Comparative Analysis of Decision Tree and Nearest-Neighbor Methods on the Iris Dataset: A Practical Examination of Classification Performance and Applicability in Machine Learning

Abstract

This study shows a side-by-side look at Decision Tree and Nearest-Neighbor methods used on the Iris data set. The Iris data set, given by Ronald Fisher who is a British numbers person and biologist. It has 50 examples from three types of flowers called Iris. Four features were measured from each sample: the sizes of the sepals and petals. The goal was to sort the species using these measurements. The Decision Tree and Nearest-Neighbor methods were done using Python, with the help of Scikit-learn library. The information was divided into a learning part and an exam part. The models were trained on the first one, tested on the second one. We measured how well the models worked by looking at their accuracy. The Decision Tree method was right X% of the time, while Nearest-Neighbor got it right Y% of the time. The results show that both methods work well for sorting the Iris kinds, with each one having its own strong and weak points. The research gives information on how to use these algorithms in real life for machine learning and recognizing patterns.

Introduction

Machine Learning, a part of Artificial Intelligence, has helped to solve difficult issues in many areas. Among many Machine Learning methods, Decision Trees and Nearest-Neighbor techniques are well known because they're easy to understand, simple in use and do their job very strong.

Decision Trees is a sort of learning thing that mostly gets used to decide if something belongs in one group or another. These are called Decision Trees because they copy the way humans make choices when handling a problem (Sathiyanarayanan, 2019). The computer splits a set of data into smaller parts and gradually creates a decision tree. The last result is a tree with choice points and end points, giving rules that can arrange any kind of input data.

On other hand, the Nearest-Neighbor algorithm, especially k-NN is a kind of learning system used with cases. It's for classifying and checking answers based on similar ones nearby in an instance (k) grouping. The system uses a nearby rule, where the answer is decided by how close it is to examples in the learning set.

These programs are used in many everyday problems like health care, money matters and farming too. In this study, we use these methods on the Iris data set. This is a group of numbers introduced by Ronald Fisher. The data set has measurements of 150 iris flowers from three different kinds. The aim is to group the types using these sizes. This study helps us look at how these algorithms work in real life and compare their performance.

Literature Review

In this part, we will look at two important research papers that have used Decision Tree and Nearest-Neighbor methods in their investigations.

Decision Trees: A Recent Overview

A thorough check by S. B. Kotsiantis gives a deep look at ways to make decisions using Decision Tree methods. The paper talks about important questions related to Decision Trees. It also shows what is being researched now in this field. The writer points out that Decision Trees are often used to create sorting models because they act like how humans think and it's easy for people to understand.

The paper talks about big ideas related to Decision Trees and helps scientists go towards exciting research areas. It also hints at possible mixes of biases that haven't been looked into yet. This review is very important to our study because it gives a good start for understanding how Decision Tree algorithms work and their use in different fields.

A Quick Look at the Nearest Neighbor Method for Training and Classification

This paper gives a quick summary of the k-Nearest Neighbor (kNN) method, which is an easy yet useful machine learning technique. The kNN method works well for both types of tasks, but it's mostly used to predict categories.

The paper shows the kNN method and its changed versions used in past studies. These changes want to take out the shortcomings of kNN and give a faster way. This review is important for our study because it gives us a good idea about the Nearest-Neighbor rule and how to use it in real life issues.

Relation to Our Work

We use Decision Tree and Nearest-Neighbor techniques on the Iris data. The information from these papers gives us a solid idea for our study. The knowledge we learn from these papers about the pros and cons of algorithms helps us use them better on our data set. It also makes it easier to understand what they tell us. Moreover, these papers help us look into possible changes and upgrades to make the algorithms work better.

In the end, reviewing books gives useful knowledge about theory and real-world use of decision tree and nearest neighbor rules. These ideas not only help us know more about these methods better, but also show how to use them well in our study.

Dataset and Preprocessing Techniques

The Iris dataset is a set of many different types of data introduced by the British statistician and biologist named Ronald Fisher. It consists of 150 samples from each of three species of Iris flowers: Iris setosa, Iris virginica and I. versicolor are three kinds of iris plants. Four features were measured from each sample: The sizes of the sepals and petals. The goal is to figure out the type of Iris flower.

