UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
Faculty of Engineering, Built Environment & IT
University of Pretoria

COS781 - Data Mining

Group Assignment

October 7, 2022

Team Members:

Name	Surname	Student Number
Jaime	Tellie	u17021627
Jesse	Mwirigi	u17306192
Daniel	Babalola	u18041494
Peter	Okumbe	u18052640
Ngoie	Mutombo	u22608754

Contents

1 Description of chosen datasets to use. [5 marks]

1.1 Census-Income

The dataset contains census data collected by the United States of America Census Bureau between the year 1994 and 1995. The data used in this phase was obtained from the UCI KDD Archive. The data was obtained by downloading it from the above-provided link. A data conversion sub-step was required due to the fact that the format of the acquired data was quite difficult to load on the data mining tool we have opted to work with. The data was converted from a .data format to .csv format. By looking at the first few lines of the data, we observe that the data is a mixture of numerical and categorical data. The target variable is represented using a string; we can note that in later stages, we might need to change this representation to 0 and 1. Below is a summary of the data:

- 199,523 Instances.
- 40 attributes (7 continuous, 33 nominal)
- 2 classes (<= 50,000 and >=50,000)

1.2 Forest Cover Type

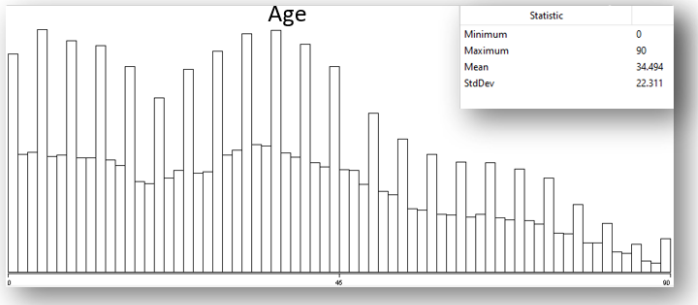
The dataset chosen for the conduction of classification modeling is the "Forest CoverType" dataset (obtained from the UCI KDD Archive), which stores data for the forest cover type observation of 30 x 30 meter cells that was obtained from the US Forest Service (USFS) Region 2 Resource Information System (RIS). The dataset was made available on the 28th of August 1998 by Jock A. Blackard, Dr. Denis J. Dean and Dr. Charles W. Anderson from Colarado State University under the department of Forest Sciences.
The dataset contains a total of 581 012 instances of data records with 54 attributes, 12 of which are measurement variable attributes, where 10 of these 12 measurement variables have quantitative data type values and 2 have qualitative data type values ('Wilderness_Area', 'Soil_Type'). A number of codes are mapped to represent the different types of values available for the 'Wilderness_Area', 'Soil_Type' and 'Cover_Type' qualitative data type variables. The 'Cover_Type' variable attribute is indicated as the classification problem target (predicted) variable that is used to predict and classify the forest cover type of the data, based on the 12 independent variables. The data type of the entire dataset is multivariate and no missing data or attribute values exist in the dataset. This was taken into consideration during the selection of the datasets, as it was thought to assist in the better conduction of both the statistical and classification modeling analysis of the data.

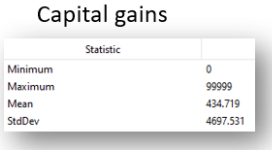
2 CRISP-DM Phase 2: Data understanding [15 marks]

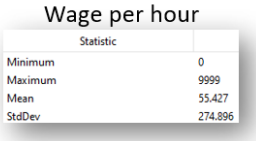
2.1 (a) Conduct appropriate statistical analysis (univariate, bivariate) on the dataset variables. [10 marks]

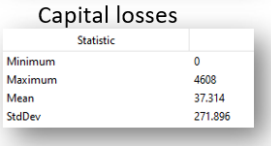
2.1.1 Census-Income Statistical Analysis

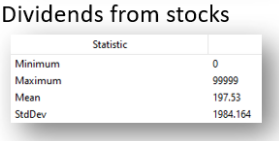
Univariate

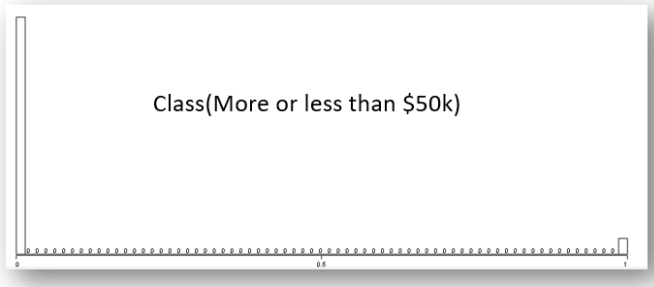


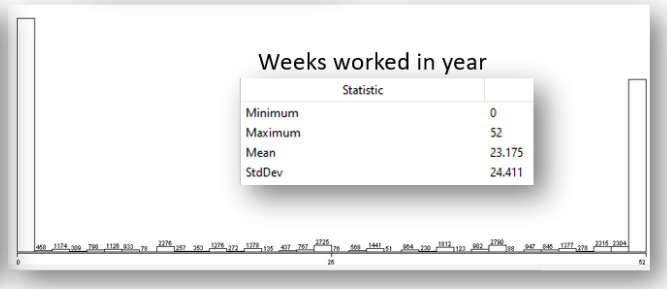










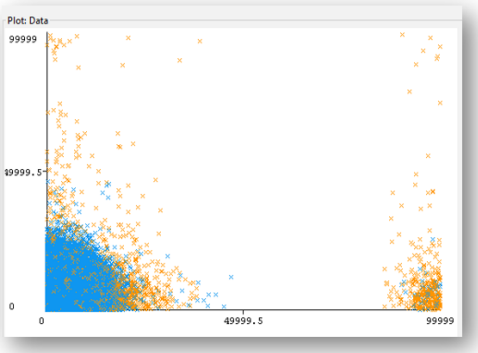


Bivariate

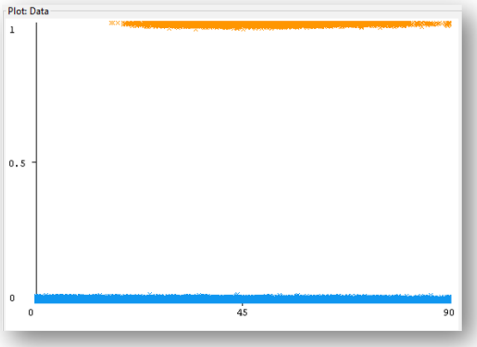
Correlation

First Attribute	Second Attribute	Correlation
Own business or self employed	Weeks worked in a year	0,266
Weeks worked in a year	Class(Income)	0,262
Capital gains	Class(Income)	0,241
Dividends from stock	Class(income)	0,176

Dividends from stocks vs Capital gains



Age vs Income



2.1.2 Forest Cover Type Statistical Analysis

	Mean	Std	Min	Max
Aspect	155.657	111.914	0.000	360.000
Hillshade 9am	212.146	26.770	0.000	254.000
Hillshade 3pm	142.528	38.275	0.000	254.000
Vertical Distance To Hydrology	46.419	58.295	-173.000	601.000
Horizontal Distance To Hydrology	269.428	212.550	0.000	1,397.000

Table 1: Statistical analysis of the chosen numerical features of the Forest Cover Type data set.

