ST3189 Coursework Project

Table of Contents

UOL SRN: 220455073

1. Data selection

We chose to work on 3 different datasets, one for each given task.
For unsupervised learning, the Wine dataset with varying quantities of 13 constituents found in different types of wines.
For regression, the Body Fat Prediction dataset with fat, density and various body measurements for 252 men.
For classification, the Sold and Rented HDB Residential Units dataset from Singapore's Housing & Development Board (HDB), containing data from 2006 to 2021.

3 distinct datasets were used to ensure that their data fits well for their respective tasks.

2. Unsupervised Learning Task:

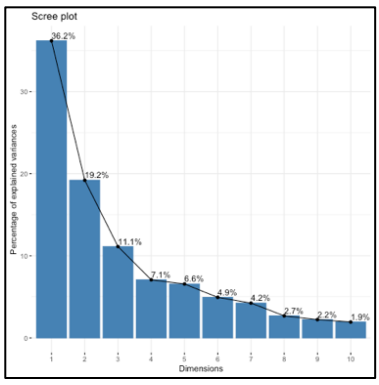
2.1 Substantive Issue

Understanding the factors that differentiate wines can help wine producers and sellers optimise production and improve marketing strategies. Many wines are classified by broad categories like red wine or white wine, but study shows that chemical compositions can influence taste, quality, and pricing.
The Wine dataset contains 13 numerical constituents like Alcohol, Magnesium, and Flavanoids found in different types of wines. We want to see how the data can be explained by principal components and how these attributes are grouped together in terms of correlation.

2.2 Research Questions (RQs)

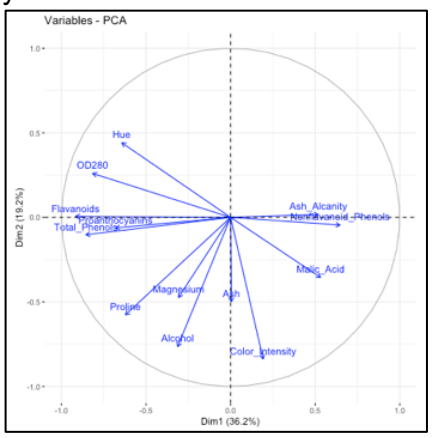
- RQ1: Can we identify distinct clusters of wines based on their chemical properties?
- RQ2: How do different clustering techniques compare in segmenting the wines?

2.3 Principal Component Analysis (PCA) and Correlation Circle



Screeplot

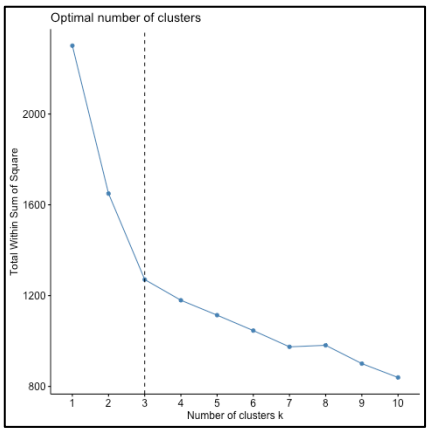
We clean the data for any NA values and duplicated rows, then normalise the data in preparation for PCA and a Correlation Circle.
PCA reduces dimensionality of a dataset by identifying principal components (PCs) that capture the most information, while removing less critical information. From the screeplot on the left, if reduction is necessary or overfitting is encountered, we might want to stop at the 5th PC as 80.2% of the information contained in the data can be explained by the first 5 PCs.



Correlation Circle

The Correlation Circle on the right shows positively correlated features grouped together, while negatively correlated features are positioned on opposing sides of the quadrants.
Distance between features and the origin measures the quality of those features on the factor map. Features like Flavanoids and OD280 that are far from the origin are well represented. It also shows that the first two PCs can explain 55.4% of the variation in the dataset.

2.4 K-means Clustering



WCSS

In this case, we apply K-means clustering technique by starting with the elbow method to find the optimal K, and group features together based on similar characteristics. The within-cluster sum of squares (WCSS) graph on the left assesses how each number of clusters will explain how closely grouped data points are within every cluster. The lesser the number of clusters, the more variation of features in each cluster. The optimal K is found where the rate of decrease in WCSS sharply changes, which is K = 3 in this case.

Using K = 3, we visualise a K-means cluster plot where we can see 3 distinct clusters without any overlapping.

