ALGORITHM PERFORMANCE EVALUATION MEASURES BY CALCULATING ACCURCY USING DIFFERENT DATASET

Shanmukha Sudha Kiran.T-**211FA04003**, Sairam.P - **211FA04284**,

Sripujitha .S -**211FA04289**, Vamsi.G - **211FA04358**

BATCH-07,3Rd B. TECH, CSE BRANCH, VFSTR DEEMED TO BE UNIVERSITY

ABSTRACT:

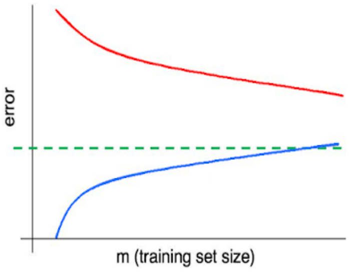
In this paper, we commence by introducing the concept of accuracy, which serves as a pivotal metric for evaluating algorithmic performance. Our focus is centered on a singular dataset, where in we apply a spectrum of algorithms, including Decision Trees, Naive Bayes, FindS, K-Means, and others. Through rigorous evaluation, we determine the Decision Tree is the optimal algorithm based on the highest accuracy achieved.

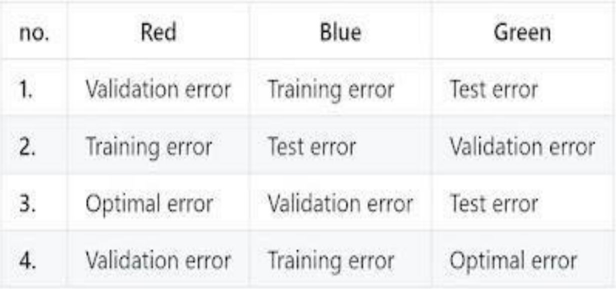
PROBLEM STATEMENT:

a. Your supervisor asks you to create a machine learning system that will help your human resources department classify jobs applicants into well-defined groups. What type of system are you more likely to recommend?

b. Your company wants to predict whether existing automotive insurance customers are more likely to buy homeowners insurance. It created a model to better predict the best customers contact about homeowners insurance, and the model had a low variance but high bias. What does that say about the data model?

c. Suppose you are working machine-learning algorithm to classify/cluster the data samples of any kind of applications. You train a learning algorithm, and find that ithas unacceptably high error on the train set. You plot the learning curve, and obtain the figure below. Identify which kind of errors from the following graph.





no.	Red	Blue	Green
1.	Validation error	Training error	Test error
2.	Training error	Test error	Validation error
3.	Optimal error	Validation error	Test error
4.	Validation error	Training error	Optimal error

INTRODUCTION:

ACCURACY:

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition: Accuracy = Number of correct predictions Total number of predictions.

Accuracy = (true positives + true negatives) / (total examples)

DECISION TREE:

A decision tree is one of the most powerful tools of supervised learning algorithms used for both classification and regression tasks. It builds a flowchart-like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

FINDS:

The find-S algorithm is a basic concept learning algorithm in machine learning. The find-S algorithm finds the most specific hypothesis that fits all the positive examples. The find-S algorithm starts with the most specific hypothesis.

NAVIE BAYERS:

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

DATASET:

Origin, Manufacturer, Color,Year,Type,Class

Jp,Honda,Blue,1980,Eco,Yes

Jp,Toyota,Green,1970,Spo,No

Jp,Toyata,Blue,1990,Eco,Yes

Usa,Audi,Red,1980,Eco,No

Jp,Honda,White,1980,Eco,Yes

Jp,Toyata,Green,1980,Eco,Yes

Jp,Honda,Red,1980,Eco,No

ALGORITHM:

//Here we got more accuracy for decision tree so make document on decision tree algorithm

1.Start

2.Import required libraries, including DecisionTreeClassifier, export_text, plot_tree, LabelEncoder, accuracy_score, and train_test_split.

3.Define sample data representing car information in lists for features like 'origin', 'manufacturer', 'color', 'year', 'type', and 'target_class'.

4.Apply label encoding to categorical features using LabelEncoder() from scikit-learn.

5.Combine encoded features into a feature matrix X after encoding.

6.Encode the target variable.

7.Initialize DecisionTreeClassifier with the criterion set to 'entropy' and fit it with the feature matrix X and the target variable y.

8.Print the rules of the decision tree using export_text().

9.Visualize the decision tree using plot_tree().

10.Split the data into training and testing sets using train_test_split().

111.Make predictions on the test set using the trained classifier.

12.Calculate the accuracy of the model using accuracy_score().

13.Print the accuracy of the model on the test set.

