CSE343: Machine Learning

Assignment-2

Divyajeet Singh (2021529)

November 5, 2023

1 Section A (Theoretical)

Solution 1

(a) (1 mark)

Solution

A random forest consists of many trees. Correlation refers to the similarity between trees - similar trees capture similar patterns in the data. Diversity, on the other hand, refers to the difference in the trees. With a large number of trees, we get a balanced prediction from the forest, which is able to capture most of the patterns in the data and make better predictions. However, if the trees are too diverse and have very little correlation, this could mean that the bigger patterns are not being effectively covered. This can reduce the performance of the trees. So, it is important to maintain some diveristy among a forest of correlated trees.

(b) (1 mark)

Solution

Naïve Bayes works by predicting the class labels that maximize the posterior probability using Bayes' Theorem, assuming that the features are independent of each other.
With the curse of dimensionality (excessive number of features), the dataset in consideration can be very sparse. Naïve Bayes performs poorly on sparse datasets, since it predicts a 0 probability for a feature that it has not seen before. Moreover, it assumes that all feautres are weighed equally, which can cause issues with its performance with a large number of features. To solve this problem, one can use dimensionality reduction techniques like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), and smoothing techniques like Laplace smoothing to prevent 0 probabilities.

(c) (1 mark)

Solution

When Naïve Bayes encounters a value of an attribute that it has not seen before, it assigns a 0 probability to it. This clearly affects the performance of the model adversely as it can return a probability of 0 for inputs having large probabilities for other attributes, thereby increasing its variance. For example, consider the dataset in Table 1.
It is obvious that the target variable depends solely on the Weather. However, the Naïve Bayes model would predict the class **0** for an input vector `[Cool, Mid]` since it has not seen

Weather	Humidity	Will-play
Hot	High	**0**
Hot	Low	**0**
Cool	High	**1**
Cool	Low	**1**

Table 1: A sample dataset for Naïve Bayes

the Humidity attribute before. This effect can be avoided by using smoothing techniques like Laplace smoothing and m-estimate, which add a small positive value to the probability of each attribute, even for the features that are not present in the datset. This prevents an unwanted **0** as the probability of a class.

(d) (1 mark)

Solution

Yes, the cardinality of the attributes affects the splitting of a tree node when using Information Gain. This means that for two feaures that produce equally good splits, the one with the higher cardinality will be chosen.
An alternative way is to use Gain Ratio instead of Information Gain to choose splits. This is a normalized version of Information Gain, which takes into account the cardinality of the attributes. This ensures that the attribute with the higher cardinality is not favoured over the other.

$$\textsc{Gain-ratio} = \frac{\textsc{Info-gain}}{\textsc{Split-Info}}$$

where

$$\textsc{Split-info} = -\sum_{i=1}^{n} \frac{N(t_i)}{N_t} \log_2 \frac{N(t_i)}{N_t}$$

For example, let’s say a dataset consists of 90 unique ages and 10000 unique Customer IDs. Then, the attribute Customer ID will be favoured over Age when using Information Gain, even though Customer ID cannot be a good indicator of the target variable.

Solution 2

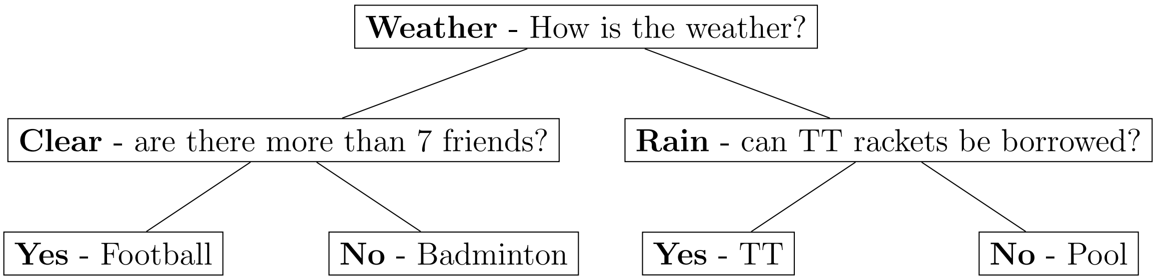
(a) (1 mark)

Solution

Let the random variable R represent ”Can borrow racket?”, F reprsent ”$>$ 7 friends?”, and W represent ”Weather”. Then, there are four possible outcomes in the given problem. These are listed with their conditional probability expressions in Table 2. The decision tree is given in Figure 1.

