Assignment 2

● Graded

Group

HARSH GUPTA
PRIYANKA ARORA
PARNIKA MITTAL
...and 3 more
View or edit group

Total Points

28 / 40 pts

Question 1

Calculations

10 / 10 pts

✓ **+ 10 pts** Description of how the model was constructed

– 2 pts Insufficient or missing details

+ 0 pts Completely wrong or else unanswered.

Question 2

Code

18 / 30 pts

+ 0 pts No rubric -- see comments for mark breakup

+ 18 pts GROUP NO: 14

Grading scheme for code:
Train time tt (in sec): tt < 0.01 (10 marks), 0.01 <= tt < 0.1 (9 marks), 0.1 <= tt < 1 (8 marks), tt >= 1 (7 marks)
Model size ms (in KB): ms < 100KB (5 marks), 100 <= ms < 500 KB (4 marks), ms >= 500 KB (3 marks)
Evaluation time et (in sec): et < 0.01 (15 marks), 0.01 <= et < 0.1 (14 marks), 0.1 <= et < 1.0 (13 marks), et >= 1.0 (12 marks)
Marks multiplier mm: (1 + prec) / 2
Final marks: ceil(mm * (tt marks + ms marks + et marks))

tt = 0.504 sec : 8 marks
ms = 15568.153 KB : 3 marks
et = 0.508 sec : 13 marks
Precision: 0.454 i.e. Multiplier (mm): 0.727
TOTAL: 18 marks

Question assigned to the following page: 1

Data Innovators - Report for Assignment 2 - CS771

1. Kshitij Bhardwaj - 230580

2. Harsh Gupta - 230445

3. Priyanka Arora - 230799

4. Parnika Mittal - 230736

5. Harshit Agarwal - 230458

Abstract

This report describes the development of a decision tree model for word prediction using bigrams (adjacent character pairs) as features. The model aims to predict a single word based on a list of bigrams. The report outlines the design decisions taken for the model, including feature engineering, the decision tree algorithm, and hyperparameters.

1 Decision Tree Algorithm

1.1 Splitting Criterion - Entropy

1. Precision on Gini Criterion: 0.3783626862783046
2. Precision on Entropy Criterion: 0.4337139539384556

Why Entropy Might Be More Suitable for This Specific Bigram Prediction Problem?

Entropy: Measures the overall uncertainty or randomness in a dataset. Higher entropy indicates more uncertainty about the correct class (word) for a given bigram feature.
Gini Impurity: Focuses on the probability of misclassifying a sample if randomly drawn from a node. Lower Gini impurity implies a better separation between classes.

Reasoning for Better Performance with Entropy:

1. Bigram Ambiguity: Bigrams might be ambiguous and not uniquely identify a specific word. For example, the bigram "th" could appear in words like "the," "this," "thing," etc.
2. Entropy Captures Ambiguity: Entropy inherently penalizes uncertainty and aims to distribute samples across all possible classes (words) for a given bigram. This can be beneficial when dealing with ambiguous features like bigrams.

Preprint. Under review.

Question assigned to the following page: 1

Imagine a dataset with bigrams "th" and "yi" and three words: "the" (2 occurrences), "thing" (1 occurrence), and "playing" (1 occurrence).

$$Gini = 1 - \sum_{i=1}^{C}(p_i)^2 \tag{1}$$

Gini Impurity Calculation:
- Node with bigram "th": Classes (words) - "the" (2) and "thing" (1).
- Gini impurity = $1 - (0.64^2 + 0.36^2) = 0.384$
- Node with bigram "yi": Class (word) - "playing" (1).
- Gini impurity = $1 - 1^2 = 0$

$$E(S) = \sum_{i=1}^{c} -p_i log_2(pi) \tag{2}$$

Entropy Calculation:
- Node with bigram "th": Entropy = - (2/4 * log2(2/4) + 1/4 * log2(1/4)) = 0.811
- Node with bigram "yi": Entropy = - (1/1 * log2(1/1)) = 0
- Overall Entropy (weighted average) might be slightly higher than Gini impurity.

1.2 Stopping Criterion

There are two main criteria that determine when to stop expanding a decision tree and make a node a leaf:

1. Purity of the Node :‘**max_depth = None**‘

This criterion checks if all samples reaching a particular node in the tree belong to the same class (category). If so, the node is considered pure, and there’s no need for further splitting. This is because the model can already make a confident prediction (the class) for any sample that reaches that node. To maximise this parameter usefulness, we set the ’max_depth = None’
If None, then nodes are expanded until all leaves are pure or until all leaves contain less than ’min_samples_split’ samples.

2. Minimum Samples Per Leaf : ‘**min_samples_leaf = 1**‘

This hyperparameter sets a threshold for the minimum number of samples allowed in a leaf node. Expanding the tree further might be pointless if there are very few samples remaining in a node. With too few samples, the decision tree might learn overly specific rules that don’t generalize well to unseen data (underfitting).

Parameter Value	1	2	4
Precision	0.4256	0.1661	0.0689

The Relative gain in Model Size Reduction and Time Taken, does not override the loss in precision.

1.3 Pruning : None

Our Problem statement does not require external Pruning strategies, it well fits the data provided to output a respectable accuracy.

Question assigned to the following page: 1

1.4 Other Hyperparameters

1. **min_samples_split : 2**
With a higher min_samples_split, the overall dataset might have more unique bigrams and next words, potentially leading to a higher initial entropy (H) before splitting.

Although there are chances that Information Gain is better overall after 3 layers or so, the precision always decreases on increasing the value.

Parameter Value	2	3	4
Precision	0.4337	0.2721	0.2106

2 References

1. scikit-learn DecisionTreeClassifier Documentation
2. Entropy, Info Gain, Gini Index and CCP Pruning Article
3. Article on Classification Trees Pruning Optimization with GridSearchCV and Cost Complexity Fn.