Univariate Analysis: Histograms and Distributions

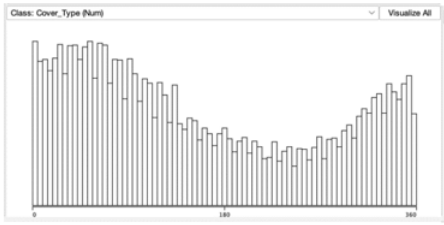


Figure 1: 'Aspect' Variable Histogram

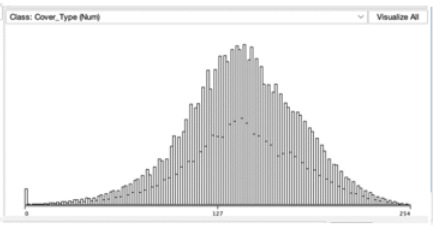


Figure 2: 'Hillshade_3pm' Variable Histogram

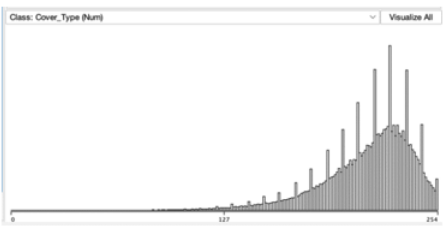


Figure 3: 'Hillshade_9am' Variable Histogram

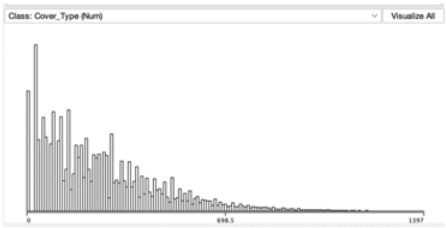


Figure 4: 'Horizontal Distance to Hydrology' Variable Histogram

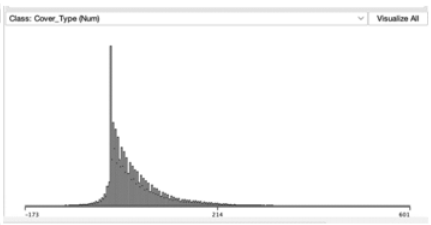


Figure 5: 'Vertical Distance to Hydrology' Variable Histogram

Bivariate Analysis: Scatter Plots

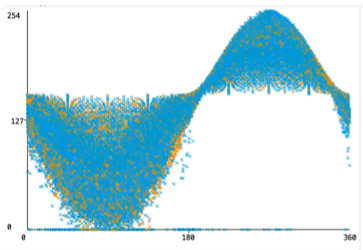


Figure 6: Hillshade_3pm vs. Aspect

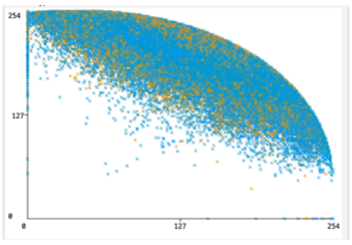


Figure 7: Hillshade_9pm vs. Hillshade_3pm

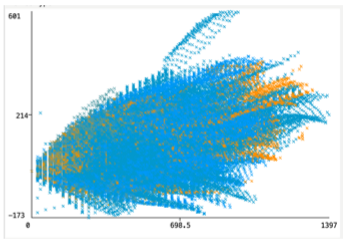


Figure 8: Vertical distance to Hydrology vs. Horizontal distance to Hydrology

2.2 (b) Provide an explanation of the analysis results for each variable. [5 marks]

2.2.1 Census-Income Statistical Analysis

1. Univariate explanation: The age ranges from 0 to 90, with a mean of 34.494 and a standard deviation of 22.311. The standard deviation is lower than the mean, meaning data are clustered around the mean.Looking at instances with a 0 age value, we see that they have no education data, and most of the other variables also are classified as children. We can already note that these might be outliers that might need to be dealt with in later stages.

2. Bivariate explanation: From the correlation table, we see very weak correlations between different attributes. We note that as the amount of capital gains or dividends from stock increase, so does the chance of earning above 50k. As we are avoiding making any assumptions, we explore the data by filtering dividends from stock in descending order. From the activity, we note that the top 134 individuals in terms of dividend amounts have an income of over 50k with the lowest dividends amount being 31262. This makes sense, especially because if an individual is receiving a dividend of +50k then by default their income is above 50k.On the dividends from stocks vs capital gains, we see that majority of +50k individuals either are earning some dividends or capital gains

2.2.2 Forest Cover Type Statistical Analysis

1. Univariate explanation: From the visual aids presented in the research report, we are able to analyze the presence of specific variables. We have identified the 5 variables in the univariate analysis to form part of our most important feature set. In particular, we are able to see that vertical and horizontal distance to a hydrology point are some of the most important features we should consider when regarding forest cover. Both figures 4 and 5 show that the variables are skewed more towards the right. Whereas figure 3 shows a graph with the skewness towards the left.

2. Bivariate explanation: Visualization played a crucial role in gaining a thorough understanding of the data. We selected relationships based on attributes that shared a strong correlation value greater than 0.5 (strong positive correlation) or less than -0.5 (strong negative correlation). In total we identified six pairs of correlated attributes. These are 'Hillshade_3pm & Aspect'; 'Hillshade_9am & Aspect'; 'Hillshade_Noon & Slope'; 'Vertical_Distance_To_Hydrology' & 'Horizontal_Distance_To Hydrology'; 'Hillshade_9am & Hillshade_3pm'; 'Hillshade_Noon & Hillshade 3pm'. We included the graphs for three of the pairs, as shown in figure 2.1.2 above.

Figure 2.2.2 below shows the correlation matrix for this numeral attributes of the data set. After observing the resultant scatter-plots, we discovered interesting patterns. Such as how the relationship between Hillshade_3pm and Aspect is a sigmoid function.

The images displayed in figure 2 illustrate the generated correlation values (highlighted in green) for each of the chosen 3 pairs of variables in the forest cover data-set that are used for bivariate data

analysis, which are generated using the Pearson sample correlation coefficient method with a 2 tailed test of significance in the IBM SPSS software.

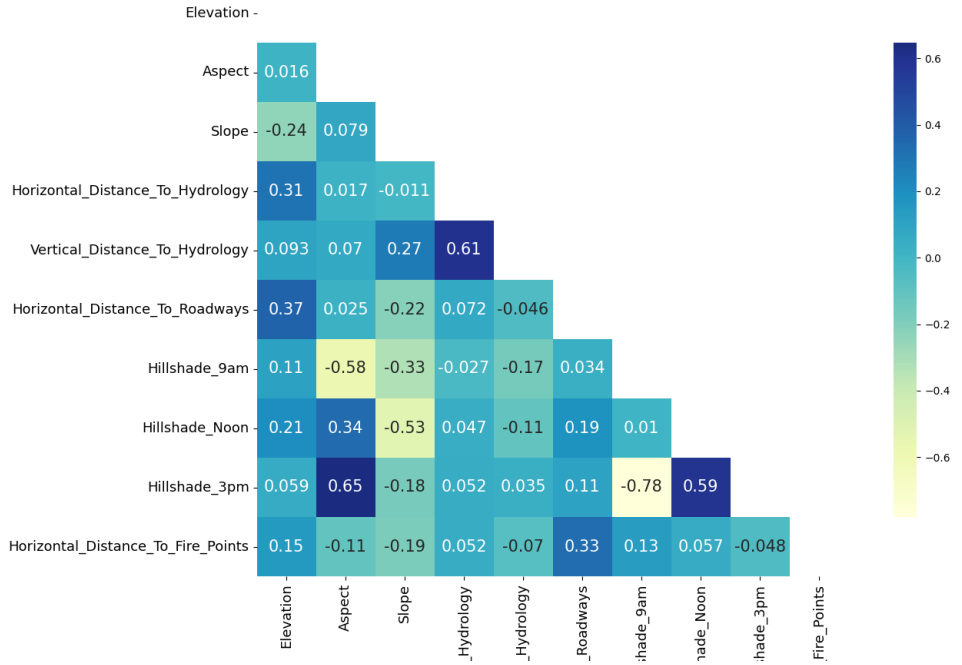


Figure 1: Correlation matrix (using Pearson correlation) for the 10 numerical attributes of the Forest Cover data-set.

