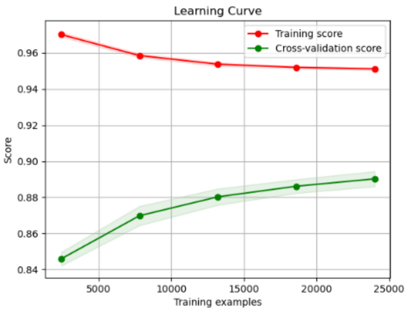
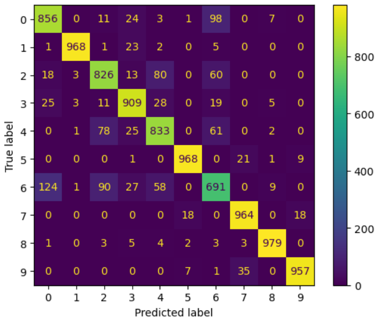


Figure 6: Decision Boundary for ANN Model

Q4. In this section, PCA reduced the data to two dimensions and ANN classifies it. Figure 6 displays the decision boundary under the first and second principal components, which illustrate the real test data on the decision boundary of the ANN model. The shape of the decision boundary for the ANN is linear, which means the region is separated by some straight lines. In addition, the area separated by the decision boundary is roughly the same as the real classification of the test data, but the point away from the centre of the real category are not correctly classified by the decision boundary line.

2.2 Support Vector Machine





True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	856	0	11	24	3	1	98	0	7	0
1	1	968	1	23	2	0	5	0	0	0
2	18	3	826	13	80	0	60	0	0	0
3	25	3	11	909	28	0	19	0	5	0
4	0	1	78	25	833	0	61	0	2	0
5	0	0	0	1	0	968	0	21	1	9
6	124	1	90	27	58	0	691	0	9	0
7	0	0	0	0	0	18	0	964	0	18
8	1	0	3	5	4	2	3	3	979	0
9	0	0	0	0	0	7	1	35	0	957

Figure 7: Learning Curve for SVM

Figure 8: Confusion Matrix for SVM result

Support vector machines can be applied in linear or nonlinear classification. SVM uses a kernel function, it transfers the input vector into a high-dimensional feature space and obtains the optimal classification plane. The categorization plane separates as many data points as feasible appropriately and keeps them away from it.

Q1. In Figure 7, the learning curve of SVM shows the relationship between the training examples and model score on training and test score. At the beginning of the curve, it can be clearly seen that when the sample size is small, the trend of the training curve and the performance are significantly higher than the validation set, which proves that the model is at risk of overfitting currently. After increasing the sample size, the situation improved, and the curves of the training set and validation set gradually approached.

Q2. The SVM model achieved a score of 0.9 in the test set, indicating its strong performance. The numbers on the diagonal of the confusion matrix in Figure 8 show that the SVM was able to correctly classify most of the test data. However, by examining the matrix, the SVM has the most error in the sixth category and perform best in the first and eighth categories.

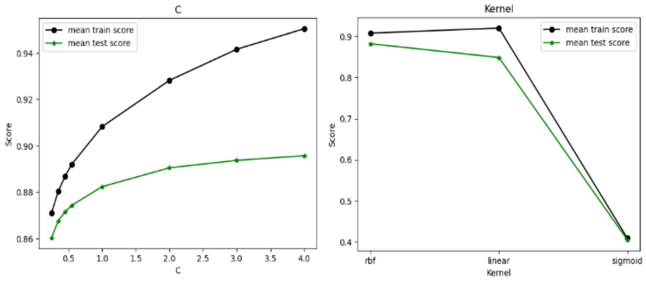


Figure 9: The model score for different hyperparameters

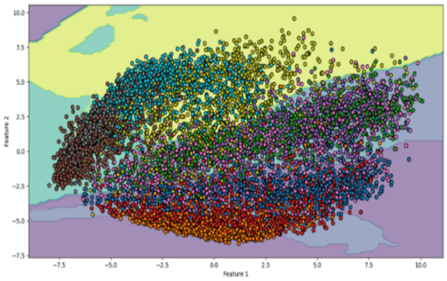


Figure 10: The decision boundary for SVM

Q3. In the SVM model, the kernel function is a key part of the SVM model because it determines the distribution of data in high-dimensional space and the shape of the decision boundary, thus it has a great impact on the performance of the model. In Figure 9, there are three different kernel functions were compared, among which RBF performed better than sigmoid and linear as kernel function in the SVM model. The reason for this is that it will measure the similarity between samples and create a space that describes the similarity, which allows similar samples to be better clustered and then linearly separated. The regularization parameter C in an SVM model determines the strength of the regularization, which affects the distance between the support vector and the decision plane. A small value of C corresponds to a strong regularization, which means stricter classification, while a large value of C corresponds to a weak regularization, which means greater error tolerance. As shown in Figure 9, the relationship between the regularization parameter C and the model score indicates that a higher value of C leads to improved performance of the SVM model.