Outcome (Y)	Conditional Probability
`TT (T)`	$\mathbb{P}[Y = \texttt{T} \mid R = \texttt{Yes}] * \mathbb{P}[R = \texttt{Yes} \mid W = \texttt{Rain}] * \mathbb{P}[W = \texttt{Rain}]$
`Pool (P)`	$\mathbb{P}[Y = \texttt{P} \mid R = \texttt{No}] * \mathbb{P}[R = \texttt{No} \mid W = \texttt{Rain}] * \mathbb{P}[W = \texttt{Rain}]$
`Football (F)`	$\mathbb{P}[Y = \texttt{F} \mid F = \texttt{Yes}] * \mathbb{P}[F = \texttt{Yes} \mid W = \texttt{Clear}] * \mathbb{P}[W = \texttt{Clear}]$
`Badminton (B)`	$\mathbb{P}[Y = \texttt{B} \mid F = \texttt{No}] * \mathbb{P}[F = \texttt{No} \mid W = \texttt{Clear}] * \mathbb{P}[W = \texttt{Clear}]$

Table 2: Conditional Probabilities for the given problem


Weather - How is the weather?
Clear - are there more than 7 friends?
Rain - can TT rackets be borrowed?
Yes - Football
No - Badminton
Yes - TT
No - Pool

(b) (1 mark)

Solution

Let us define two random variables P and T to represent the prediction of the app and the truth respectively. Both of them can take values from the set $\{\texttt{R}, \texttt{C}\}$, where `R` and `C` represent `Rain` and `Clear` respectively.
We are given that the app correctly predicts rainy days with an 80% accuracy and clear days with a 90% accuracy. This means that

$$\mathbb{P}[P = \texttt{R} \mid T = \texttt{R}] = 0.8 \implies \mathbb{P}[P = \texttt{C} \mid T = \texttt{R}] = 0.2 \tag{1}$$

$$\mathbb{P}[P = \texttt{C} \mid T = \texttt{C}] = 0.9 \implies \mathbb{P}[P = \texttt{R} \mid T = \texttt{C}] = 0.1 \tag{2}$$

We are also given that the app predicts rain on 30% of the days. This means

$$\mathbb{P}[P = \texttt{R}] = 0.3 \implies \mathbb{P}[P = \texttt{C}] = 0.7 \tag{3}$$

Let us find the probability that it rains on a day when the app predicts rain.

$$\begin{aligned} \mathbb{P}[T = \texttt{R} \mid P = \texttt{R}] &= \frac{\mathbb{P}[P = \texttt{R} \mid T = \texttt{R}] * \mathbb{P}[T = \texttt{R}]}{\mathbb{P}[P = \texttt{R}]} && (4) \\ &= \frac{0.8 * \mathbb{P}[T = \texttt{R}]}{0.3} && (5) \end{aligned}$$

To find the probability that it rains on a day, we use the law of total probability on Equation (3). Note that $\{T = \texttt{R}\}$ and $\{T = \texttt{C}\}$ are complementary events.

$$\begin{aligned} \mathbb{P}[P = \texttt{R}] &= \mathbb{P}[P = \texttt{R} \mid T = \texttt{R}]\ \mathbb{P}[T = \texttt{R}] + \mathbb{P}[P = \texttt{R} \mid T = \texttt{C}]\ \mathbb{P}[T = \texttt{C}] && (6) \\ 0.3 &= 0.8 * \mathbb{P}[T = \texttt{R}] + 0.1 * (1 - \mathbb{P}[T = \texttt{R}]) && (7) \\ 0.3 &= 0.1 + 0.7 * \mathbb{P}[T = \texttt{R}] && (8) \\ \implies \mathbb{P}[T = \texttt{R}] &= \frac{0.2}{0.7} && (9) \\ & && (10) \end{aligned}$$