Correlations

		Hillshade_3pm	Aspect
Hillshade_3pm	Pearson Correlation	1	.647**
	Sig. (2-tailed)		.000
	N	581012	581012
Aspect	Pearson Correlation	.647**	1
	Sig. (2-tailed)	.000	
	N	581012	581012

**. Correlation is significant at the 0.01 level (2-tailed).

Hillshade_3pm vs. Aspect

Correlations

		Hillshade_3pm	Hillshade_9am
Hillshade_3pm	Pearson Correlation	1	-.780**
	Sig. (2-tailed)		.000
	N	581012	581012
Hillshade_9am	Pearson Correlation	-.780**	1
	Sig. (2-tailed)	.000	
	N	581012	581012

**. Correlation is significant at the 0.01 level (2-tailed).

Hillshade_3pm vs. Hillshade_9am

Correlations

		Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Hydrology
Vertical_Distance_To_Hydrology	Pearson Correlation	1	.606**
	Sig. (2-tailed)		.000
	N	581012	581012
Horizontal_Distance_To_Hydrology	Pearson Correlation	.606**	1
	Sig. (2-tailed)	.000	
	N	581012	581012

**. Correlation is significant at the 0.01 level (2-tailed).

Vertical_Distance_To_Hydrology vs. Horizontal_Distance_To_Hydrology

Figure 2: Generated correlation values for the 3 chosen pairs of variables used for bivariate analysis

3 3. CRISP-DM Phase 3 and 4: Data preparation and modeling [50 marks]

3.1 (a) Cluster analysis activities [20 marks]

3.1.1 i. Use one of the datasets for cluster analysis.

WEKA [1] software was the main tool utilised in the clustering activity of the Census Income Data.

3.1.2 ii. Conduct dimensionality reduction before performing the cluster analysis. [5 marks]