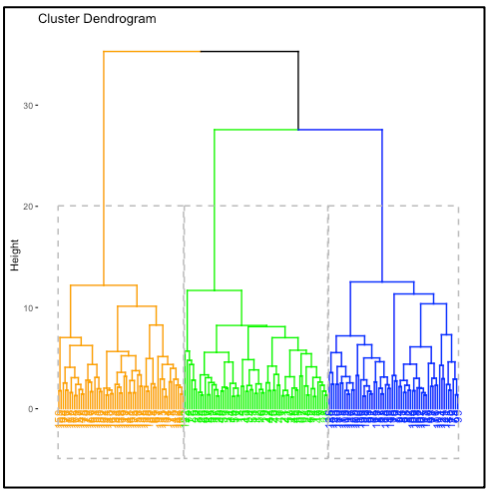
By extracting the clusters, descriptive statistics show us that features like Alchohol, Ash, and Magnesium have the highest mean in Cluster 1, while other features like Malic Acid, Non-flavanoid Phenols, and Color Intensity have the highest mean in Cluster 3.

K-means clusterplot

```
# A tibble: 3 x 14
  Cluster Alcohol Malic_Acid   Ash Ash_Alcanity Magnesium Total_Phenols
    <int>   <dbl>      <dbl> <dbl>        <dbl>     <dbl>          <dbl>
1       1    13.7       2.00  2.47         17.5      108.           2.85
2       2    12.3       1.90  2.23         20.1       92.7          2.25
3       3    13.1       3.31  2.42         21.2       98.7          1.68
  Flavanoids Nonflavanoid_Phenols Proanthocyanins Color_Intensity   Hue
       <dbl>                <dbl>            <dbl>           <dbl> <dbl>
1       3.00                0.292             1.92            5.45 1.07
2       2.05                0.358             1.62            2.97 1.06
3       0.819               0.452             1.15            7.23 0.692
  OD280 Proline
  <dbl>   <dbl>
1  3.16    1100.
2  2.80     510.
3  1.70     619.
```

Statistics (mean)

2.5 Hierarchical Clustering



Dendrogram

Alternatively, we can apply hierarchical clustering to explore natural groupings based on the features.

We compute a distance matrix where every element represents the pairwise distance between data points. The metric used for this is called the Euclidean distance. Using Ward's method to minimise the total within-cluster variance, the clusters formed at each step are as packed and homogeneous as possible. Drawing across a horizontal line at the longest vertical lines will result in 3 intersections, meaning 3 distinct clusters.

2.6 Summary

With 3 distinct clusters of wine properties, RQ1 is addressed. Unsupervised learning techniques help to gather insights on how and what features are grouped together. Without setting a random seed to produce identical results every time, using k-means clustering more than once can create slightly different clusters. Hierarchical clustering builds a hierarchy of clusters, and the number of clusters are chosen based on it. These clustering techniques address RQ2 and can still be informative with varying cluster numbers depending on what you are looking for. We can detect similarities in different types of wine and relate this to sales data to discover which types of wine produces the highest sales.

3. Regression Task

3.1 Substantive Issue

Accurate body fat estimation is vital for public health, fitness, and medical records, but some measurements can be difficult to collect for its computation.

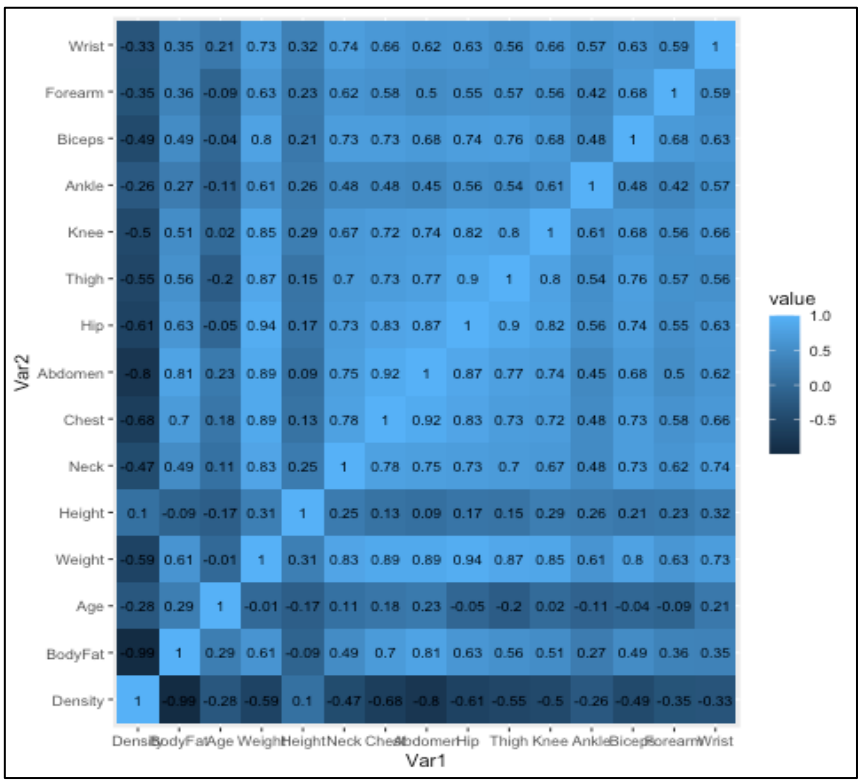
The Body Fat Prediction dataset includes 15 columns of data such as fat, density and other body measurements. Since collecting results for Density is resource intensive and time-consuming, we aim to predict Density, followed by determining Body Fat percentage using Siri's (1956) equation.