Q4. Figure 10 illustrates the decision boundary for the SVM model, which uses the first and second principal components as axes. The decision boundary is nonlinear, as indicated by the multiple curving lines dividing the feature space into sections corresponding to the different classes. The nonlinear decision boundary can effectively handle a wide variety of classification tasks, allowing for more toleration of differences between the categories.

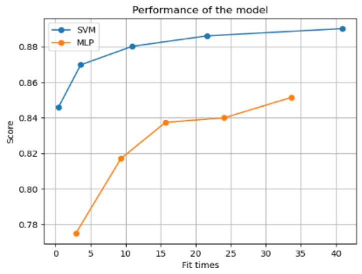


Figure 11: The fitting time for SVM

Q5. By comparing the learning curves (Figure 3 & 7), confusion matrices (Figure 4 & 8), decision boundary (Figure 6 & 10) and fitting time (Figure 11) of ANN and

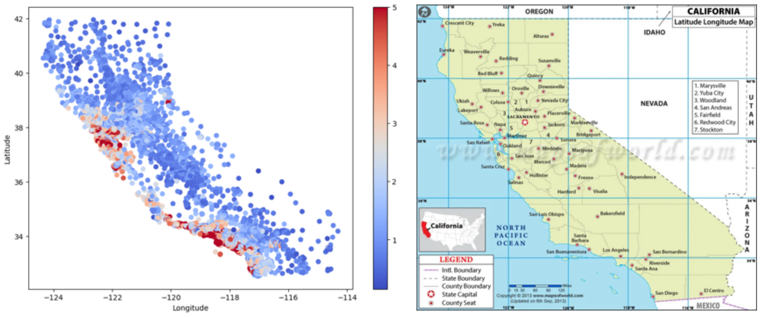
SVM, both can be used as an effective classifier for the MNIST dataset. The linear decision boundary of an artificial neural network (ANN) is fast and efficient, but it may not be suitable for complex, nonlinear relationships in the data and may not be as accurate as a support vector machine (SVM). In comparison, a nonlinear decision boundary may be more flexible, but it may require longer training times and be more subject to overfitting when the sample size is small. This can be observed in the learning curve for the model. In summary, SVM is the better model, although it requires longer training time, it has higher accuracy.

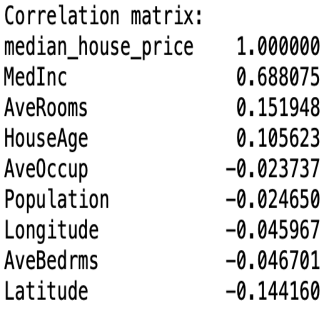
California housing price dataset

The California housing price dataset will be used in the next two sections. The dataset is directly called from sklearn. This data set has a total of 20640 sample, and each sample has 8 features, so the dimension of this data set is (20640, 8).

3. Bayesian linear regression with PyMC

In Bayesian linear regression, the relationship between the dependent and independent variables is modelled as a linear function, with the parameters of the linear function (such as intercept and slope) being treated as random variables. The prior probabilities of these parameters are specified using probability distributions. Then, MCMC sampling of the observed data is used to compute the posterior probability of the parameters, which can be used to make predictions about the dependent variable given new values of the independent variable.