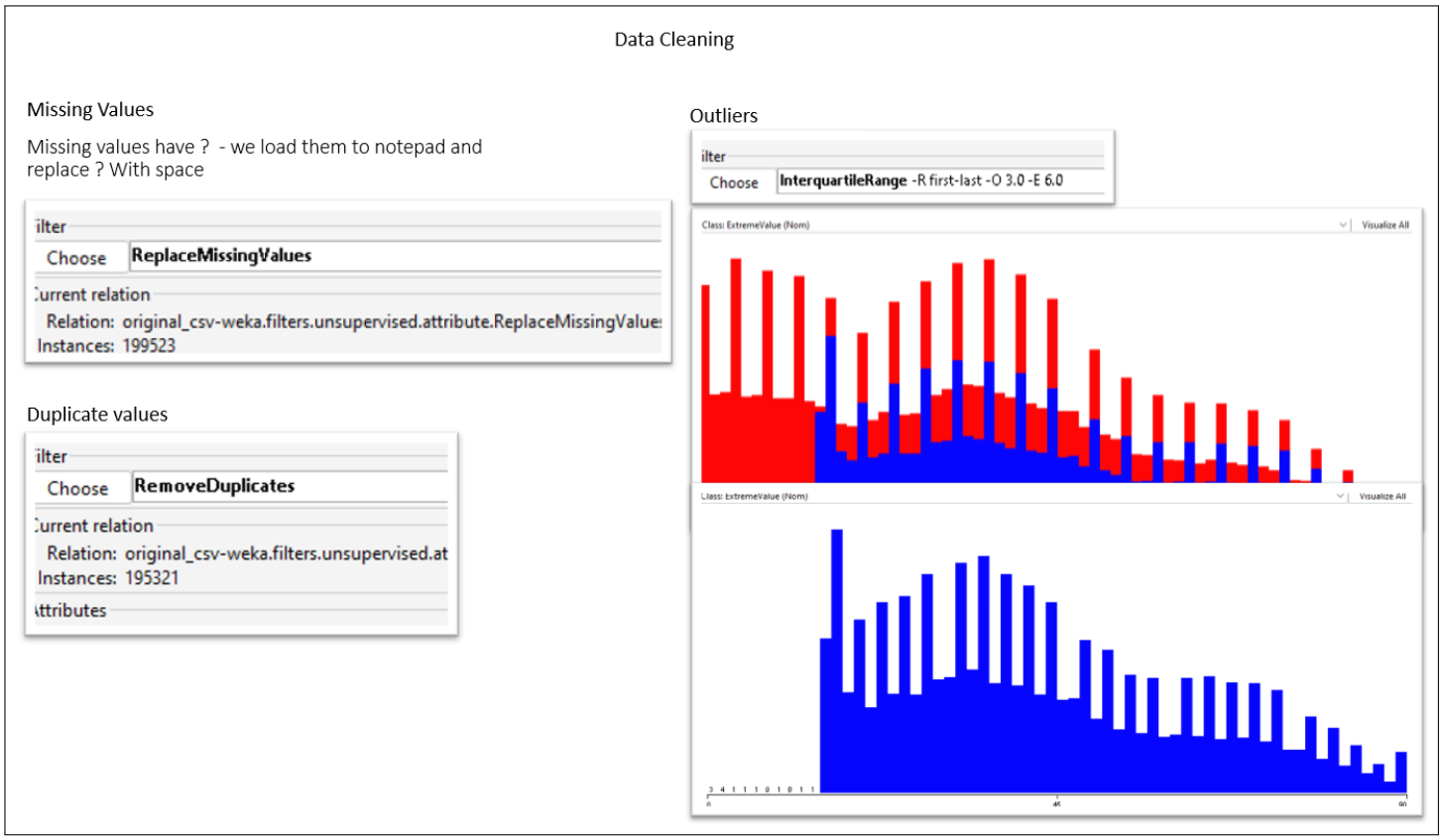


Figure 3: Data cleaning activities

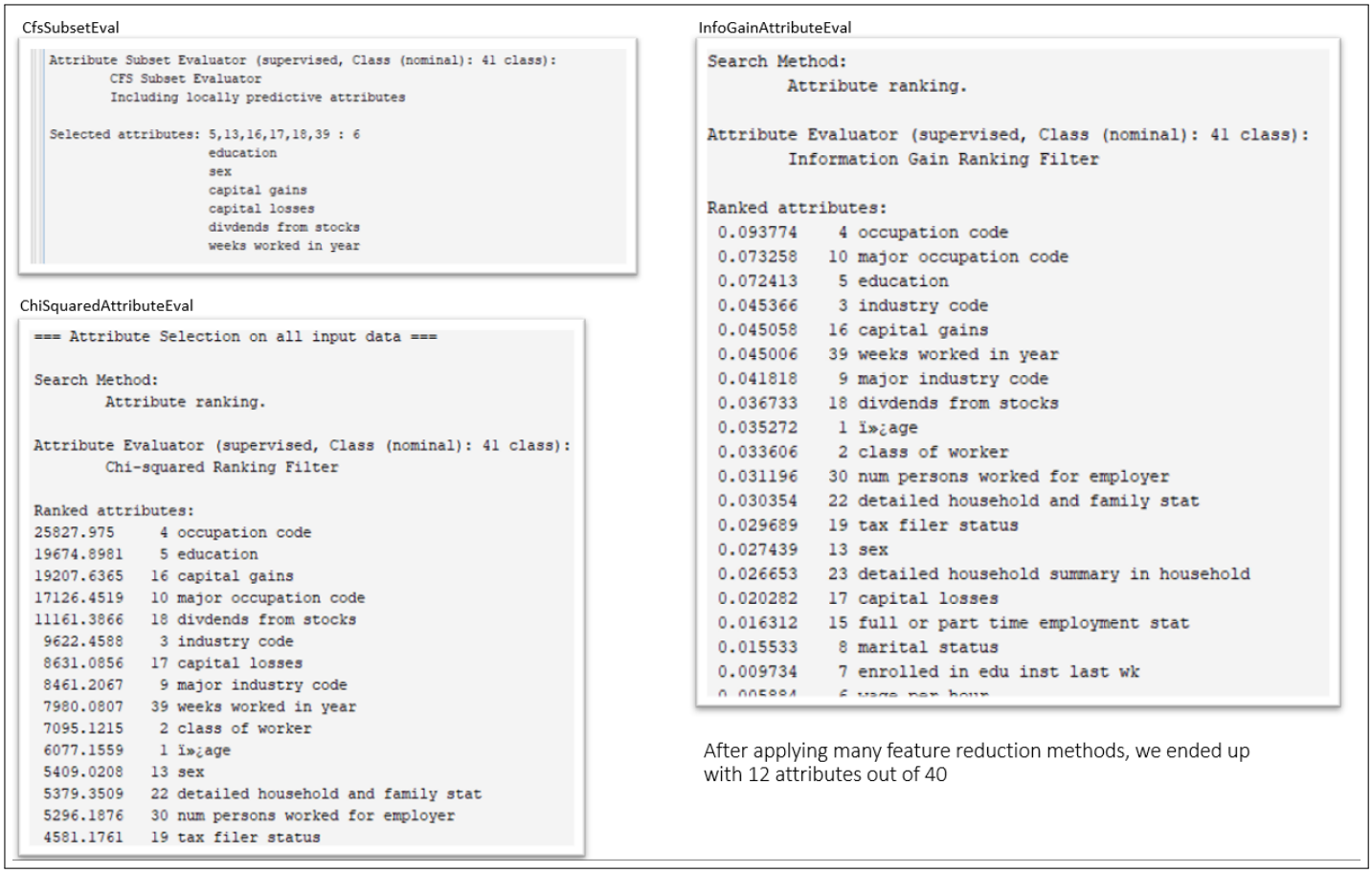


Figure 4: Dimensionality reduction activities

3.1.3 iii. Use two methods of your choice to conduct the analysis. [10 marks]

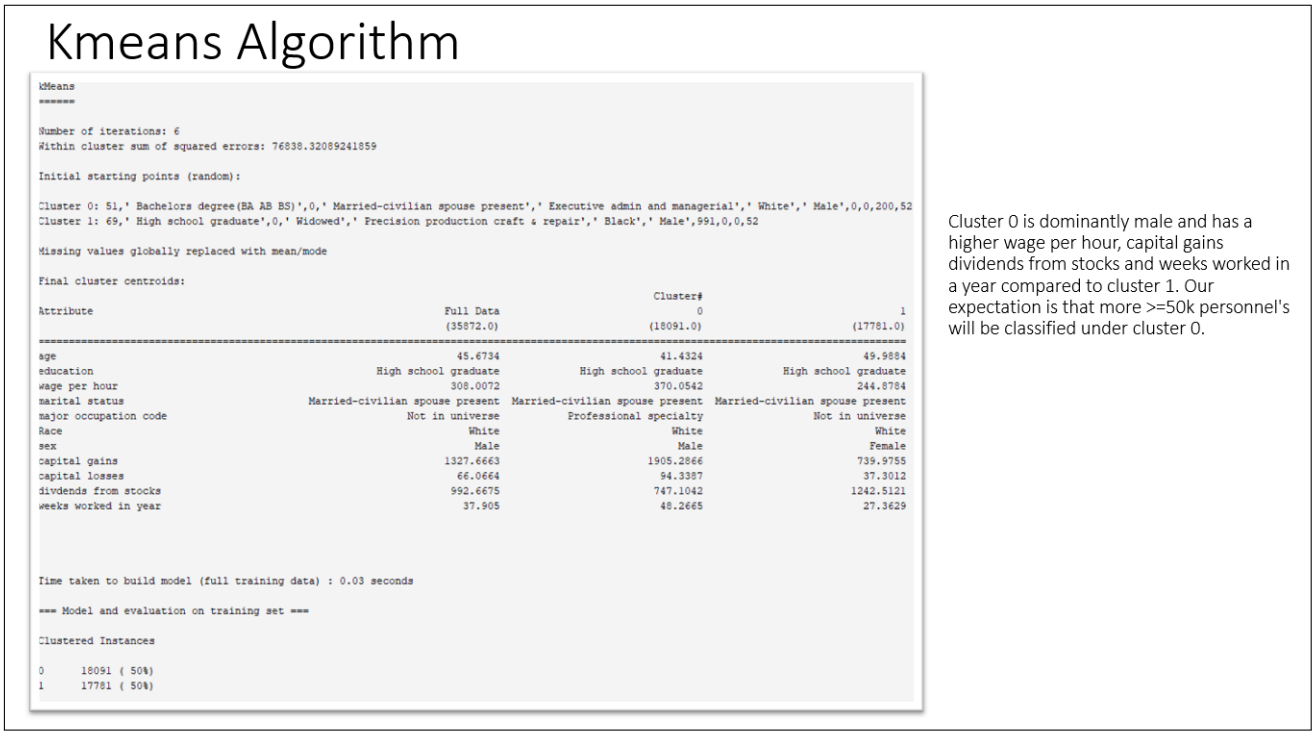


Figure 5: K-means clustering using Weka

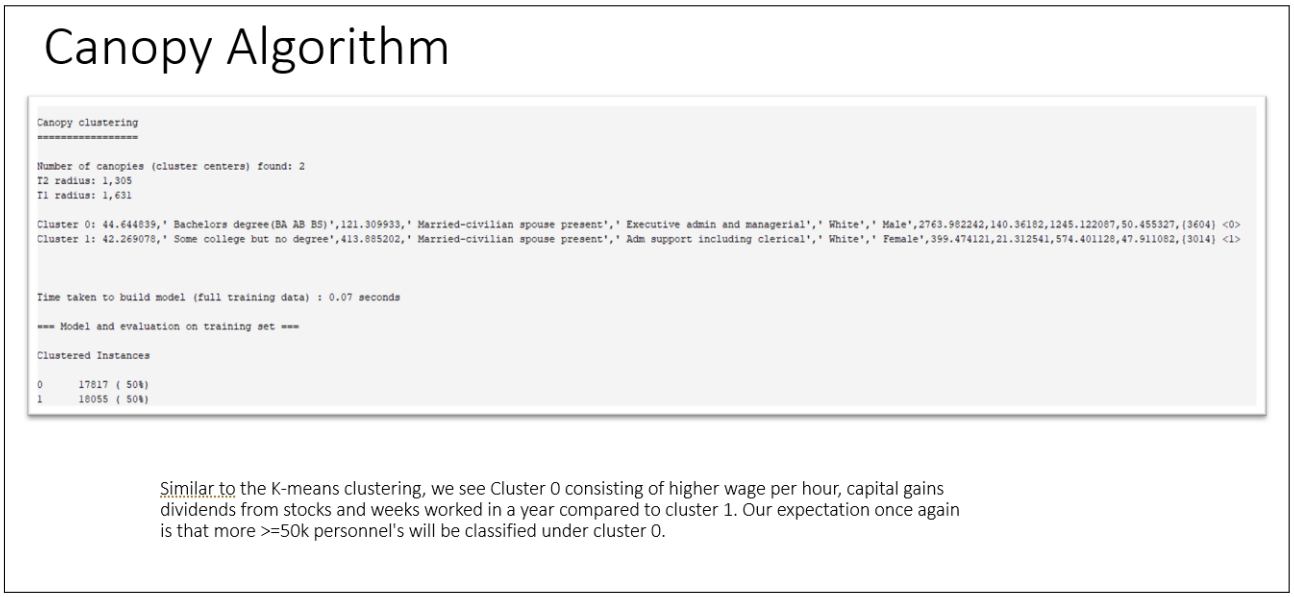


Figure 6: Canopy clustering using Weka

3.1.4 iv. Perform a statistical comparison of the results of the cluster analysis for the two methods. [5 marks]