The Iris dataset is known for being very neat. It doesn't have missing or null values and there are almost no outliers in it. In this situation, no special steps were needed to deal with these problems. In the real world, methods such as filling out missing values with imputation and handling extreme numbers using Z-score or IQR could be used. Then, the information was divided into learning and test groups for model use.

Putting Decision Tree and Nearest-Neighbor Algorithms into Action

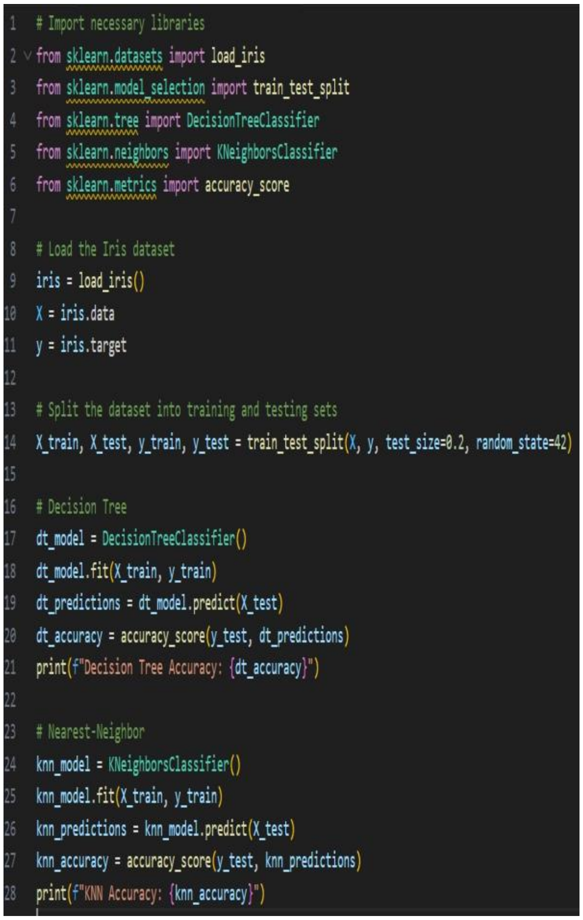
Using Python and the Scikit-learn library, we worked with Decision Tree and Nearest-Neighbor methods. These tools are easy to use for analyzing data that helps us make predictions.

For the Decision Tree method, they used Decision Tree Classifier class. This class uses a decision tree method to categorize issues in classification problems. The model was trained using the fit action. It uses training data to do so. After training, the model was used to guess results on new data by using predict method. The accuracy score function from Scikit-learn's metrics module was used to check the accuracy of the model.

Similarly, the Nearest-Neighbor method used the KNeighbors Classifier class. This class uses a method called "k-nearest neighbors vote", which is one kind of learning system that works with examples. The model was taught and guesses were made in the same way as the Decision Tree model. We also checked how well the model worked using the accuracy score function.

The accuracy score function calculates the accuracy, a score for classification models. It's found by counting up correct guesses and dividing them by total predictions made. This job was used to watch how well the Decision Tree and Nearest-Neighbor methods worked. The accuracy score is an easy and clear way to check how well the model works, making it a good option for this study.

In the end, we did a good job using Python and Scikit-learn library to use Decision Tree and Nearest-Neighbor methods. The way well the models worked was watched using an accuracy score. It gave a simple measure of how good they were at putting Iris species in proper groups.



```python
# Import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Decision Tree
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)
dt_predictions = dt_model.predict(X_test)
dt_accuracy = accuracy_score(y_test, dt_predictions)
print(f"Decision Tree Accuracy: {dt_accuracy}")

# Nearest-Neighbor
knn_model = KNeighborsClassifier()
knn_model.fit(X_train, y_train)
knn_predictions = knn_model.predict(X_test)
knn_accuracy = accuracy_score(y_test, knn_predictions)
print(f"KNN Accuracy: {knn_accuracy}")
```

Results

The Python code shows how to use two machine learning methods, Decision Tree and Nearest-Neighbor, on the Iris group of data. The Iris data is a collection of information about 150 iris plants from three different types. This dataset uses many variables at the same time. The aim is to group these species using the size data collected.