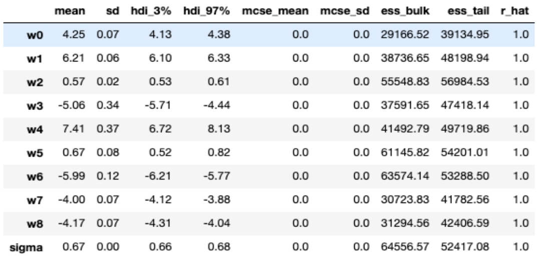
```
Correlation matrix:
median_house_price    1.000000
MedInc                0.688075
AveRooms              0.151948
HouseAge              0.105623
AveOccup             -0.023737
Population           -0.024650
Longitude            -0.045967
AveBedrms            -0.046701
Latitude             -0.144160
```

Figure 12: The relationship between longitude, latitude, and median house price

Q1. The generated graph demonstrates the association between longitude and latitude on median house prices in the California housing dataset. The position of the dot represents the latitude and longitude position, and the colour represents the level of the median house price. Comparing the picture with the map, for example, the house price in coastal areas is higher than that in the inland area. In addition, by calculating the correction matrix of the median house price in the data set, longitude and latitude are negatively correlated with house price, which means that when longitude and latitude decrease, the house median price increase.

Q2. The dataset did not contain any missing values, so the Local Outlier Factor algorithm was used to identify outlier data points. This algorithm compares the local density of each data point with the density of its neighbours and considers a point to be an outlier if its local density is significantly lower than that of its neighbours. In this case, only the surrounding two neighbours are considered, and a total of 1148 outlier data points were removed from the dataset, leaving 19492 data points. After that, because of the different range of values between the feature in the data, normalization was applied to all the features to ensure consistent units and improve the accuracy and efficiency of the model.

Q3. The traces on the right in Figure 13 are the sampled values used to construct the estimated posterior densities on the left. Figure 14 contains statistical information on the distribution, including mean and standard deviation, in this case, the standard deviation of the prior is not significant, which illustrates the low volatility and more concentrated distribution of the data. In the last column in Figure 14, the r_hat quantity shows how similar 4 Markov chains are. In the table, all r value is exactly 1, so this is evidence that four chains have converged.



	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
w0	4.25	0.07	4.13	4.38	0.0	0.0	29166.52	39134.95	1.0
w1	6.21	0.06	6.10	6.33	0.0	0.0	38736.65	48198.94	1.0
w2	0.57	0.02	0.53	0.61	0.0	0.0	55548.83	56984.53	1.0
w3	-5.06	0.34	-5.71	-4.44	0.0	0.0	37591.65	47418.14	1.0
w4	7.41	0.37	6.72	8.13	0.0	0.0	41492.79	49719.86	1.0
w5	0.67	0.08	0.52	0.82	0.0	0.0	61145.82	54201.01	1.0
w6	-5.99	0.12	-6.21	-5.77	0.0	0.0	63574.14	53288.50	1.0
w7	-4.00	0.07	-4.12	-3.88	0.0	0.0	30723.83	41782.56	1.0
w8	-4.17	0.07	-4.31	-4.04	0.0	0.0	31294.56	42406.59	1.0
sigma	0.67	0.00	0.66	0.68	0.0	0.0	64556.57	52417.08	1.0