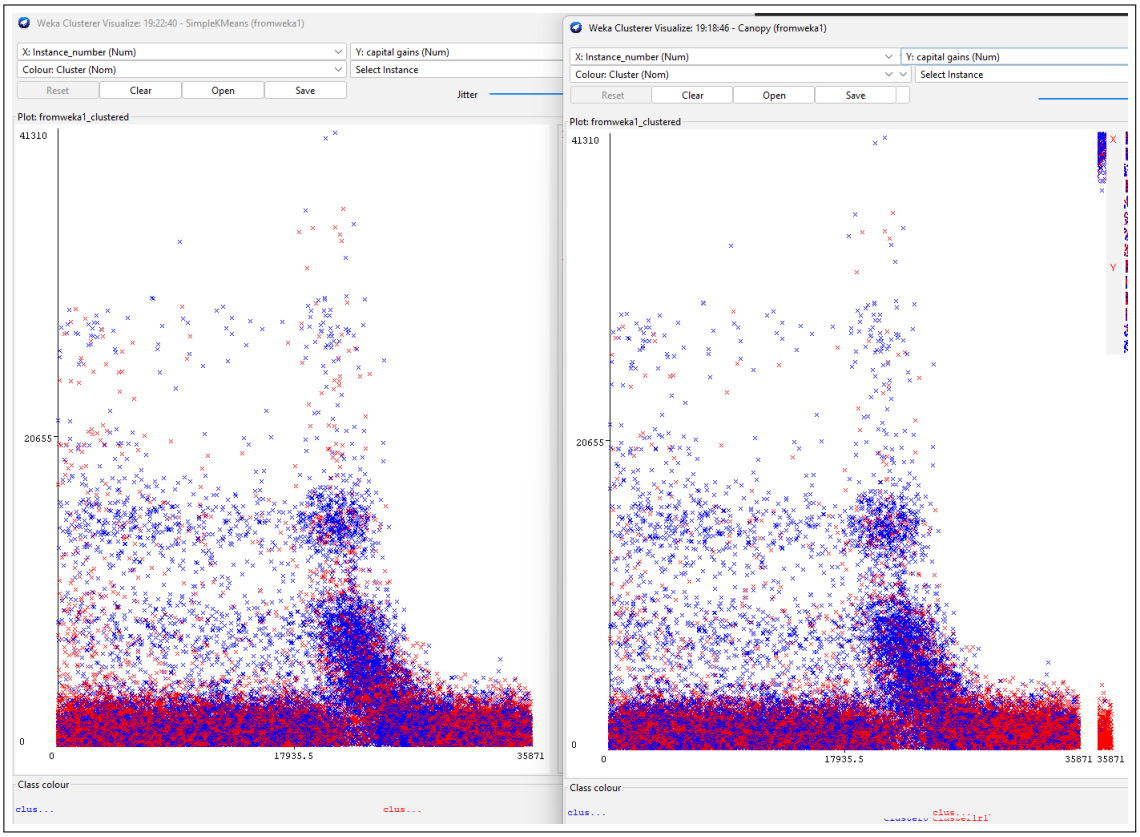


Figure 7: Capital gain comparison of the algorithms

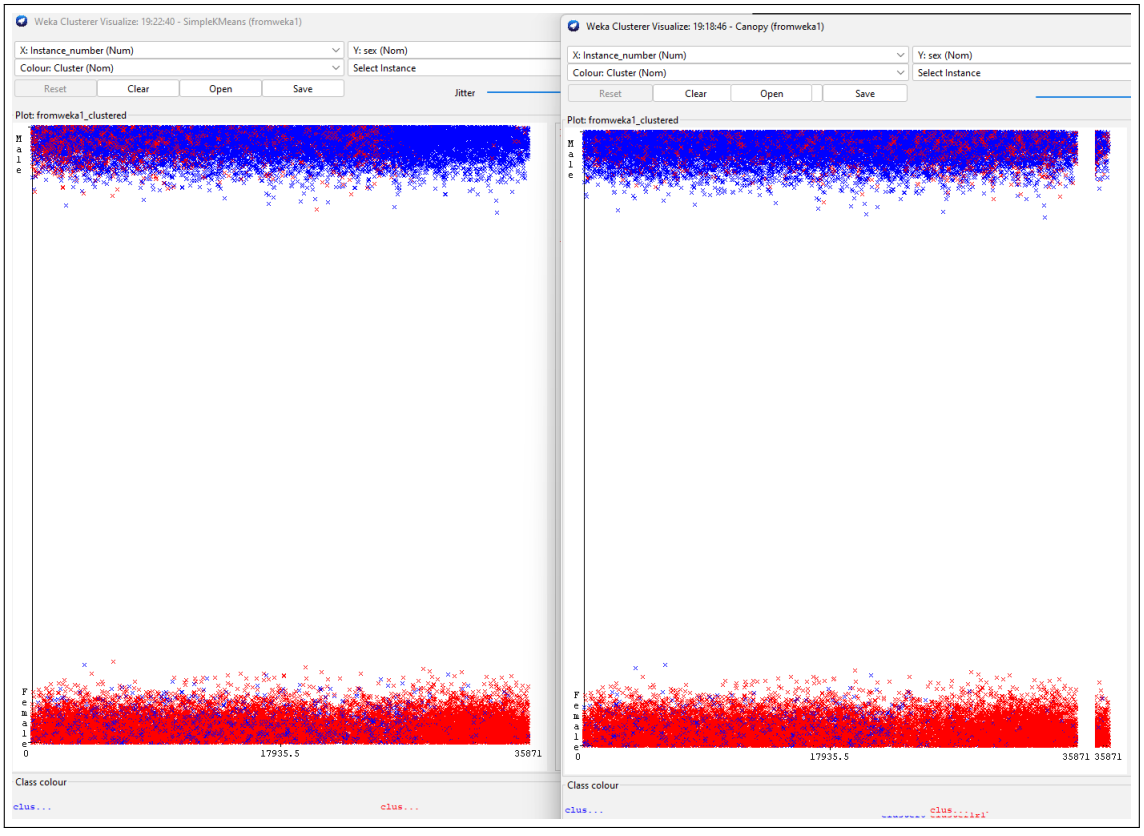


Figure 8: Sex comparison of the algorithms

3.1.5 Summary report

- **Context** -: The case study analyses the census income based on the basic characteristics regarding a potion of the population from data collected between 1994 and 1995 as mentioned earlier.

- **Objectives** -: The goal of this analysis study is to classify each individual instance of the dataset into a known or unknown number of most similar categories or classes. The categories will then be explicated based on statistical features, such as mean and standard deviation of certain variables or attributes. To achieve the analysis, supervised learning was utilized.

- **Organization of the data** -: The data used in the case study was collected in a .data format file. Each row records data regarding individual form the age of 0 to 90 years old together with their basic information such gender, marital status, education, employment and more. The dataset contains 199,523 rows i.e. each row represent a person who participated in the survey and 42 columns i.e. attributes that where collected per person.

- **Data pre-processing** -: The data had several missing values which were replaced by the modes of respective columns. Other attributes contained outliers which were dealt using different methods such as removing all the rows with an Age of less than 16. Duplicate records were also eliminated.

- **Exploratory data analysis** -: Univariate analysis was done by checking measures of dispersion such as standard deviation and visualizing through histograms. Bivariate analysis was carried out by looking at the correlation.

- **Model specification** -: The analysis used non-hierarchical clustering methods due the massiveness of the dataset. K-means and Canopy algorithms were utilized to conduct cluster analysis and compared the performance of the two models.

- **Model interpretation** -: It was noted that the more clusters were achieved, the more biased the clustering got. We explored other clustering methods such as DBSCAN using an epsilon value of 6,8 and min points value of 130 and achieved similar clusters.