3.2 Research Questions (RQs)

- RQ1: How are certain body measurements correlated to each other?
- RQ2: Can a regression model predict body density accurately for its use in the computation of body fat percentage?

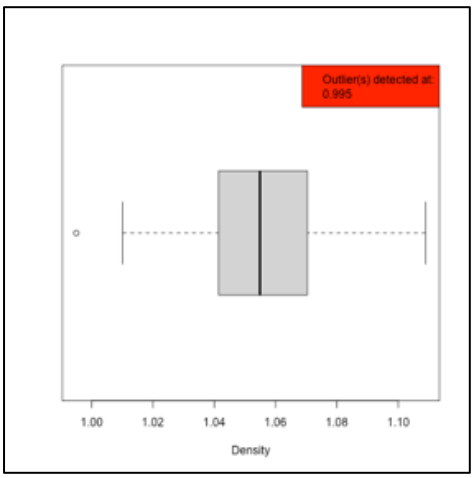
3.3 Correlation Heatmap and Boxplots

We start by cleaning the data and check for skewness where features like Height, Ankle and Hip are more skewed. This shows that outliers are present, and we will identify them later.



Correlation heatmap

We create a correlation heatmap to visualise correlations between features. Multicollinearity is present in the data. Generally, high correlation coefficients of absolute values greater than 0.7 or 0.8 prove this. We aim to create new features using a combination of existing features to reduce multicollinearity problems in the data. Height and Weight features can be used to create a new feature, Body Mass Index (BMI) = [Weight (lbs) / Height (inches)^2] * 703. From the correlation map, Abdomen-Chest and Hip-Thigh have the highest correlation coefficients, at 0.92 and 0.9 respectively. We will create 2 more new features, Abdomen-Chest ratio and Hip-Thigh ratio. Then, we drop features used for the new features created. Since we will determine Body Fat from Siri's equation at the end after predicting Density, we will also remove Body Fat as a feature here.



Boxplot

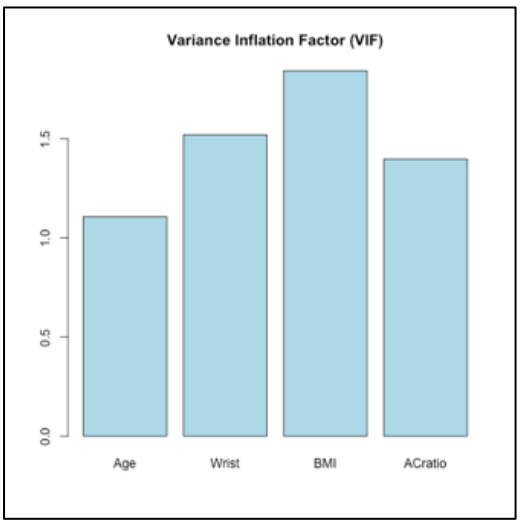
We proceed with plotting boxplots for each feature in the data to identify outliers like the example shown on the left, which shows a boxplot for the Density column and indicates all outliers detected in a red box.

Features like Forearm and BMI have the most outliers at 5 each, while the Age feature has no outliers. Rows with an outlier in any column will be removed for better modeling purposes.

3.4 Backwards Stepwise and Variance Inflation Factor (VIF)

We applied Backward-Stepwise for linear regression based on the resultant dataset to check for the best model – this can change later when we split the data for training and testing. Backward-Stepwise starts with the full model then subsequently deletes the least important predictors based on Akaike Information Criterion (AIC).

The best model was shown to be Density ~ Age + Wrist + BMI + Abdomen-Chest Ratio, with the lowest AIC of -2099.23. The lower the AIC, the better the model. We can also make use of a barplot to compare VIF values, which can indicate the degree of multicollinearity.