Figure 14: Summary for all parameters

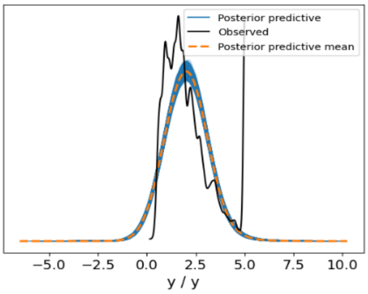


Figure 15: Predictive posterior check

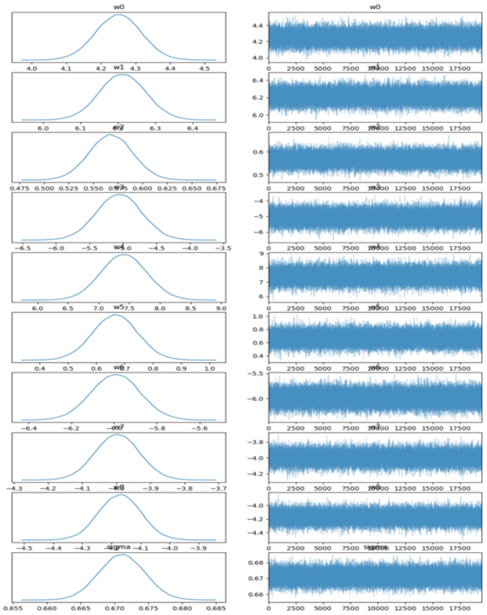


Figure 13: Posterior distribution for all parameters

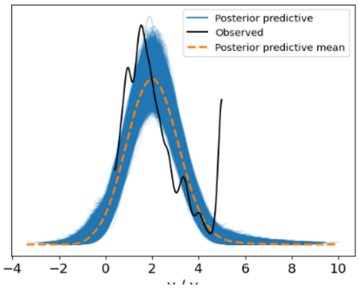


Figure 16: 500 samples posterior distributions

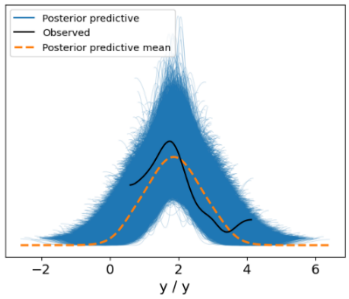


Figure 17: 50 samples posterior distributions

Q4. Figure 15 shows the relationship between the predicted posterior data and the observed data, which can be seen that the predicted posterior is not a perfect fit to the true data, but there is some approximation. By applying the model to the real data, the score is 0.578 and by calculating the mean (2.037), variance (1.234) and standard deviation (1.111) of the posterior sample, the result are very similar to the original data (2.037, 1.243, 1.115), which means the model can capture the distribution of the observed data. Therefore, the model still produced a approximation to the desired posterior distributions.

Q5. Comparing Figures 15, 16 and 17, there is a greater difference in posterior predictive, which is the blue area in the Figure. When the sample size decreases and the number of samples remains the same, the posterior predictive distribution is not accurate in fitting the observed data. Therefore, for the Bayesian linear regressor, if the sample size is large enough, the posterior predictive is more stable.

4. Trees and ensembles

4.1 Decision Tree

Q1. The classification and regression tree is a binary tree, that loops all features and uses a binary segmentation method to divide the data subset so that each node will only split 2 branches. CART uses a greedy algorithm that adds one node at a time. The greedy algorithm will start at the root and perform an exhaustive search for each possible variable and threshold to find a new node. For each search, CART will calculate the average of the target variables to determine a new node for each leaf. Then compute the error if the tree stops adding nodes here. Then choose the carriable and threshold that minimize the error. After that, add a new node for the chosen variable and threshold. After finishing all searches, all data points should associate with each leaf node. When the tree has been constructed, the branches that do not reduce error beyond a small tolerance value should be pruned.

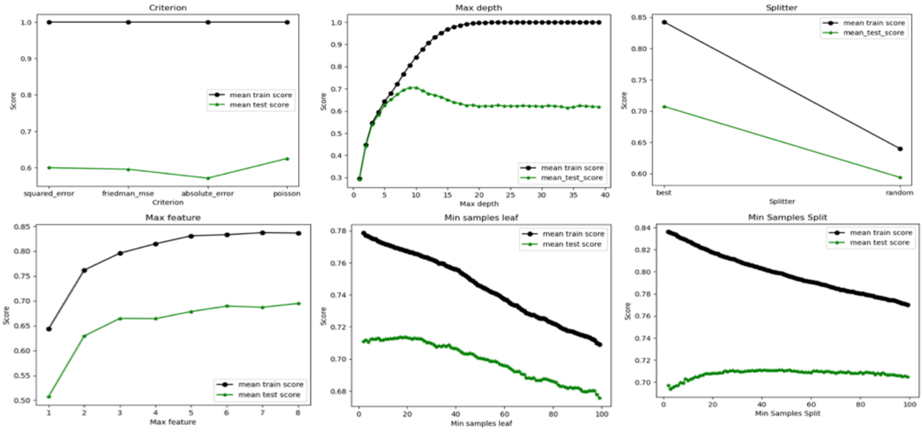


Figure 18: Hyperparameter tuning for decision tree model

Q2. This section, it is compared the training score and test score for different settings of the criterion, max depth, splitter, min sample leaf and min samples split parameters. From Figure 18, all the parameters have some impact on the model except the criterion parameter change which did not have a large impact on the model. Min samples leaf and min samples split are used to set the minimum number of samples needed for the splits and leaf nodes. When these two parameters become larger, the structure of the tree will become simple, which will affect the learning ability of the model. Furthermore, the parameter that has the strongest influence on the model is the max depth. This parameter is applied to control the depth of the decision tree, and any branches that exceed the maximum depth are pruned. If the parameter is set excessively large, the decision tree will split until it reaches its local optimum point, or the split sample number is less than the min sample split parameter. Thus, the tree is more complex and might retain some specific parameters for the training set, which reduces the generalisation ability of the model and easily overfitted. The max depth parameter figure shows that when the depth exceeds 10, there is a clear tendency of overfitting because the gap between the test and train score becomes larger.