3.2 (b) Predictive classification modeling activities [30 marks]

3.2.1 i. Use the second dataset for classification modeling.

Both the Python [2] and WEKA [1] software tools were used for the conduction of the classification modeling analysis activities on the forest cover type dataset.

3.2.2 ii. Data partitioning and sampling (training data, validation data, test data)

WEKA

WEKA was used to generate a visualization of the built classification tree model using a reduced sample number of 10 000 instances of the entire dataset obtained by applying the 'ReservoirSample' unsupervised instance filter on the entire Forest Cover Type dataset in WEKA (figure 9), in order to assist in better and easier visualization and interpretation of the generated classification tree, where as the entire Forest Cover Type dataset's number of instances was used in python for the building and assessment of the classification tree model's performance. The classification tree model was build on a training and testing set split of 75% for training and 25% for testing in both WEKA and python.

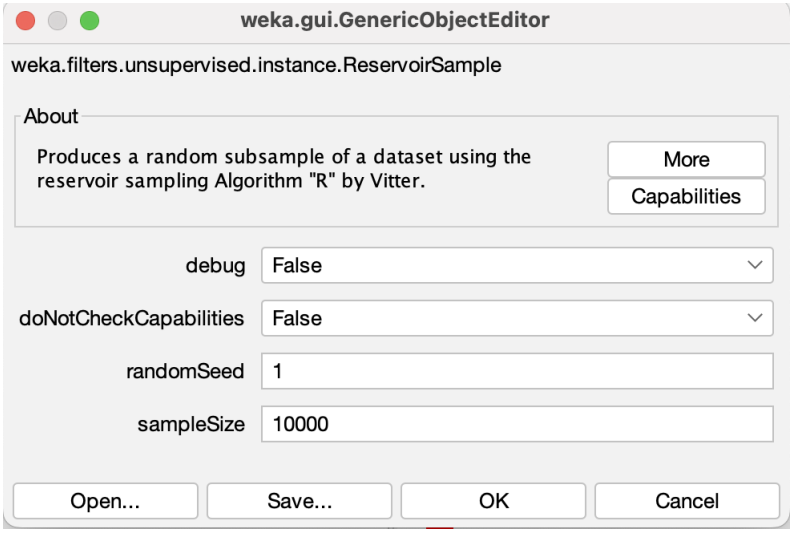


Figure 9: Sample reduction of Forest Cover Type dataset instances used for the classification tree model analysis in WEKA

Python

```
""" Sampled data based on the top 20 most important attributes according to
    RandomForestClassifier & ExtraTreesClassifier """
sample = data[['Elevation','Horizontal_Distance_To_Roadways','
    Horizontal_Distance_To_Fire_Points','Horizontal_Distance_To_Hydrology','
    Vertical_Distance_To_Hydrology','Aspect','Wilderness_Area4','Hillshade_Noon','
    Hillshade_3pm','Hillshade_9am','Slope','Soil_Type22','Soil_Type10','Soil_Type4
    ','Soil_Type34','Soil_Type34','Wilderness_Area3','Soil_Type12','Soil_Type2','
    Wilderness_Area1', 'Cover_Type']]
#from sklearn.preprocessing import MinMaxScaler
#Specifying the feature range for the scaler function
scaler = MinMaxScaler(feature_range = (0,1))
```

```
""" Moving sample features into X and target variable into Y """
X = sample.iloc[:,:-1]
y = sample['Cover_Type']
""" Scaling the sampled data according to a range funtion """
X_scaled = scaler.fit_transform(X)
#from sklearn.model_selection import train_test_split
#Spliting the data set into a 75%-25% train-test data set respectively
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size =
    0.25, random_state = 53)
print(X_train.shape, X_test.shape)
```

Listing 1: Python Dimensionality Reduction and Sampling

```
(435758, 20) (145253, 20)
```

Listing 2: Sampled Training-Test Split Output

3.2.3 iii. Dimensionality reduction (feature selection). [5 marks]

Both data transformation (figure 10) and dimensionality reduction (figure 11) was applied on the Forest Cover Type dataset used for the conduction of the classification model analysis in WEKA. The data transformation step involved making use of discretization, which is a method followed for the dividing and partitioning of numeric data into categorical data using bin intervals, in order to convert the 'forest_cover_type' predicted variable datatype from numeric to nominal, as the building of a classification model tree requires the datatype of the predicted variable of the dataset being used for analysis to be categorical. Dimensionality reduction was performed to obtain a decreased sample space of the amount of data attributes used for analysis with selecting and keeping only those variable attributes that are calculated to retain the most information about the entire dataset, such that extra redundant and irrelevant data is not used and taken into consideration during data analysis, which could lead to time wasted during the generation and analyzation of results if such extra data is included in the analysis of the data.

The approach followed for the conduction of dimensionality reduction on the Forest Cover Type dataset involved feature selection of data based on correlation analysis, where the correlation between each variable attribute and the predicted variable was automatically calculated by WEKA through the use of the correlation based feature selection approach by applying the 'CorrelationAttributeEval' unsupervised attribute filter on the entire Forest Cover Type dataset in WEKA (figure 11). Out of all the existing variable attribute features in the dataset, only the variable attribute features whose ranked positive correlation value is greater than the cut-off correlation value or whose negative correlation value is less than -(the cut-off correlation value). The variable attributes whose correlation values are closest to 0 and -0 are excluded from selection. The cut-off correlation value point was chosen as 50% of the maximum highest ranked correlation value = 0.5 x 0.28604 = 0.14302. As illustrated in figure 12, only the top 4 ranked variable attribute features were selected, as the correlation values of the top 4 ranked variable attribute features are greater than 0.14302.

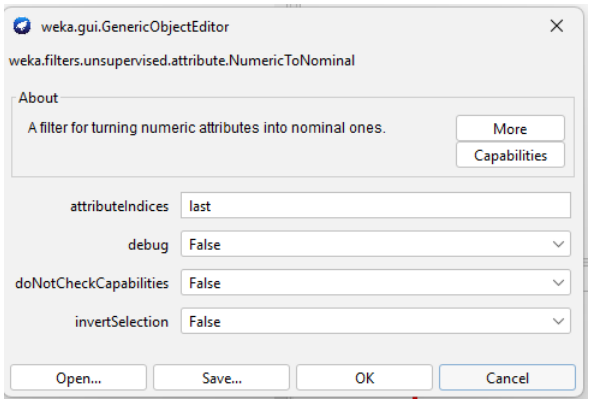


Figure 10: Data transformation of the 'forest_cover_type' predicted variable attribute used for the classification tree model analysis in WEKA

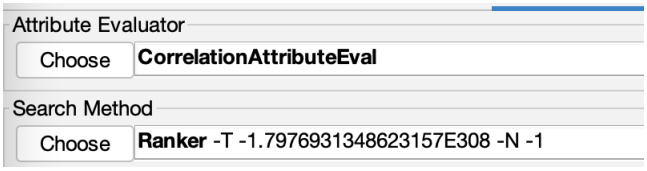


Figure 11: Generation of dimensionality reduction on Forest Cover Type dataset attributes used for the classification tree model analysis in WEKA

