Machine Learning (CS60050)
Assignment 1 Report

Members :

- Pranav Nyati (20CS30037)
- Shreyas Jena (20CS30049)

Question 1: ID3 Decision Tree Algorithm:

- Procedure:
  - We have used the standard ID3 algorithm for training the decision tree on a Customer Segmentation dataset from the file 'Dataset_A.csv'.
  - After training, to prevent overfitting, and increase the accuracy on the test set, we implemented Error-Based Pruning to reduce the number of nodes in the decision tree based on accuracy on the validation set.
  - The procedure of training and pruning was performed 10 times on 10 different random data-splits and the maximum accuracy among all these 10 iterations was obtained.
  - A plot of accuracy vs depth of Decision Tree was obtained for the model with max accuracy for the test data.

- Reading the data and handling missing values:
  - Many attributes in the dataset have NaNs values. Since we are supposed to make a Decision Tree classifier, we considered all the attributes to be categorical.
  - For the attribute **Age**, the age values in all the training points were divided into **10 bins** and each such bin was given a categorical label from **0 to 9**, to reparamatrize the attribute from numerical attribute to a categorical attribute.
  - In all the attributes which had missing values, the missing values were replaced with the mode (attribute value with maximum frequency) value corresponding to that attribute.
  - We proceed to training part after handling the missing values in the data.

- Training and Pruning of Tree on 10 random data-splits
  - We created random splits of the dataset in 10 iterations : In each iteration, the data was split into a **(64 -Train, 16 - Val, 20 - Test)** with different data-points in each of the sets in each iteration, sampled without replacement using the **sample** function in Pandas.
  - In each interaction, the ID3 tree was first trained using **Information-Gain** as the criteria for choosing an appropriate attribute on whose basis to split the data further down the tree.

    Information Gain: For a collection S, $Entropy(S) = -\sum_{i} p(i) log_2 \mathrm{p(i)}$. The information gain is the reduction in entropy after choosing an attribute A. Mathematically,

    $$InformationGain(S,\ A)\ =\ Entropy(S)\ -\ \sum_{v\,\in\, values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

- After training the accuracy of the tree was calculated on the test set (before pruning). Now, using the validation set, the tree is pruned till accuracy increases further for the validation set.
- The test accuracy is measured again after the tree is pruned using **Reduced Error Pruning,** and there is an increase of around **1.5 - 3 %** in the test accuracy after pruning.
  The accuracy is measured as follows:

  $$Accuracy\ =\ \frac{No\ of\ examples\ correctly\ classified}{No\ of\ training\ examples}$$

- The above 3 steps are repeated for the 10 random train-val-test splits and the best accuracy is measured among these iterations, and the corresponding best model is saved.
- The **Test Accuracy (before pruning)**, **Test accuracy (after pruning)**, **Val accuracy(before pruning), Val accuracy(after pruning), the Num of nodes in the tree before and after pruning , the Depth of the tree before and after pruning** are all written in the **A1_Q1_output.txt** file.
- The best accuracy ID3 tree is printed in hierarchical order in the **A1_Q1_BestModel.txt** file. (**Note**: Please note that since the tree is very large , so it was not possible to create a schematic image of the tree using any library).

- Test Accuracy before and after pruning for a run of the code:

Random Split No	**Test Accuracy (Before Pruning)**	**Test Accuracy (After Pruning)**	**No of Nodes (Before Pruning)**	**No of Nodes (After Pruning)**
0	0.39776951672862454	0.419454770755886	4332	3090
1	0.4021065675340768	0.43308550185873607	3961	2657
2	0.42627013630731103	0.43308550185873607	3929	2797
3	0.3946716232961586	0.4163568773234201	3914	2697
4	0.41697645600991323	0.41511771995043373	2743	1904
5	**0.4231722428748451**	**0.4399008674101611**	**3641**	**2548**
6	0.43246592317224286	0.4368029739776952	3845	2522
7	0.42069392812887235	0.42874845105328374	3147	2010
8	0.4138785625774473	0.42874845105328374	3897	2802
9	0.4256505576208178	0.43742255266418834	3185	2241

The best accuracy for that run of code was achieved for iteration 5, and is highlighted in bold.
Thus, for this run, Best Test Accuracy after pruning = **43.99 %**

- Plot of Accuracy vs Depth for the Best Model:
    - The accuracy of the ID3 model is measured as it's depth is varied and a plot is made between Accuracy (in %) on y-axis and Depth of the Tree along x-axis.
    - In the way our ID3 algorithm works during pruning, the Depth of the Tree before Pruning is **9** and even after pruning the Depth remains **9** in all the 10 iterations. Therefore the plot is created for Depths varying from 0 to 9.