Q3. Based on the previous parameter optimisation, all parameters except the splitter influence the training time of the model. For criterion parameters, between the four criterion parameters, using the absolute error method as the calculation method for feature selection requires the most training time. In addition, for the max depth and max feature parameters, as the numbers get larger, the depth of the model and the amount of learning increases, resulting in an increase in training time. However, when max depth is greater than the optimised depth of the model, the training time is not affected. For the min sample split and min sample leaf parameters, as the number of parameters increases, the depth of the tree decreases, and the training time decreases.

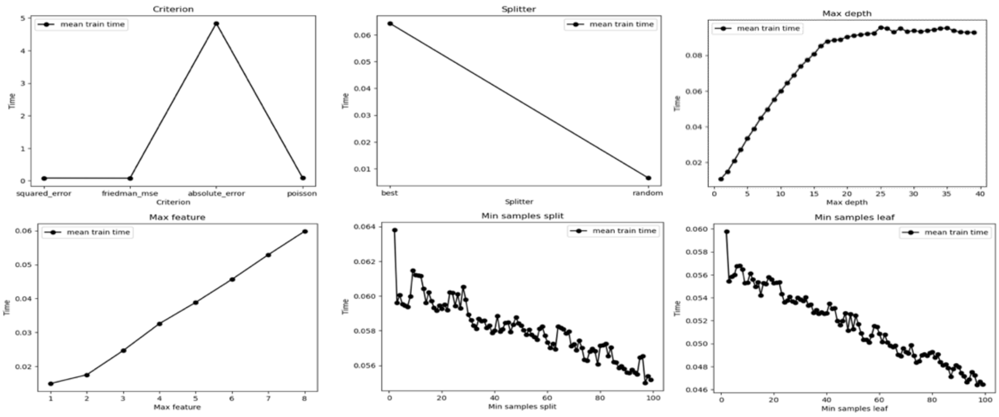


Figure 19: Fit time with different hyperparameter

Q4. After hyperparameter tuning using the cross-validation method, a decision tree model with the best parameters was obtained. In this task, 20% of the normalised original data set was used as the test data set and eighty percent of the data was used as the training data set. When the model was trained using this dataset, the accuracy of the trained model on the test set was 0.729, with mean absolute and mean squared errors of 0.364 and 0.413. Based on the predicted results, Figure 20 captures 300 of the prediction and compares them with the real value, the model is roughly the same in the overall trend, but does not fit some of the excessive house prices.

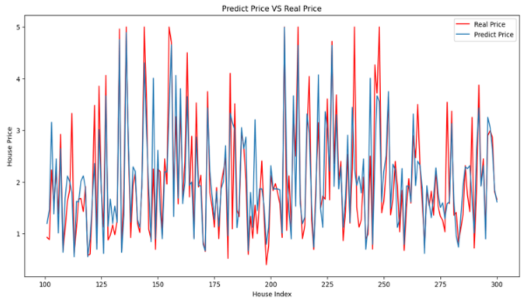


Figure 20: The prediction price curve

Q5. The difference between decision trees and linear regression is that linear regression analyses the overall structure of the data, which attempts to find a linear relationship between the independent and dependent variables. While decision trees analyse the partial structure, which will reduce the impact of extreme values or outliers on the overall model training. Therefore, when dealing with nonlinear related data, or has more complex characteristics, decision trees should perform better than linear regression.

Q6. After visualising the decision tree model, the judgement at each node will use ranges to classify the data, which can easily be overfitted. This is because the model is based on the training set to determine the range at each node, but these ranges might not cover the full test set of data properly, which leads to some points are not classified correctly in the tree model.

