

CS60050: MACHINE LEARNING

Assignment 1

Decision Tree Bayesian (Naïve Bayes) Classifier

Group 14

Kulkarni Pranav Suryakant - 20CS30029

Sidharth Vishwakarma - 20CS10082

Dataset

This data set contains a total of 8068 customer details which are categorized into four segments (A, B, C, D). Each customer has the following attribute values (although some attribute values are not available for some customers)

Attribute Name	Definition
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 (including the customer)
Var_1	Anonymised Category for the customer
Segmentation	Customer Segment of the customer

Decision Tree

Tasks

1. Split Dataset A into 80%-20% to form training and testing sets, respectively. Build a Decision Tree Classifier using the ID3 algorithm. Train the classifier using Information Gain (IG) measure (no packages to be used for Decision Tree Classifier).
2. Repeat (1) for ten random splits. Print the best test accuracy and the depth of that tree.
3. Perform reduced error pruning operation over the tree obtained in (2). Plot a graph showing the variation in test accuracy with varying depths. Print the pruned tree obtained in a hierarchical fashion with the attributes clearly shown at each level.
4. Prepare a report including all your results.

Algorithm

- Calculate the entropy of each attribute an in the dataset S.
- Divide the set S into subgroups based on the attribute with the highest information gain.
- Create a decision tree node with that attribute.
- Recursively apply the remaining attributes to subgroups.

Pseudo Code

ID3 (Examples, Target_Attribute, Attributes)

Create a root node for the tree

If all examples are A, Return the single-node tree Root, with label = A.

If all examples are B, Return the single-node tree Root, with label = B.

If all examples are C, Return the single-node tree Root, with label = C.

If all examples are D, Return the single-node tree Root, with label = D.

If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.

Otherwise Begin

A ← The Attribute that best classifies examples.

Decision Tree attribute for Root = A.

For each possible value, v_i , of A,

Add a new tree branch below Root, corresponding to the test $A = v_i$.

Let Examples(v_i) be the subset of examples that have the value v_i for A

If Examples(v_i) is empty

Then below this new branch add a leaf node with label = most common target value in the examples

Else below this new branch add the subtree ID3 (Examples(v_i), Target_Attribute, Attributes – {A})

End

Return Root

Important Terms & Definitions

- Entropy: $\sum_{i=1}^n p_i \times -(\log_2 p_i)$, $i = 1, 2, \dots, n$
 - i is + or - for binary outcomes
 - i varies from A to B in this example
- Information Gain:
 - Gain(S, A): reduction in entropy after choosing attribute A.

$$Entropy(S) = \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Procedure

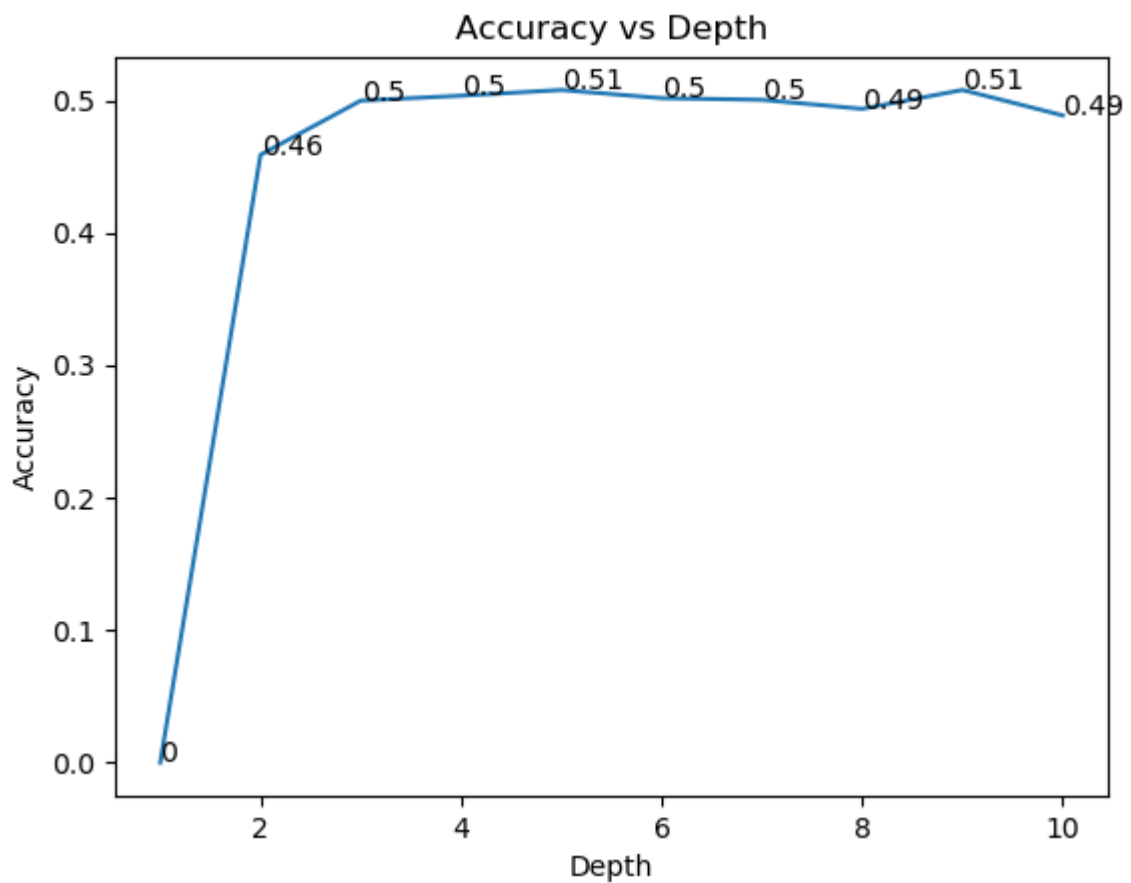
- Missing values in the data are filled with a most common value of that attribute. Continuous attributes like Age and Work_experience are divided into uniform discrete intervals of length of ten.
- The data is split into 80%-20% to form training and testing sets, respectively.
- Decision tree classifier is built on the training dataset using the ID3 algorithm; the classifier is trained using information gain measure.
- Decision tree classifier is then tested on ten random splits. The best test accuracy and depth of the tree is then printed in output.txt.
- Reduced error pruning operation is performed over best tree obtained.
- Relation of test accuracy with depth is observed by varying depth from 1 to 10. (result saved in the file plot.png)

