ISSN 2959-6157

Analysis of the Apple Quality Dataset

Yansong Zhu

Abstract:

This article introduces a dataset containing apple features, with 4000 rows, including apple identifiers, size, weight, sweetness, crispness, juiciness, ripeness, acidity, and other characteristics. The data set can support classification and regression tasks, where quality features can be used as classification targets or converted to numerical values for regression. In addition, the data set's features are evenly distributed, which is beneficial to model training. The experiment used two algorithms, decision tree, and random forest, for classification tasks. The results showed that the accuracy of the random forest reached 90.625%, which was better than the 80.625% of the decision tree. This confirms the effectiveness of the dataset in classification tasks and the superior performance of the random forest model.

Keywords: Data analysis, Machine learning, Statistics

1. Introduction

With people's increasing pursuit of food quality, scientific evaluation of the quality of fruits and other agricultural products has become particularly important. As one of people's favorite fruits, apple's quality evaluation has attracted widespread attention. This paper introduces a dataset containing Apple features to support the automatic assessment of Apple quality. The dataset contains 4,000 samples, each containing apple identifiers: size, weight, sweetness, crispness, juiciness, ripeness, and acidity, among other characteristics. These characteristics comprehensively reflect the quality of the apple from both physical and sensory aspects. The dataset is suitable for classification and regression tasks, where quality features can be used as classification targets or converted to numerical values for regression analysis.

2. Dataset overview

This dataset contains the following characteristics:

1. 'A_id': The identifier of the apple, a unique number used to distinguish different apples.
2. 'Size': The size of the apple, which is a numeric feature that indicates the size of the apple.
3. 'Weight': The weight of an apple, a numerical feature that indicates the weight of an apple.
4. 'Sweetness': An apple's sweetness, a numerical characteristic that indicates how sweet an apple is.
5. 'Crunchiness': The 'crunchiness' of the apple is a numerical characteristic that indicates how crisp the apple is.
6. 'Juiciness': The juiciness of an apple, which is a numerical characteristic that indicates how juicy an apple is.
7. 'Ripeness': The ripeness of an apple, which is a numerical characteristic that indicates how ripe an apple is.
8. 'Acidity': The acidity of an apple, which is a numerical characteristic that indicates how acidic an apple is.
9. 'Quality': The quality of the apple is a definite characteristic that indicates whether the quality is good or bad.

These characteristics can be used to describe the physical and organoleptic properties of apples and their quality.

The size of the dataset is 4000×9, that is, there are 4000 rows and 9 columns, and the statistical description of the dataset is as follows:

	A_id	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness
count	4000	4000	4000	4000	4000	4000	4000
mean	1999.50000	-0.5030	-0.9895	-0.470479	0.985478	0.51211	0.49827
Std	1154.84486	1.9280	1.6025	1.943441	1.402757	1.93028	1.87442
min	0.000000	-7.1517	-7.1498	-6.894485	-6.055058	-5.9618	-5.8645
25%	999.750000	-1.8167	-2.0117	-1.738425	0.062764	-0.8012	-0.77167
50%	1999.50000	-0.5137	-0.9847	-0.504758	0.998249	0.53421	0.50344
75%	2999.25000	0.80552	0.03097	0.801922	1.894234	1.83597	1.76621

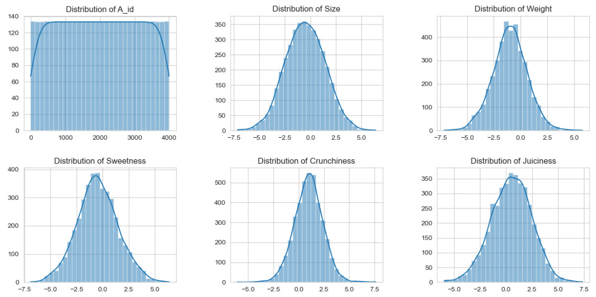
max	3999.00000	6.406367	5.790714	6.374916	7.619852	7.364403	7.237837

Here is a statistical description of the dataset:

- Count: Indicates the number of non-missing values for each feature, with 4000 non-missing values for each feature, except for one missing value for each of the A_id, Size, Weight, Sweetness, Crunchiness, Juiciness, Ripeness, and Quality features.
- Mean: Represents the average value of each feature.
- Std: Represents the standard deviation of each feature.
- Min: Represents the minimum value for each feature.
- 25%, 50%, 75%: represent the first quartile (Q1), median (Q2), and third quartile (Q3), respectively, for each trait.
- Max: Represents the maximum value for each feature.

We can see the range and distribution of values for each feature. For example, the Size feature can range from -7.15 to 6.41, with a mean value of -0.50 and a standard deviation of 1.93. Again, we can make similar observations about other features. These statistics can help us better understand the characteristics and distribution of data and provide a basis for subsequent data preprocessing and modeling.