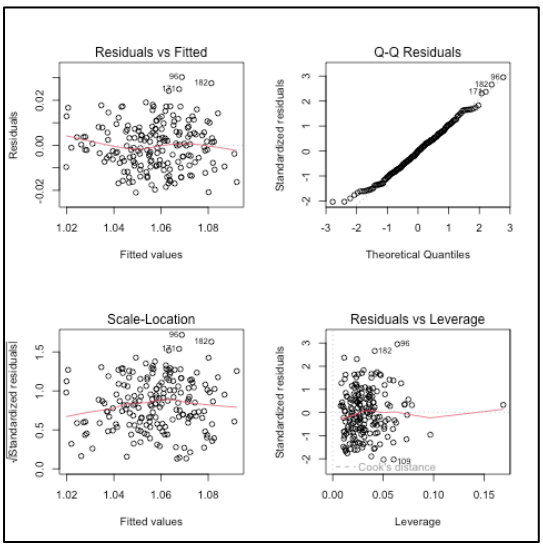
VIF comparison

Since VIF values are between 1 and 5, the predictor features are moderately correlated but not significant enough to require further correction to address multicollinearity.

In the following models, we will use the same split of data, 80% and 20% as training set and testing set respectively. We will begin by training the model based on the training set, then use the testing set to evaluate how well the trained model performs.

3.5 Linear Regression Analysis

We train a Backward-Stepwise linear regression model based on the training set. The Biceps feature is now added compared to the first Backward-Stepwise we did based on the full dataset. This is possibly due to a smaller dataset (training set) lacking enough observations to detect its effect on the model. However, for consistency purposes, we will continue using the trained model: Density ~ Age + Biceps + Wrist + BMI + Abdomen-Chest Ratio, with the lowest AIC of -1667.56. We will test this trained model against the testing set and use root mean squared error (RMSE) as a measure of how accurate a predictive model is. The lower the RMSE, the more accurate the predictions are to the actual data.



Linear Regression Analysis

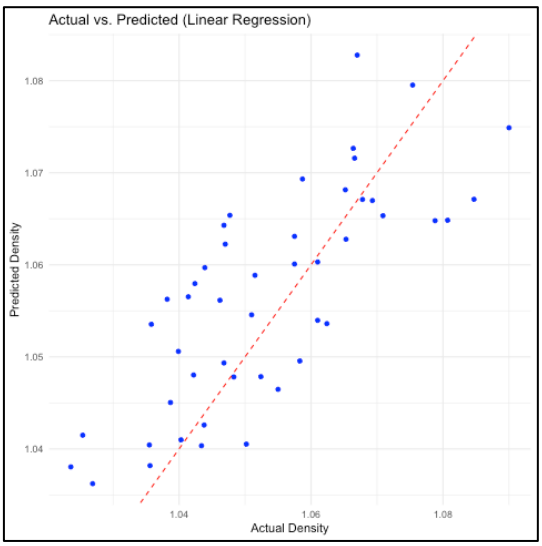
By plotting the trained linear regression model, these 4 graphs will be shown for analysis.

The Residuals vs Fitted graph shows if residuals have non-linear patterns. A non-linear relationship between predictors and the response variable can be identified when the red line is not equally spread around the dotted horizontal line. It appears to be a decent spread on the dotted horizontal line, indicating that we do not have non-linear relationships.

The Q-Q Residuals graph shows if residuals are normally distributed. It appears to be normally distributed from how well the residuals line up on the diagonal dotted line.

The Scale-Location graph shows whether residuals are spread equally over the ranges of predictors. The assumption of homoscedasticity is observed here as the red line appears to be almost horizontal.

The Residuals vs Leverage graph shows influential outliers when the data points are outside the dashed lines. Since Cook's red dashed distance lines cannot be seen here, all points are within the dashed lines, meaning that there are no influential outliers.

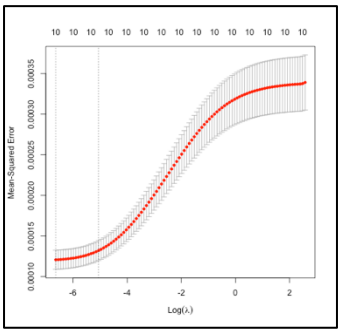


Actual vs. Predicted

The RMSE for Linear Regression using the testing set is 0.0104, which is similar to other reports using the same dataset. By using the above showcased techniques, we were able to address and reduce multicollinearity, a critical problem in regression, to increase the model's accuracy.

Next, we scale the same training and testing sets for other models such as Ridge Regression, Lasso Regression, and Random Forest. For Ridge and Lasso Regression, we cross-validate to find the optimal lambda value, a regularisation parameter that helps to avoid overfitting.