Substituting this value in Equation (5), we get

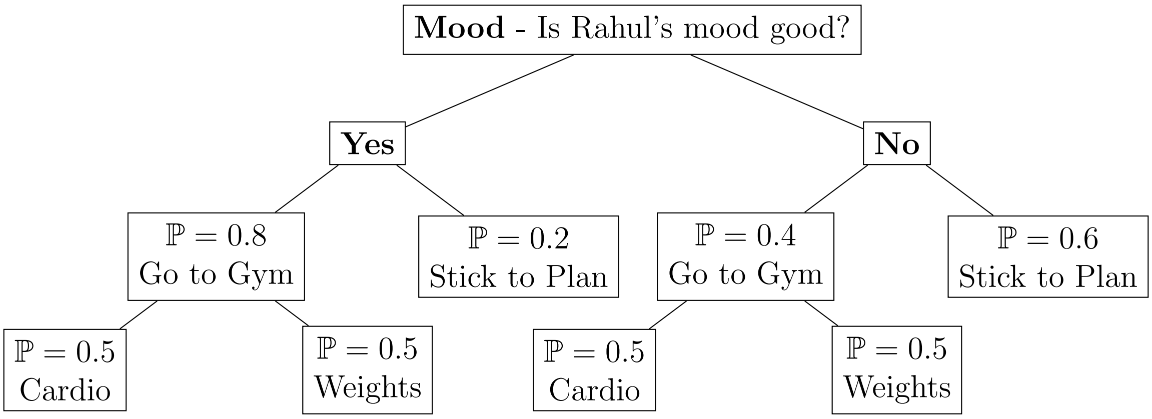
$$\mathbb{P}[T = \texttt{R} \mid P = \texttt{R}] = \frac{0.8 * \frac{0.2}{0.7}}{0.3} = \frac{16}{21} \approx 0.7619 \tag{11}$$

Therefore, the required probability is approximately 0.762.

(c) (1.5 marks)

Solution

In the updated question, Rahul decides whether to play or go to the gym based on his mood. So, the updated decision tree is given in Figure 1.


Mood - Is Rahul's mood good?
Yes
No
$\mathbb{P} = 0.8$ Go to Gym
$\mathbb{P} = 0.2$ Stick to Plan
$\mathbb{P} = 0.4$ Go to Gym
$\mathbb{P} = 0.6$ Stick to Plan
$\mathbb{P} = 0.5$ Cardio
$\mathbb{P} = 0.5$ Weights
$\mathbb{P} = 0.5$ Cardio
$\mathbb{P} = 0.5$ Weights

Let M represent Rahul's mood, which can take values from $\{\texttt{G}, \texttt{B}\}$, where `G` and `B` represent `Good` and `Bad` respectively. Let A represent the chosen activity, which can take values from $\{\texttt{Gym}, \texttt{Stick-to-plan}\}$. Then, the list of outcomes along with the expressions for their conditional probabilities is given below.

1. Stick to Plan[1] (`S`)

$$\begin{aligned}&\mathbb{P}[Y = \texttt{S} \mid M = \texttt{G}] * \mathbb{P}[M = \texttt{G}] + \mathbb{P}[Y = \texttt{S} \mid M = \texttt{B}] * \mathbb{P}[M = \texttt{B}]\\ &= 0.2 * \mathbb{P}[M = \texttt{G}] + 0.6 * \mathbb{P}[M = \texttt{B}]\end{aligned}$$

2. Do Cardiological Exercises (`C`)

$$\begin{aligned}&\mathbb{P}[Y = \texttt{C} \mid A = \texttt{Gym}] * \mathbb{P}[A = \texttt{Gym} \mid M = \texttt{G}] * \mathbb{P}[M = \texttt{G}] +\\ &\mathbb{P}[Y = \texttt{C} \mid A = \texttt{Gym}] * \mathbb{P}[A = \texttt{Gym} \mid M = \texttt{B}] * \mathbb{P}[M = \texttt{B}]\\ &= 0.5 * 0.8 * \mathbb{P}[M = \texttt{G}] + 0.5 * 0.4 * \mathbb{P}[M = \texttt{B}]\end{aligned}$$

3. Do Weight Training (`W`)

$$\begin{aligned}&\mathbb{P}[Y = \texttt{W} \mid A = \texttt{Gym}] * \mathbb{P}[A = \texttt{Gym} \mid M = \texttt{G}] * \mathbb{P}[M = \texttt{G}] +\\ &\mathbb{P}[Y = \texttt{W} \mid A = \texttt{Gym}] * \mathbb{P}[A = \texttt{Gym} \mid M = \texttt{B}] * \mathbb{P}[M = \texttt{B}]\\ &= 0.5 * 0.8 * \mathbb{P}[M = \texttt{G}] + 0.5 * 0.4 * \mathbb{P}[M = \texttt{B}]\end{aligned}$$

(d) (1.5 marks)