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 55 Cover_Type):
        Correlation Ranking Filter
Ranked attributes:
 0.28604     1 Elevation
 0.22405    14 Cache_la_Poudre_Wilderness_Area
 0.15171    26 ST_4744
 0.14725    36 ST_7201
```

Figure 12: Attribute selection of Forest Cover Type dataset attributes used for the classification tree model analysis in WEKA

3.2.4 iv. Classification tree. [10 marks]

The generation of the classification tree model in WEKA was obtained by applying the 'J48' tree classifier filter on the entire Forest Cover Type dataset in WEKA (figure 13) using a training and test split of 75% and 25% respectively. The results and visualization of the generated classification tree in WEKA is illustrated in figures 14 and 15 below.

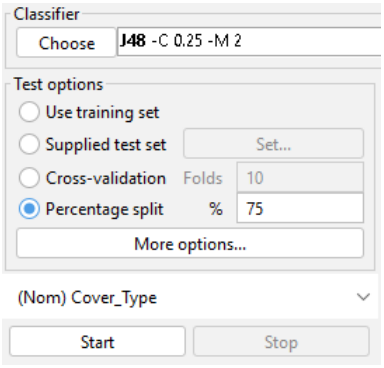


Figure 13: Generation of classification tree model in WEKA

```
Time taken to build model: 0.13 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances        1660               66.4    %
Incorrectly Classified Instances       840               33.6    %
Kappa statistic                          0.4319
Mean absolute error                      0.1348
Root mean squared error                  0.263
Relative absolute error                 76.0087 %
Root relative squared error             88.0969 %
Total Number of Instances             2500
```

Figure 14: Results of generated classification tree model in WEKA

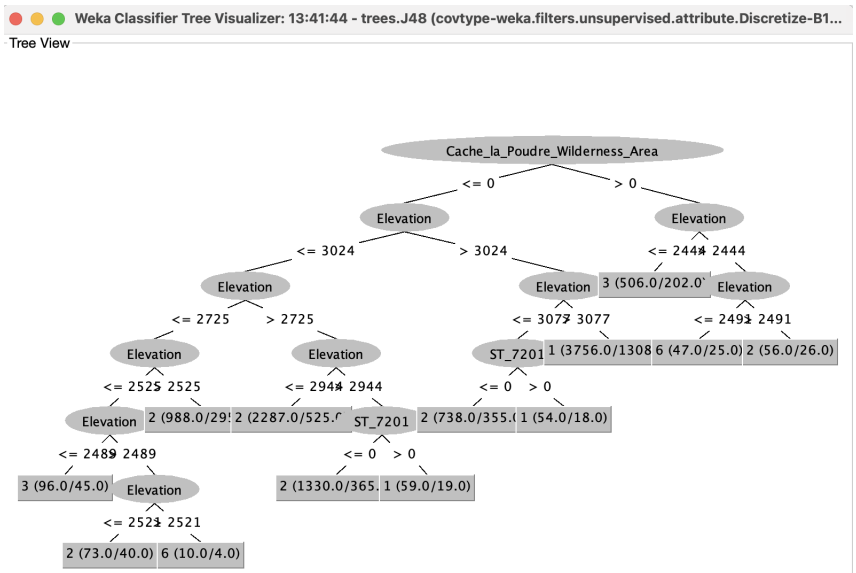


Figure 15: Visualization of generated classification tree model in WEKA

3.2.5 v. MLP ANN. [10 marks]

A model evaluation function was written to evaluate a given classifier against the train-test data set. The evaluator function measures the time taken to train the model and the time taken to cross validate (CV) the model with the expected classification. The 10 K-Fold CV method was chosen, as it gives a good estimate of the skill of the model on new data, and it falls inline with expected project criteria.

Time taken will be different based on system specifications. The evaluations were ran on a 8GB Windows 11 with a 2.10Ghz 2 core CPU.

```
import numpy as np # for scientific computing
from sklearn.model_selection import cross_val_score # to measure performance
# fucntion
def model_evaluation(clf):
    clf = clf
    t_start = time.time()
    clf = clf.fit(X_train, y_train)
    t_end = time.time()
    c_start = time.time()
    # 10 K - Fold Cross Validation
    accuracy  = cross_val_score(clf, X_train, y_train, cv = 10, scoring = 'accuracy')
    f1_score = cross_val_score(clf, X_train, y_train, cv = 10, scoring = 'f1_macro')
    c_end = time.time()
    # Average calculated according to the 10 observation's accuracy and f1 scores
    acc_mean = np.round(accuracy.mean() * 100, 2)
    f1_mean = np.round(f1_score.mean() * 100, 2)
    t_time = np.round((t_end - t_start) / 60, 3)
    c_time = np.round((c_end - c_start) / 60, 3)
    clf = None
    # Output
    print("Accuracy score:", acc_mean,"% and F1 score:", f1_mean,"% taking", t_time,"minutes to train and", c_time,
          "minutes to evaluate cross validation and metric scores.")
```

Listing 3: Model Evaluator applying 10 test sets

```
""" from sklearn.neural_network import MLPClassifier
Running the MLP ANN Classifier, using two different setting types"""
model_evaluation(MLPClassifier(solver='adam', max_iter=10000, alpha=1e-5, hidden_layer_sizes=(5,2), random_state=1))
model_evaluation(MLPClassifier(solver='adam', max_iter=3000, alpha=1e-5, hidden_layer_sizes=(15,), random_state=1))
```

Listing 4: Python MLP ANN Application

```
Accuracy score: 71.31 % and F1 score: 41.59 % taking 1.696 minutes to train and 26.249 minutes to evaluate cross validation and metric scores.
Accuracy score: 74.48 % and F1 score: 59.19 % taking 5.662 minutes to train and 76.897 minutes to evaluate cross validation and metric scores.
```

Listing 5: MLP ANN Model Evaluator Output

3.2.6 vi. Create 10 test sets for the models.

The recommended size of 10 test sets were created in order to test the quality and validity of the built classifier models. The procedure followed for the creation of the 10 test sets for the classification tree model is illustrated in figures 16 and 18 respectively seen below. Results of the classification tree model's conducted tests using the created test sets are displayed in figure 17.

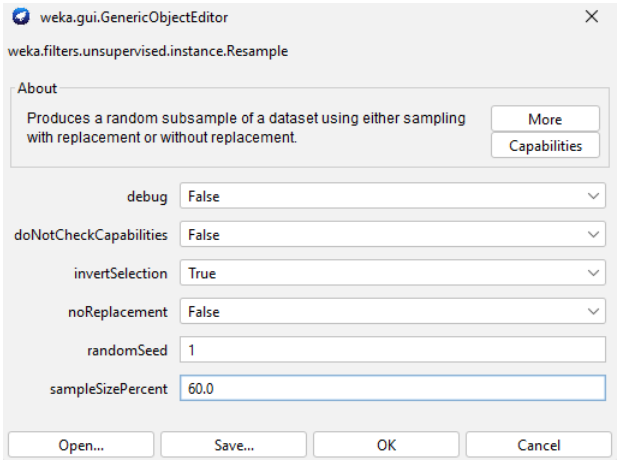


Figure 16: Generation of test sets for classification tree model using WEKA

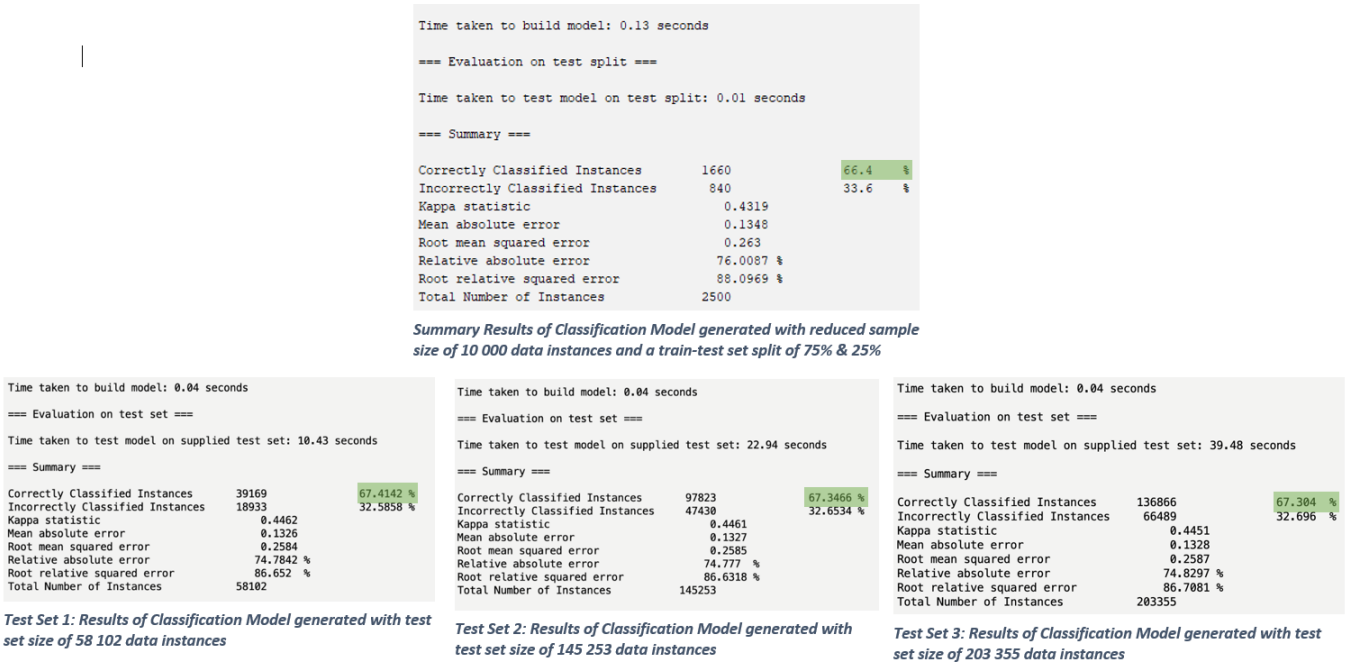


Figure 17: Test set results of classification tree model generated in WEKA

As illustrated in figure 17 above, we have generated multiple test sets to measure the performance of our classification tree model. The accuracy score of the built model provides an accuracy score of about 67%. Although the score is not the highest it is still satisfactory. We created various test sets but only used 3 to compare them against one another in order to analyze the consistency of the model. As can be seen in Figure 13, each test set contained different data and different sample sizes. These sample sizes ranged from 58 000 to 200 000. In the end, each test set was put through the model and produced a similar accuracy score of around 67%. This ensures the consistency and validity of the classification tree model.