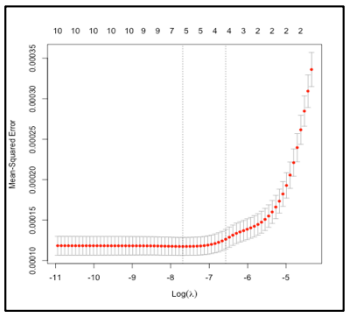
3.6 Ridge and Lasso Regression (Penalties and Cross-validation)



Ridge Penalty

Ridge and Lasso Regression control the estimated coefficients by introducing a penalty parameter. While the ridge penalty forces the features to approximately but not equal to zero, the lasso penalty forces coefficients to zero.

Another notable difference is that ridge does not perform feature selection whereas lasso does auto feature selection. Thus, depending on a need to retain features or not, a choice between ridge and lasso will be obvious.

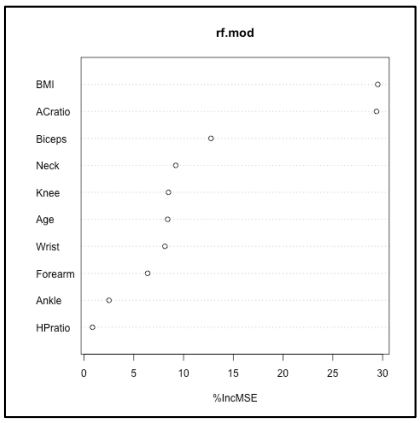


Lasso Penalty

We use 10-fold cross-validation (CV) models for ridge and lasso. From the 10-fold CV plots for ridge and lasso, there is an improvement in mean-squared error (MSE) when the penalty Log (λ) increases. The numbers along the top of the plots represent the number of features in the model. As penalty increases, ridge retains all 10 features while lasso reduces the number of features.

The RMSE for Ridge Regression is 0.01098, and the RMSE for Lasso Regression is 0.01447. Ridge has a lower RMSE than Lasso, indicating that retaining features is preferred to generate accurate density predictions for this dataset, as opposed to Lasso's automatic feature selection.

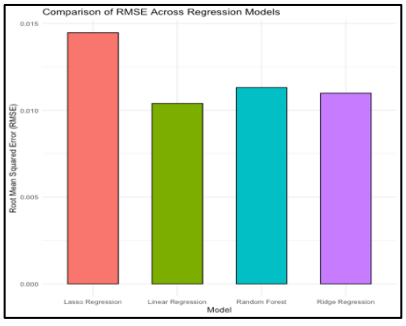
3.7 Random Forest (Feature Importance)



Feature Importance

For a Random Forest model, it grows every tree by training it based on partitions of the total training set. These partitions are executed at random with replacement. Some features are selected at random, and the optimal split on these selected features is used to split the node. Conversely, a standard tree only creates every split after assessing each feature and selecting the optimal split from all features. In our case of Random Forest, the number of trees is set to 500, and the number of features tried at each split is 3. The feature importance plot on the left shows the % increase in MSE of density predictions for each feature. The higher the % increase in MSE, the more important the feature is.
BMI and Abdomen-Chest Ratio play a vital role in predicting body density. In contrast, Hip-Thigh Ratio and Ankle have almost no influence on the prediction of body density.

3.8 RMSE Comparison



RMSE comparison

The RMSE comparison graph on the left compares the RMSE of the above-mentioned models against each other, based on their performance on testing sets. Linear Regression has the lowest RMSE out of the models, showing that it offers the highest accuracy of body density predictions. In this case, it is very likely that a linear regression model would be used for body density prediction. However, it is critical to note that this recommendation applies strictly to this specific dataset as it can change for other datasets.

3.9 Siri's Body Fat Percentage Equation

```
> testset[5,]
   Age Neck Knee Ankle Biceps Forearm Wrist   BMI ACratio HPratio
34  41 39.8 44.2  25.2   37.5    31.5  18.7 30.47     0.9    1.61
> # Linear regression has the lowest RMSE so it is used to predict bodyfat.
> predict.bodyfat(testset[5,], lr.mod)
Density:  1.04052212505393 g/cc
Percentage Body Fat:  25.7227050547778 %
```

Console Output

After obtaining the predicted Body Density value, we use it to compute Body Fat percentage using Siri's equation: Fat (%) = [(4.95 / Density) – 4.5] * 100. If we were to take the 5th row of the testing set to predict Body Density using the linear regression model, we get an estimated 1.0405 g/cc. Using the equation, the computed Body Fat percentage is 25.7%.