Solution

We are given the following (here, F repsents the number of hours of sleep Rahul got the previous night)

$$\mathbb{P}[M = \texttt{G}] = 0.6 \qquad \mathbb{P}[M = \texttt{B}] = 0.4 \tag{12}$$

$$\mathbb{P}[F = 7 \mid M = \texttt{G}] = 0.7 \qquad \mathbb{P}[F = 7 \mid M = \texttt{B}] = 0.45 \tag{13}$$

First, we find out the probability that Rahul is in a good mood, given he slept for 7 hours the previous night.

$$\mathbb{P}[M = \texttt{G} \mid F = 7] = \frac{\mathbb{P}[F = 7 \mid M = \texttt{G}] * \mathbb{P}[M = \texttt{G}]}{\mathbb{P}[F = 7 \mid M = \texttt{G}] * \mathbb{P}[M = \texttt{G}] + \mathbb{P}[F = 7 \mid M = \texttt{B}] * \mathbb{P}[M = \texttt{B}]} \tag{14}$$

$$= \frac{0.7 * 0.6}{0.7 * 0.6 + 0.45 * 0.4} = \frac{42}{60} = 0.7 \tag{15}$$

[1] This option is essentially the tree given in Figure 1, starting from the root. We calculate the probability of this outcome as a whole - as will be seen next, this will not change the answer since the probability of this tree as a whole will be less than the most likely activity.

Now, we simply substitute the value of $\mathbb{P}[M = \texttt{G} \mid F = 7]$ in the conditional probabilities of the outcomes to get the required probabilities. Note that Rahul's mood being good or bad are complementary events.

1. Stick to Plan: $\mathbb{P}[Y = \texttt{S}] = 0.2 * 0.7 + 0.6 * 0.3 = 0.32$
2. Do Cardiological Exercises: $\mathbb{P}[Y = \texttt{C}] = 0.5 * 0.8 * 0.7 + 0.5 * 0.4 * 0.3 = 0.34$
3. Do Weight Training: $\mathbb{P}[Y = \texttt{W}] = 0.5 * 0.8 * 0.7 + 0.5 * 0.4 * 0.3 = 0.34$

Since the probability of sticking to the original plan of playing games is already lower than the probabilities of the other two outcomes, the answer does not change. Since the "Stick to Plan" option is further divided in the decision tree in Figure 1, the probability of the outcomes of the four games are naturally smaller than the probability of the "Stick to Plan" option as a whole. This means that they cannot be the most likely outcome.
Hence, the most likely outcome is to do either cardiological exercises or weight training.

2 Section B (Library Implementation)

This section required us to use the Decision Tree and Random Forest Classifiers in Python to predict the presence of heart diseases using the UCI Heart Disease Dataset. The `scikit-learn` package was used for this problem. The solution can be found in the `main.ipynb` notebook.

Best Criterion for Splitting in Decision Trees

An exploratory data analysis was performed on the dataset. For the given dataset, I averaged the performance of D-Trees trained using both criterions over more than 10,000 runs. It was clear that (both for the binary and multiclass classification), the Gini Impurirt criterion performed better than the Entropy criterion. The results are shown in Table 3.

Criterion	Mean Accuracy
Gini Impurity	0.8133334
Entropy	0.7483777

Table 3: Average performance of decision trees using different splitting criteria

Optimal Decision Tree

Grid search was performed using the `GridSearchCV` class in `scikit-learn` to find the optimal hyperparameters for the decision tree. The optimal hyperparameters were found to be `max_features` = `sqrt` and `min_samples_split` = 6.
The accuracy score of the decision tree using these hyperparameters is 0.834.

Optimal Random Forest

A random forest classifier using the same "Gini Impurity" criterion for splitting was trained using the optimal hyperparameters that are found using grid search over a large hyperparameter space. The optimal hyperparameters were found to be `n_estimators` = 300, `max_depth` = 7, and `min_samples_split` = 2.
The accuracy score of the random forest using these hyperparameters is 0.8834. A classification report is also presented in the main notebook.

3 Section C (Algorithm Implementation using Packages)

This section required us to implement a decision tree classifier from scratch using the `numpy` package in a class called `MyDecisionTree` with some specific functionaity. The implementation can be found in `utils.py`. The implementation was then tested on the given dataset of Thyroid patients.

Results

The implemented classifier was tested on the given dataset to perform binary classificiation. The results are shown in Table 4.

Criterion	Accuracy Score
Gini Impurity	0.990706
Entropy	0.986988

Table 4: Performance of the manual implementation of the decision tree classifier

References

1. The Curse of Dimensionality
2. UCI Machine Learning GitHub Repository