Results

- Testing on ten random splits:

Split No.	Test Accuracy (in %)
1	41.57
2	41.38
3	43.30
4	43.68
5	42.44
6	43.86
7	43.68
8	42.99
9	43.49
10	42.06

- Plot (Accuracy vs Depth):



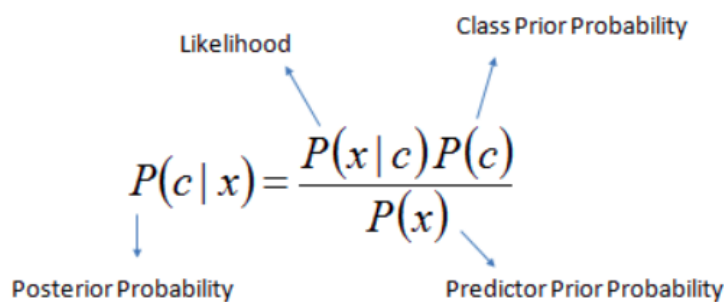
Bayesian (Naïve Bayes) Classifier

A. Tasks

1. Randomly divide Dataset A into 80% for training and 20% for testing. Encode categorical variables using the appropriate encoding method (**in-built function allowed**).
2. A feature value is considered an outlier if its value is greater than mean + 3 x standard deviation ($\mu + 3 \times \sigma$). A sample having maximum such outlier features must be dropped. Print the final set of features formed. Normalize the feature as required.
3. Train the Naïve Bayes Classifier using 10-fold cross validation (**no packages to be used for Naïve Bayes Classifier**). Print the final accuracy.
4. Train the Naïve Bayes Classifier using **Laplace correction** on the same train and test split. Print the final accuracy.
5. Prepare a **report** including all your results.

B. Algorithm

From $P(c)$, $P(x)$, and $P(x|c)$, the posterior probability, $P(c|x)$, may be calculated using the Bayes theorem. The naive Bayes classifier makes the assumption that the impact of a predictor's value (x) on a certain class (c) is unrelated to the values of other predictors. It is known as class conditional independence.



The diagram shows the formula for posterior probability: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from the terms to their labels: $P(x|c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', $P(c|x)$ is labeled 'Posterior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'.

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

- The posterior probability of a class (target) given a predictor is denoted by $P(c|x)$ (attribute).
- The prior probability of the class is $P(c)$.
- The likelihood, or $P(x|c)$, is the probability of a predictor for a particular class.
- $P(x)$ is the predictor's prior probability.

B.Procedure

1. In this assignment, the classifier was designed using the traditional Naive Bayes approach. The training data was sent to us in.csv file. The data has been mixed up. We took into account an 80:20 ratio between the Train and Test set. Regarding five-fold cross-validation, we gave the results. The information's characteristics include **ID, Gender, Ever_Married, Age, Graduated, Profession, Work_Experience, Spending_Score, Family_Size, Var_1**

Target Variable: **Segmentation**

2. The BayesianClassifier class:

Data Members:

- 'smoothing': The variable to determine whether the Naïve Bayes Classifier is using Laplace correction or not (1 if Laplace Correction is used else 0).
- 'features': A NumPy array of the feature of the data on which the model is to be trained.
- 'labels': A NumPy array containing the target value corresponding to the features in the training data.
- 'num_of_classes': The number of unique features in the training data.
- 'num_of_features': The number of features on which the model needs to be trained.
- 'prior_prob': The variable which stores the prior probability.
Prior Probability is given by $P(\mathbf{X})$.
- 'likelihood': A python dictionary containing the values of the likelihood of the features.
The likelihood is given by the formula $P(\mathbf{X} | C_i)$.