4. Classification

4.1 Substantive Issue

Understanding housing market trends is crucial for policymakers, investors, and residents intending to make housing decisions. In competitive markets, predicting whether a property is more likely to be sold or rented can influence real estate investment and housing policies.
The Sold and Rented HDB Residential Units dataset was collected from Singapore's Housing & Development Board (HDB), containing data from 2006 to 2021. There are 5 variables: Financial Year, Property Type, Category, Flat Type, No. of Units. Our response variable is Category, which indicates whether the total number of units of a specific property type and flat type in specific financial year is Sold or Rented.

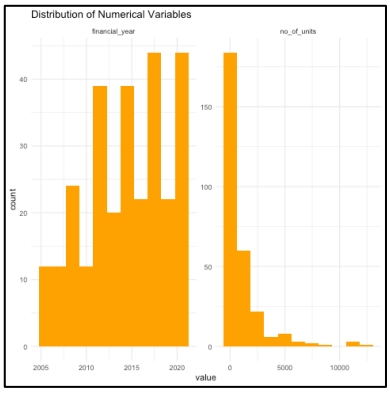
4.2 Research Questions (RQs)

- RQ1: Can a classification model accurately predict whether a residential unit will be sold or rented based on its features?
- RQ2: How do different classification models vary in their predictions for sold or rented?

4.3 Feature engineering and Distribution plots

Unlike Regression which predicts a numerical variable, Classification aims to predict a categorical variable. In this case, we want to train models to predict whether residential units are sold or not, given features like Financial Year, Property Type, Flat Type, and No. of Units.

We start by cleaning the dataset for NA values. Numerical variables like Financial Year and No. of Units are ensured to be numerical. Take note that although Financial Year here could be treated as a categorical variable, we are aiming to analyse trends over time, so it is left as a numerical variable. Categorical variables are converted to factors for Logistic Regression and Decision Tree models, whereas one-hot encoding is used for Support Vector Machines (SVM) and K-nearest neighbours (KNN).

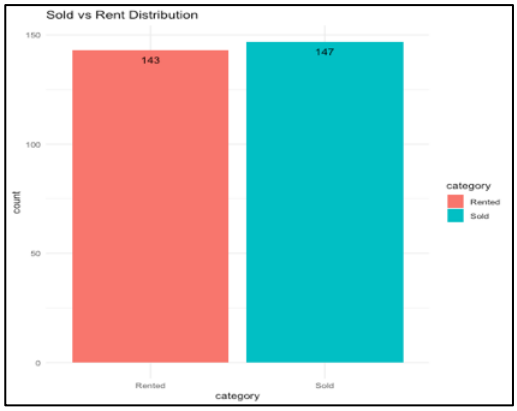


Numerical variable distribution

We use a histogram to assess numerical variables. The Financial Year distribution shows the count of observations for each financial year. It seems to be evenly distributed with higher frequencies in recent years (2015-2021), suggesting that more samples are available in those years. Basically, more property transactions occurred in recent years.

The No. of Units distribution shows the count of observations for the number of units sold/rented. It is highly right-skewed, with majority of values near 0. The outliers are located on the other end with the highest number of units. It is more common to have a small number of units sold/rented in this dataset.

We plot a distribution of the response variable, Category. There are 143 samples labeled as Rented, and 147 samples labeled as Sold. This is an estimated 49% Rented and 51% Sold. This suggests that the response variable is balanced with no significant class imbalance in the dataset. Thus, models are less likely to be biased towards the majority class.



Sold vs Rented distribution

To consolidate, the numerical features are Financial Year and No. of Units. The categorical features are Property Type and Flat Type. The response variable is Category, which is stored as a factor with two levels: Rented and Sold. The first level is assigned as 0, which is Rented. The second level is assigned as 1, which is Sold. Thus, this is suitable for binary classification.

4.4 Logistic Regression Analysis

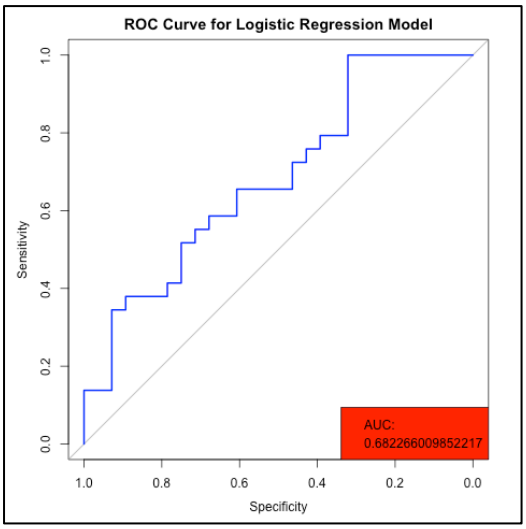
We normalise the dataset then split the dataset into 80% training set and 20% testing set. We start by training a Logistic Regression model using the training set. Consequently, we test this model against the testing set. By converting predictions from this model into class predictions, we can use a confusion matrix for evaluation.

```
Confusion Matrix and Statistics

          Reference
Prediction Rented Sold
    Rented     15   10
    Sold       13   19

               Accuracy : 0.5965
                 95% CI : (0.4582, 0.7244)
    No Information Rate : 0.5088
    P-Value [Acc > NIR] : 0.1164

                  Kappa : 0.1912

 Mcnemar's Test P-Value : 0.6767

            Sensitivity : 0.6552
            Specificity : 0.5357
         Pos Pred Value : 0.5938
         Neg Pred Value : 0.6000
             Prevalence : 0.5088
         Detection Rate : 0.3333
   Detection Prevalence : 0.5614
      Balanced Accuracy : 0.5954

       'Positive' Class : Sold
```