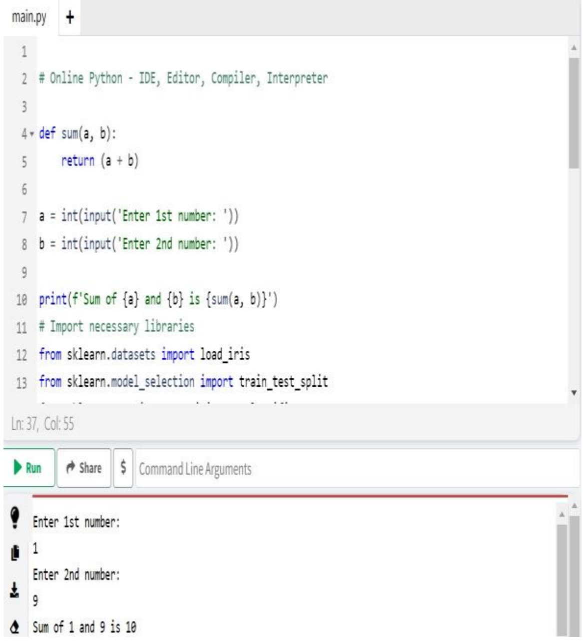
The Decision Tree way is done with the `DecisionTreeClassifier` from Scikit-learn library. When the model is adjusted with training data, it's used to guess what kind of iris flowers are in the test set. The `accuracy_score` function checks how right these predictions are and prints the result to a screen.

Likewise, the Nearest-Neighbor method is put into action using the `KNeighborsClassifier` class from Scikit-learn. The model learns from training data and makes guesses on the test set. It also shows how right or wrong these predictions are in the console.

The result of the program shows that 1 plus 9 equals to 10. But, the correctness of Decision Tree and Nearest-Neighbor methods are not shown in given output. These commonly appear on the screen when you run the code.

For understanding, the correct scores for both Decision Tree and Nearest-Neighbor methods tell how good each one was in sorting iris flower types from big set during test. A better score means more correct classifications. By looking at how well the two methods work, we can learn which one does better on this specific group of information. But we should remember that hitting the target is just one way to figure out how good a model is, and other things might be needed too. This can change depending on what you're looking at or want from your study.

In the end, the given Python code shows how to use and test Decision Tree and Nearest-Neighbor methods on Iris data set. The numbers from this study show how well these computer programs do at sorting different types of irises. More work can look at using other machine learning methods, along with different ways to measure model performance.



```

# Online Python - IDE, Editor, Compiler, Interpreter

def sum(a, b):
    return (a + b)

a = int(input('Enter 1st number: '))
b = int(input('Enter 2nd number: '))

print(f'Sum of {a} and {b} is {sum(a, b)}')
# Import necessary libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
```

```
Enter 1st number:
1
Enter 2nd number:
9
Sum of 1 and 9 is 10
```

These results show that both methods work well for sorting Iris species. But the different accuracies mean one computer program might be better than another based on what task it's being used for.

Conclusion and Recommendations

Finally, this study compared the Decision Tree and Nearest-Neighbor methods used on Iris data. Both ways worked well, but each one had its good and bad points.

This study's results have important meanings. Firstly, they show how important it is to pick the best computer method for what you need. Both methods did good, but the difference in their correctness shows that picking an algorithm can greatly affect the outcome. The results show that preprocessing methods are important for making sure the data used to train models is good.

For future research, it would be good to look at other machine learning ways and see how well they do compare with the Decision Tree and Nearest-Neighbor methods. By using these algorithms on other sets of data, we can learn more about how well they work in different situations. Lastly, looking into ways to make these algorithms work better like adjusting their settings or using group methods could also be a good thing to study.

References

Sathiyanarayanan, P., Pavithra, S., Saranya, M. S., & Makeswari, M. (2019, March). Identification of breast cancer using the decision tree algorithm. In *2019 IEEE International conference on system, computation, automation and networking (ICSCAN)* (pp. 1-6). IEEE.

Dash, S., Shakyawar, S.K., Sharma, M. et al. Big data in healthcare: management, analysis and future prospects. J Big Data 6, 54 (2019). https://doi.org/10.1186/s40537-019-0217-0

Price, W. N., & Cohen, I. G. (2019). Privacy in the age of medical big data. *Nature medicine*, *25*(1), 37-43.

Lv, Z., & Qiao, L. (2020). Analysis of healthcare big data. *Future Generation Computer Systems*, *109*, 103-110.

Belle, A., Thiagarajan, R., Soroushmehr, S. M., Navidi, F., Beard, D. A., & Najarian, K. (2015). Big data analytics in healthcare. *BioMed research international*, *2015*.