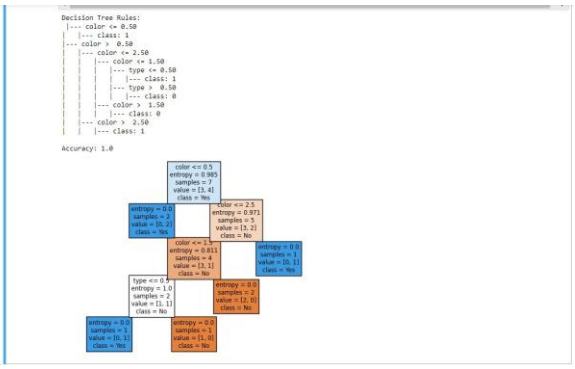
14.Stop

SOURCE CODE:

```
from sklearn.tree import DecisionTreeClassifier, export_text, plot_tree

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

origin = ['Jp', 'Jp', 'Jp', 'Usa', 'Jp', 'Jp', 'Jp']

manufacturer = ['Honda', 'Toyota', 'Toyata', 'Audi', 'Honda', 'Toyata', 'Honda']

color = ['Blue', 'Green', 'Blue', 'Red', 'White', 'Green', 'Red']

year = [1980, 1970, 1990, 1980, 1980, 1980, 1980]

type = ['Eco', 'Spo', 'Eco', 'Eco', 'Eco', 'Eco', 'Eco']

target_class = ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No']

label_encoder = LabelEncoder()

origin_encoded = label_encoder.fit_transform(origin)

manufacturer_encoded = label_encoder.fit_transform(manufacturer)

color_encoded = label_encoder.fit_transform(color)

type_encoded = label_encoder.fit_transform(type)

target_class_encoded = label_encoder.fit_transform(target_class)

X = list(zip(origin_encoded, manufacturer_encoded, color_encoded, year, type_encoded))

y = target_class_encoded



clf = DecisionTreeClassifier(criterion='entropy')

clf.fit(X, y)

tree_rules = export_text(clf, feature_names=['origin', 'manufacturer', 'color', 'year', 'type'])

print("Decision Tree Rules:\n", tree_rules)

plot_tree(clf, feature_names=['origin', 'manufacturer', 'color', 'year', 'type'], class_names=label_encoder.classes_, filled=True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
```

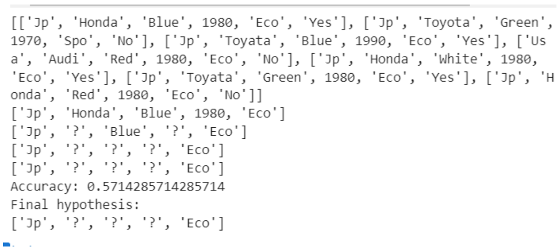
OUTPUTS:

Decision Tree:



```
Decision Tree Rules:
 |--- color <= 0.50
 |   |--- class: 1
|--- color >  0.50
|   |--- color <= 2.50
|   |   |--- color <= 1.50
|   |   |   |--- type <= 0.50
|   |   |   |   |--- class: 1
|   |   |   |--- type >  0.50
|   |   |   |   |--- class: 0
|   |   |--- color >  1.50
|   |   |   |--- class: 0
|   |--- color >  2.50
|   |   |--- class: 1

Accuracy: 1.0
```

Find-S:



```
[['Jp', 'Honda', 'Blue', 1980, 'Eco', 'Yes'], ['Jp', 'Toyota', 'Green',
1970, 'Spo', 'No'], ['Jp', 'Toyata', 'Blue', 1990, 'Eco', 'Yes'], ['Us
a', 'Audi', 'Red', 1980, 'Eco', 'No'], ['Jp', 'Honda', 'White', 1980,
'Eco', 'Yes'], ['Jp', 'Toyata', 'Green', 1980, 'Eco', 'Yes'], ['Jp', 'H
onda', 'Red', 1980, 'Eco', 'No']]
['Jp', 'Honda', 'Blue', 1980, 'Eco']
['Jp', '?', 'Blue', '?', 'Eco']
['Jp', '?', '?', '?', 'Eco']
['Jp', '?', '?', '?', 'Eco']
Accuracy: 0.5714285714285714
Final hypothesis:
['Jp', '?', '?', '?', 'Eco']
```

RESULT:

For give dataset after applying various algorithm ,Accuracy was:

Decision Tree- 1

Navie Bayers- 0.83

Find-S -   0.57

REFERENCE:

https://www.datacamp.com/tutorial/decision-tree-classification-python

https://www.edureka.co/blog/find-s-algorithm-in-machine-learning/

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

CONCLUSION:

In conclusion, the evaluation measures reveal that the decision tree algorithm is having more accuracy when compared to other algorithms.