Confusion matrix and Statistics

True Positives (TP) show 15, which indicates the number of correctly predicted Rented.
True Negative (TN) show 19, which indicates the number of correctly predicted Sold.
False Positives (FP) show 10, which indicates the number of incorrectly predicted Rented instead of Sold.
False Negatives (FN) show 13, which indicates the number of incorrectly predicted Sold instead of Rented.
Accuracy is quite low at 59.65%, which is slightly better than random guessing.
Sensitivity at 65.52% indicates that the model is slightly better at identifying actual Sold cases than Rented ones, but still lacking.
Specificity at 53.57% indicates that the model is not good at identifying actual Rented cases.
By addressing a few of these statistics, we can tell that this model is not very reliable. Feature engineering and proper model tuning could improve its performance.

Another way of evaluating the performance of classification models is to plot the Receiver Operating Characteristic (ROC) curve based on testing data and extract the Area Under Curve (AUC) value for comparison.

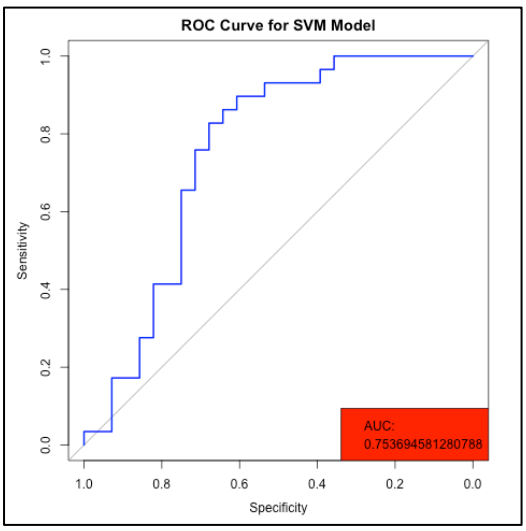


Logistic Regression ROC

From the ROC plot on the left, the Sensitivity axis represents the True Positive Rate (TPR), which is the proportion of correctly predicted Sold residential units. The Specificity axis represents the False Positive Rate (FPR), which is the proportion of correctly predicted Rented residential units.

The AUC for Logistic Regression model's ROC curve is 0.68. This suggests that the model has poor discrimination. If other models manage to get a higher AUC, those models will be preferred over logistic regression model.

4.5 SVM ROC

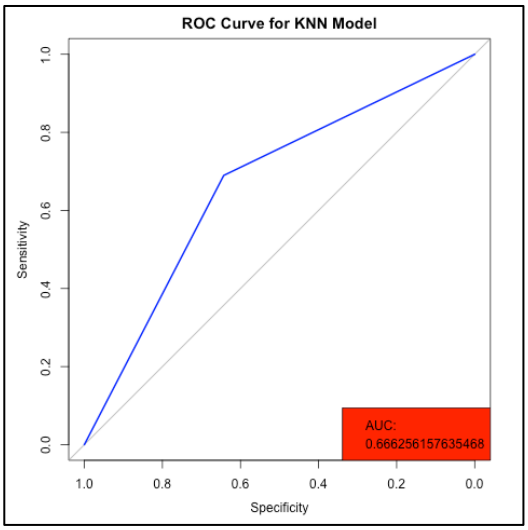


SVM ROC

For a SVM model, we execute parameters tuning using a radial basis function (RBF) to boost its flexibility and robustness for a better fit. The RBF parameters used are C and Gamma. C controls the penalty of misclassification, while Gamma alters the decision boundary by grouping similar data points. We use the best model from the optimisation for prediction, against the testing data. We extract probabilities from the predictions.

The AUC for SVM model's ROC curve is 0.75. This suggests that the model has acceptable discrimination. The SVM model performs better than the Logistic Regression model.