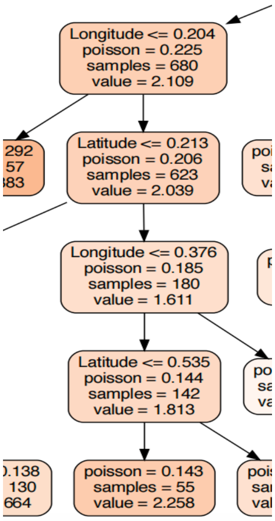


Figure 21: decision tree

4.2 Ensemble method

The ensemble approach combines numerous machine learning models to train an strong estimator to handle restricted challenges. Ensemble approaches include averaging methods such as bagging and random forests, and boosting methods, such as AdaBoost. By comparing the performance of three ensemble methods based on a decision tree, the AdaBoost has the best performance (Figure 22).

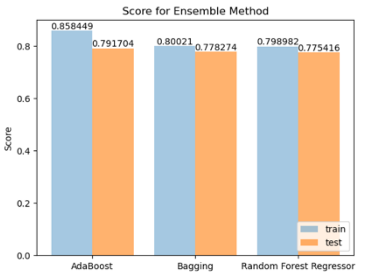


Figure 22: Ensemble method result

Q1. AdaBoost is a supervised learning model within machine learning. The main idea is to train different weak estimators on the same training set, and then combine these weak estimators to create a stronger final estimator. The algorithm itself is implemented by changing the weights of the data, it starts by initialising the weights of each sample in the training data using the same weights. The weights for each sample are then determined based on the correct or incorrect classification of each sample in the training set and the overall accuracy of the previous estimator, which means the weights of samples that have been correctly learned will be reduced, and the weights of samples that have been incorrectly learned will be increased. The new dataset with the modified weights is given to the next estimator for training, and the training process continues in this iterative way. Finally, the estimators from each training process are merged and used as the final strong estimator. By using this method, the estimator can avoid unnecessary training data features and focus on the key training data feature. Using this approach, it is possible to build a strong estimator based on carious weak estimators. The model is trained through data with different weights, which can make the next model focus on the mistakes of the previous weak estimator to increase the diversity of weak estimators. By using a strong estimator composed of various weak estimators, complex data can be better processed.

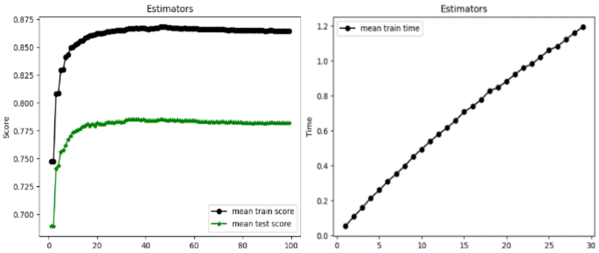


Figure 23: Score and time for different number of

Q2. The number of estimators determines the maximum number of estimators used by the AdaBoost method, and the relationship between the number of estimators and test scores is shown in the figure. When the number is small, the model does not have a good score on both the training and test set, which means the model was under fitted and there are not enough trees in AdaBoost to capture all the data features. As the number increases, the accuracy of the model starts to improve because the diversity of the trees in AdaBoost becomes larger, allowing the strong estimator can accept complex data. However, when the number reaches the best number, this case is 28, the accuracy of model will stop increasing because no more different trees can be obtained. Additionally, increasing the number of estimators also leads to an increase the fit time of the model.

Model	**Ada-Boosting**	**Decision Tree**	**Bayesian Linear**
MAE	**0.400**	**0.400**	**0.489**
MSE	**0.279**	**0.355**	**0.450**
SCORE	**0.792**	**0.738**	**0.634**

Q3. Based on the setting for AdaBoosting in the previous part, comparing the AdaBoosting, Decision Tree and Bayesian Linear regression, the AdaBoosting has better performance, which has the highest score of 0.792 and minimum MAE and MSE (0.400 and 0.279). The reason is that compare with the other two models, AdaBoosting can combine many types of the based estimator and it focuses more on the error data from the last estimator, which can make the final estimator accept more complex data. The decision tree can do regression tasks on nonlinear data, but if the feature of the data is complex, it is easily overfitted. The reason why Bayesian linear regression does not have better performance is that this data is nonlinear, and Bayesian linear regression needs to find the linear relationship between the dependent and independent variable.

Reference

[1]. Bishop, C.M. (2006) Pattern recognition and machine learning by Christopher M. Bishop. New York: Springer Science+Business Media, LLC.

[2]. Malik, V. (2022) California latitude and longitude map: California latitude and longitude, USA States. Available at: https://www.mapsofworld.com/usa/states/california/lat-long.html (Accessed: December 1, 2022).