**K173795**

**Muhammad Mustafa Manga**

**LAB-6**

**Using 10 x 10 Fold CV**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 |
| DT using gini  (without pruning) | 0.01136 | 0.03831 | 0.10805 | 0.02915 | 0.000575 |
| DT using gini  (with pruning) | 0.01370 | 0.03424 | 0.10198 | 0.03615 | 0.001089 |
| DT using entropy  (without pruning) | 0.00618 | 0.02157 | 0.06634 | 0.01482 | 0.00032 |
| DT using entropy  (with pruning) | 0.02025 | 0.03366 | 0.10723 | 0.03459 | 0.00108 |

**Advantages:**

* K-fold cross-validation works well on small and large data sets.
* All of our data is used in testing our model, thus giving a fair, well-rounded evaluation metric.
* K-fold cross-validation may lead to more accurate models since we are eventually utilizing our data to build our model.

**Disadvantages:**

* The computing power is high.
* So it may take some time to get feedback on the model’s performance in the case of large data sets.
* Slower feedback makes it take longer to find the optimal hyper parameters for the model.

**Using 70% Holdout approach repeated 100 times**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Dataset1 | Dataset2 | Dataset3 | Dataset4 | Dataset5 |
| DT using gini  (without pruning) | 0.00756 | 0.01996 | 0.06815 | 0.01478 | 0.000348 |
| DT using gini  (with pruning) | 0.01109 | 0.02224 | 0.06825 | 0.01691 | 0.00055 |
| DT using entropy  (without pruning) | 0.00624 | 0.02073 | 0.66285 | 0.01505 | 0.00034 |
| DT using entropy  (with pruning) | 0.01523 | 0.02253 | 0.06565 | 0.02114 | 0.00055 |

**Advantages:**

1. Simplest method
2. Easily can work on large data
3. Fast method as compared to other method

**Disadvantages:**

* Not working for small data set.(here it comes the Role of K-Fold Cross validation)

**Hold-out vs. Cross-validation**

* Cross-validation is usually the preferred method because it gives your model the opportunity to train on multiple train-test splits. This gives you a better indication of how well your model will perform on unseen data. Hold-out, on the other hand, is dependent on just one train-test split. That makes the hold-out method score dependent on how the data is split into train and test sets.
* The hold-out method is good to use when you have a very large dataset, you’re on a time crunch, or you are starting to build an initial model in your data science project. Keep in mind that because cross-validation uses multiple train-test splits, it takes more computational power and time to run than using the holdout method.