4.6 KNN ROC

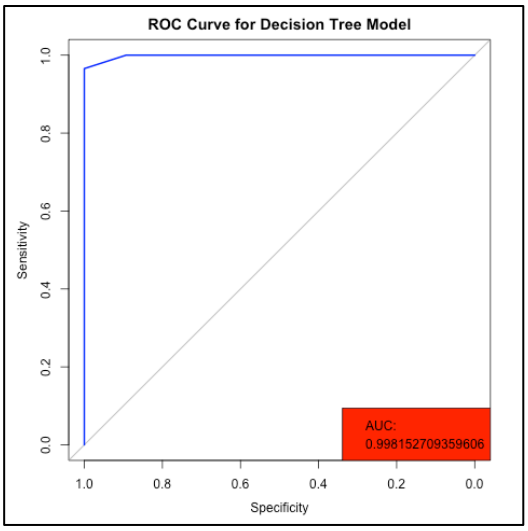


KNN ROC

For a KNN model, the algorithm uses proximity to make predictions about the class or grouping of a data point. We want to identify the optimal k value that is used to check the number of neighbours for classification. In this case, these groups refer to either Sold or Rented.

Through a 10-fold CV, the trained KNN model has an optimal k value of 3. This is the best k value to reduce the overfitting and underfitting of data, while capturing local patterns in the data. The AUC for KNN model's ROC curve is 0.67. This suggests that the model has poor discrimination and achieves a similar performance as the Logistic Regression model.

4.7 Decision Tree ROC



Decision Tree ROC

For a Decision Tree model, the algorithm predicts outcomes by splitting data based on features. Ultimately, it will result in a classification decision at the leaf nodes. In this case, the leaf nodes represent either Sold or Rented.

After training the model, we produce CV results where the CV error rates and standard deviations are represented in the columns xerror and xstd respectively. As a general rule of thumb, we prune a decision tree using the CP of the smallest tree, which has CP of 0.01 (we use a slightly higher value of 0.011) with standard deviation of 0.047423.

The AUC for Decision Tree model's ROC curve is 0.99. This suggests that the model has outstanding discrimination and it is the best performing model for this dataset.

4.8 AUC Comparison

To summarise, the AUCs based on testing data for Logistic Regression, SVM, KNN, and Decision Tree models are 0.68, 0.75, 0.6, and 0.99 respectively.
The Decision Tree model seems to perform the best out of all classification models here. However, an AUC of 0.99 may be a sign of overfitting. Despite applying a pruning technique, the AUC for the Decision Tree model did not show signs of improvement. This indicates that further tuning or feature engineering may be required. More robust models like Gradient Boosting are often better at handling overfitting.
The AUCs for Logistic Regression and KNN suggest that they perform poorly based on this dataset. Possible causes for this include poor feature selection, model complexity, and incorrect hyperparameter tuning (e.g regularisation for Logistic Regression, k neighbours for KNN).
Thus, due to the above reasons, the SVM model seems to have the best fit with acceptable discrimination for the classification of Sold and Rented cases.

5. Conclusion

For the unsupervised learning task, we discovered unique patterns and clusters of the constituents found in different types of wines.

For the regression task, our analysis on the Body Fat dataset extracted critical features to help us predict Body Density. The Linear Regression model was found to be the most accurate in Body Density prediction, and it allows us to compute Body Fat percentage successfully using the prediction.

For the classification task on predicting Sold or Rented residential units, the SVM model was found to be decent at discriminating between Sold and Rented residential units.

Our implementation of diverse models for these 3 tasks on real-world datasets has yielded informative and fascinating insights into the potential of machine learning. This will be beneficial for future data analysis by comparing between models and deciding on the most appropriate model for a particular dataset. Afterall, every dataset is different.

6. References

1. *BMI formula: Body mass index: Ramsay Health Care UK | BMI Formula | Body Mass Index | Ramsay Health Care UK*. Available at: https://www.ramsayhealth.co.uk/treatments/weight-loss-surgery/bmi/bmi-formula (Accessed: 03 April 2025).
2. J. V. G. A. DURNIN and M. M. RAHAMAN (1967) *The assessment of the amount of fat in the human body ...*, *The assessment of the amount of fat in the human body from measurements of skinfold thickness*. Available at: https://www.cambridge.org/core/services/aop-cambridge-core/content/view/DA80501B784742B9B2F4F454BDEE923B/S0007114567000728a.pdf/div-class-title-the-assessment-of-the-amount-of-fat-in-the-human-body-from-measurements-of-skinfold-thickness-div.pdf (Accessed: 03 April 2025).
3. Bobbitt, Z. (2021) *What is considered a good AUC score?*, *Statology*. Available at: http://www.statology.org/what-is-a-good-auc-score/ (Accessed: 03 April 2025).