```python
import numpy as np # for scientific computing
from sklearn.model_selection import cross_val_score # to measure performance
# fucntion
def model_evaluation(clf):
    clf = clf
    t_start = time.time()
    clf = clf.fit(X_train, y_train)
    t_end = time.time()
    c_start = time.time()
    # 10 K - Fold Cross Validation
    accuracy  = cross_val_score(clf, X_train, y_train, cv = 10, scoring = 'accuracy')
    f1_score = cross_val_score(clf, X_train, y_train, cv = 10, scoring = 'f1_macro')
    c_end = time.time()
    # Average calculated according to the 10 observation's accuracy and f1 scores
    acc_mean = np.round(accuracy.mean() * 100, 2)
    f1_mean = np.round(f1_score.mean() * 100, 2)
    t_time = np.round((t_end - t_start) / 60, 3)
    c_time = np.round((c_end - c_start) / 60, 3)
    clf = None
    # Output
    print("Accuracy score:", acc_mean,"% and F1 score:", f1_mean,"% taking", t_time,"minutes to train and", c_time,
          "minutes to evaluate cross validation and metric scores.")
```

Figure 18: Generation of test sets for MLP ANN model using Python

3.2.7 vii. Use Student's paired samples t-test to compare the tree and ANN model performance. [5 marks]

The Train-test size is made up of two brackets indicating the number of instances for the training and testing set respectively and the number of attributes used for those instances. We followed a 75 to 25 percent train-test split rule.

Model	**Platform**	**Train-Test Size**	**Accuracy**
MLP ANN Classifier	Python	(435 758, 20) (145 253, 20)	74.48%
Classification Tree Classifier	WEKA	(7 500, 4) (2 500,4)	66.4%

Table 2: Table of results.

4 CRISP-DM Phase 5: Evaluation

- **Context** - The purpose of the conduction of this group assignment was to perform both cluster and predictive modeling data analysis on 2 chosen datasets obtained from the UCI KDD Archive, in order to obtain an understanding of how to utilize, follow and practically apply each of the phases that form a part of the CRISP-DM process for the mining and analyzation of data provided by real datasets.

- **Objectives** - The aim of the analysis of the Census-Income dataset is to assess the income patterns of the collected data of various individuals, where as the aim of the analysis of the Forest Cover Type dataset is to predict forest cover types, which involves discovering the amount of land area that would be covered by a forest. Furthermore, the analysis of the Forest Cover Type dataset also assists in providing an understanding to measure which predictive classification models can accurately classify and present a good classification representation of the utilized dataset's predicted variable output values, by making use of the available predictor variables in the utilized dataset.

- **Organization of the data** - This involved the conduction of the data preparation phase in the CRISP-DM process on the respective Census-Income and Forest Cover Type datasets used for data analysis. Verification of data quality, data transformation and dimensionality reduction were some of the data mining activities performed during the organization of the data used in the respective datasets before the conduction of analysis. The Forest Cover Type dataset was discovered to have no missing values, which greatly influenced the decision to make use of the dataset for conducting data analysis and this greatly helped to ease and fasten the process of cleaning and verifying the data used, unlike the data found in the Census-Income dataset, which was presented with a number of incomplete and missing data that had to be catered for before engaging in the conduction of the data analysis activities with the data in the Census-Income dataset.

- **Exploratory data analysis** - This involved the conduction of the data understanding and data preparation phases in the CRISP-DM process on the respective Census-Income and Forest Cover Type datasets used in order to collect, describe, explore and verify the data of the respective datasets, and then further select, clean, construct, integrate and format the data in order to verify the quality of the data in the datasets used for data analysis. Statistical bivariate and univariate analysis was conducted on both datasets for a better understanding of the data patterns of the individual variables, as well as to be made aware of any potential relationships that may exist between the variables of the datasets and the strength of these existing relationships.

- **Model specification** - Both an MLP ANN and classification tree model was built and generated for the forest cover type dataset using a training and test set split of 75% (training set) and 25% (test set). Once the results of the respective models were obtained, they were compared in order to evaluate the accuracies of the built models.

- **Model comparison** - Both models used cross-validation as part of their accuracy measuring strategy. The MLP ANN classifier performed quite well with a 75% accuracy, though with some more fine tuning the performance could be able to be improved upon but at a cost. The tree classification performed in WEKA received an accuracy of 66.4%, 8.08% less than the MLP ANN in performance. Another reason for the MLP ANN performing better than the WEKA classifier is due to the greater amounts of data and features fed into the MLP ANN. Neural networks typically perform better with more data. Though a balancing act needs to be struck with the size of data versus computational efficiency.

- **Model interpretation** - The decision tree is used in classification but more particularly is used when the data is categorical. Multiple models exist and were used in the report to conduct the analysis. The decision tree performed more poorly than its counterpart model, the MLP ANN. It provided satisfactory statistics with it correctly being able to classify 2/3 instances. However, from the results obtained we have noticed that **Elevation** and **ST7201** played the most important roles in determining the leaves in this classification tree. These leaves in turn represent the classified groups of Forest cover insurance.

References

[1] Eibe Frank, Mark A. Hall, and Ian H. Witten. *The WEKA Workbench*. Morgan Kaufmann, 2016.

[2] Python. Online: https://www.python.org/, 2001. Accessed: October 1, 2022.

[3] Paolo Giudici. *Applied data mining: statistical methods for business and industry*. John Wiley & Sons, 2005.

[4] Hand David, Mannila Heikki, and Smyth Padhraic. *Principles of Data Mining*. MIT Press, 2001.