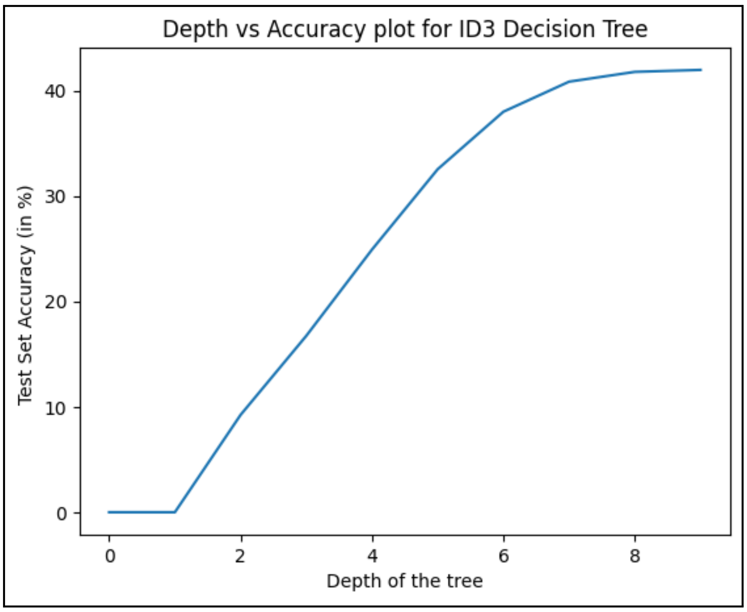

Fig. Accuracy vs Depth Plot for the Best Accuracy Model

Question 2:
Naive Bayes Algorithm

Introduction:

- In the given problem, we implement the Naive Bayes algorithm to classify users into different customer segments based on their personal information.
- The tasks carried out are :
    1. Randomly divide the dataset into 80% train and 20% test sets.
    2. Employ 10-fold cross-validation to train and validate the Naive Bayes classifier. Also, report the test accuracy for the test split.
    3. Implement a Naive Bayes classifier with Laplace smoothing on the same training data and report the corresponding test accuracy.

Theoretical Background:

- We wish to obtain a Naive Bayes classifier that learns a mapping from the users to the customer class.
- For a given user d, the Naive Bayes classifier predicts the class with maximum posterior probability $\hat{c} = argmax$ P(c|d) (For all c ∈ C)
- Naive Bayes classifier makes use of the Bayes theorem to obtain:

  $\hat{c} = argmax$ P(d|c).P(c) (For all c ∈ C)

- If d has the features x1, x2, …, xn, we use the Naive Bayes assumption that each likelihood is independent of others to obtain:

  $\hat{c} = argmax$ P(x1|c).P(x2|c)....P(xn|c).P(c)

Dataset:

- The given dataset contains 8068 rows, each specifying a person, and 11 columns, using a mixture of numerical and categorical features for selecting various information about the given person.
- It possesses the following attributes:

  ID - Unique ID
  Gender - Gender of the customer
  Ever_Married - Marital status of the customer
  Age - Age of the customer
  Graduated - Is the customer a graduate?
  Profession - Profession of the customer
  Work_Experience - Work Experience in years
  Spending_Score - Spending score of the customer
  Family_Size - Number of family members for the customer
  Var_1 - Anonymised Category for the customer
  Segmentation - Customer Segment of the customer

- Data preprocessing was carried out in the following steps:

  1. NaN values in the dataset were replaced by the feature mean for numerical data, and the feature mode for categorical data.
  2. Categorical features were encoded using label encoding.
  3. Outliers for numerical data were decided according to the threshold :

     $$threshold = \mu + 3 \times \sigma$$

     where, μ = Numerical feature mean,
     σ = Numerical feature standard deviation.

     A total of 183 samples with features having outlier values were removed from the dataset, leaving behind 7885 valid samples.
  4. The numerical feature values were normalized to unit mean and variance.

- The remaining data was then subjected to an 80% - 20% train-test split, which gave rise to two datasets with the following sizes:

  - **Train set -** 6308 samples
  - **Test set -** 1577 samples

Training, Validation and Testing:

- The method of 10-fold cross-validation was employed for model training and validation. It involves randomly splitting data into 10 splits, setting aside 1 split for validation and the rest for training.
- Model training involves computing the priors for each output class and the likelihoods for feature values with respect to the output classes.
- For each cross-validation step, the model is applied on the validation split.
- Finally, the trained model was trained on the test split to obtain the test accuracy (without and using Laplace smoothing).

Results:

(Laplace smoothing factor = 1)

Iteration no.	Test accuracy (w/o Laplace smoothing)	Test accuracy (using Laplace smoothing)
1	0.49524	0.50095
2	0.51173	0.50221
3	0.50285	0.49143
4	0.49270	0.50412
5	0.50919	0.48636
6	0.50031	0.48890
7	0.47875	0.49524
8	0.49334	0.50856
9	0.48192	0.47939
10	0.50538	0.49080