Member Functions:

- 'fit': Utility function to train the model on the given value of the train features and labels.
- 'predict': take input a set of test data and according predict the output values about it.
- 'get_prior_prob': Utility function to calculate the prior probability of the predictor.
- 'get_likelihood': Function to calculate the likelihood, which is the probability of predictor given class.
- 'get_marginal_prob': The utility function to calculate the marginal probability, the probability of an event irrespective of the outcomes of other random variables.

3. The Categorical_Encoder class:

Data Members:

- 'Segmentation_dict': The python dictionary to map the categorical value of the 'Segmentation' column to the corresponding number using Ordinal Encoding.

Member Functions:

- 'encode': This function uses the LabelEncoder class of sci-kit-learn, was used for encoding the categorical variables, and uses ordinal encoding for the 'Segmentation' column.

4. Some essential helper functions:

- 'fill_missing_values': The missing values are handled by replacing the most frequent value for categorical attributes and the mean value for the continuous attributes.
- 'outlier_removal': This function removes those samples from the dataset which have their feature value greater than mean + 3 x standard deviation ($\mu + 3 \times \sigma$).
- 'normalize': The function that normalizes the continuous data features following the given formula: **Data feature Value / (Max among all the values - Min among all the values)**.
- 'split_n_folds': Utility function to split the training data into n sections for n-folds cross-validation.
- 'calculate_accuracy': Analyses the predictive values' accuracy in relation to the actual values. The ratio of the number of accurate forecasts to the total number of predictions is known as accuracy.

5. Ten-fold cross-validation:

There are five overall iterations. Each iteration uses the remaining data as the training set and one-fifth of the training set as the validation set. The validation set is used to test the model after it has been trained on the

training set. Using this method, which involves averaging the five aforementioned accuracies, makes it easier to produce a generalization accuracy.

C.Results

1. Dividing the dataset and Encoding the categorical variables:

- Using sci-kit-learn `train_test_split()`, the given data was split randomly into 80%-20% to form training and testing sets, respectively.
- Ordinal encoding was done by first assigning the original order of the variable through a dictionary. Then, mapping each row for the variable as per the dictionary.
- LabelEncoder class of sci-kit-learn was used for encoding the other categorical variables since it does not increase the size of the dataset, unlike the one-hot encoding.

2. Outlier removal and Normalization:

- If a feature value exceeds $\text{mean} + 3 * \text{standard deviation}$, it is regarded as an outlier.
- The selective selection of data approach removes a sample with an outlier characteristic from the dataset.
- The continuous values in the dataset need to be normalized before the actual training of the model to get better results.
- The features pertaining to the continuous values were normalized by dividing them from the max-min of their corresponding values in the whole dataset.

3. Training the model on the train data (without Laplace Correction):

- The model of the Bayesian (Naïve Bayes) Classifier was trained on the training data, which was randomly selected from the dataset given at the scale amount of 80%.
- The final accuracy that the model gave upon testing using the test data which was extracted from the test data (the remaining 20%) was **51.93405199746354%**.

4. Training the model on the train data (with Laplace Correction):

- The model of the Bayesian (Naïve Bayes) Classifier was trained on the training data using Laplace smoothing, which was randomly selected from the dataset given at the scale amount of 80%.
- The final accuracy that the model gave upon testing using the test data which was extracted from the test data (the remaining 20%) was **51.93405199746354%**.

5. Training the Naïve Bayes Classifier using 10-fold cross-validation (without Laplace Correction):

Iteration No.	Test Accuracy (in %)
1	50.06337135614702
2	57.03422053231939
3	49.936628643852976
4	49.17617237008872
5	48.41571609632446
6	49.238578680203043
7	50.38071065989848
8	51.52284263959391

9	47.33502538071066
10	53.29949238578681

Maximum Accuracy of the model using N-Fold Cross Validation (without Laplace smoothing): **57.03422053231939%**

6. Training the Naïve Bayes Classifier using 10-fold cross-validation (with Laplace Correction):

Iteration No.	Test Accuracy (in %)
1	50.06337135614702
2	56.90747782002535
3	49.936628643852976
4	49.049429657794674
5	48.542458808618505
6	49.111675126903553
7	50.50761421319797
8	51.52284263959391
9	47.33502538071066
10	53.29949238578681

Maximum Accuracy of the model using N-Fold Cross Validation (with Laplace smoothing): **56.90747782002535%**

•