We plotted their histograms and kernel density estimation (KDE) curves. This helps us understand the distribution shape, central tendency, and degree of diffusion of each feature, and the results are as follows:



It can be seen that the distribution of the above six features is unimodal and concentrated near the mean value, which indicates that the data distribution is very uniform, which is helpful for the training of the model.

3. Classification and regression task

This dataset can be used for classification tasks and regression tasks. Here's how to use it for both tasks:

Classification task: The Quality feature of the dataset is a categorical variable with two possible categories: good and bad. Therefore, this feature can be used as the target variable and the rest as input features to train a classification model. The classification task aims to predict whether a new apple sample will be good or bad.

Regression task: Although the Quality feature is definite, if we are interested in numerical prediction, we can convert the Quality feature into numeric form, such as converting good to 1 and bad to 0, and then use this value as the target variable to train a regression model. The regression task aims to predict the quality score of a new apple sample.

Except the Quality feature, the other features in the dataset are numeric and can, therefore, be used directly as input features for regression tasks. For example, we can try to predict the characteristics of an apple, such as size, weight, sweetness, crunchiness, juvenileness, ripeness, or acidity.

To test the performance of the dataset on the classification task, I used two methods commonly used in machine learning, Decision Tree and Random Forest, to complete the classification task. Decision trees are a type of classification and regression algorithm that is widely used in machine learning. It works by dividing the data through a series of tests, each based on a feature in the dataset. These tests are organized into a tree structure, where each inner node represents a test of a feature, each branch represents the test result, and each leaf node represents a final classification or regression value. The learning process of the decision tree is to select the best features from the training data and divide them until the stopping conditions are met, such as information gain, Gini impurity, or entropy reduction. Decision trees are easy to understand and interpret, handle numerical and categorical features, and do not require feature scaling and preprocessing. Still, they are prone to overfitting, sensitive to noise in the training data, and unstable because small data changes can lead to significant changes in the tree structure.

A random forest is an ensemble learning algorithm that consists of multiple decision trees. In a random forest, each tree is built based on randomly selected samples and features from the original dataset. This randomness makes each tree slightly different, improving the model's generalization ability. In the classification task, the random forest determines the final result by voting for all trees. The advantages of random forests are that they generally have better performance and generalization capabilities than individual decision trees, can be used for classification and regression tasks, can provide feature importance, and are more robust to noise and outliers. However, random forests take longer to train than a single decision tree, and the results are not easy to interpret because multiple trees are involved in making predictions and, in some cases, may not perform as well as other algorithms, such as gradient boosting trees. Random forest is a powerful machine-learning algorithm that reduces the risk of overfitting and improves the model's generalization ability by combining the advantages of multiple decision trees.

```
From sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

dataset.dropna(inplace=True)

X = dataset.drop(columns=['Quality', 'A_id'])
y = dataset['Quality']

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

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

models = {
   'Decision Tree': DecisionTreeClassifier(),
   'Random Forest': RandomForestClassifier()
}

results = {}

For name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   accuracy = accuracy_score(y_test, y_pred)
   results[name] = accuracy

print(results)
```

Finally, the accuracy of the two methods was tested, and the following results were obtained:

```
{'Decision Tree': 0.80625, 'Random Forest': 0.90625}
```

Finally, the accuracy of the two methods was tested, and the following results were obtained:
{'Decision Tree': 0.80625, 'Random Forest': 0.90625}

We can see that the random forest model performs better on this dataset than the single decision tree model. Random forests achieve an accuracy rate of 90.625%, while decision trees have an accuracy rate of 80.625%. This suggests that the random forest learns better about the relationship between features and target variables on this dataset and can more accurately predict whether the quality of apples is "good" or "bad." These results confirm that the dataset can be successfully applied to classification tasks. Due to the high accuracy of the random forest model, we can assume that the features in the dataset contain enough information to distinguish between apples of different qualities. In addition, the performance of the random forest model also shows that by integrating the learning ability of multiple decision trees, we can build a powerful classification model that has a good fit for the training data and a good generalization ability for unseen data.

4. Conclusion

This report introduces an apple feature data set, which contains apple identifiers, size, weight, sweetness, crispness, juiciness, ripeness, and acidity, with 4000 samples. The dataset is suitable for classification and regression tasks, where quality features can be used as classification targets or converted to numerical values for regression analysis. Through statistical analysis of the data set, we observed that the distribution of each feature is relatively uniform, which is conducive to model training. In the classification task, we used two algorithms, a decision tree, and a random forest, for prediction. The results showed that the accuracy of the random forest reached 90.625%, which was higher than the 80.625% of the decision tree. This confirms the effectiveness of the dataset in classification tasks and highlights the advantages of the random forest model in feature learning. Random forest reduces the risk of overfitting and improves the model's generalization ability by integrating multiple decision trees, thereby obtaining more accurate classification results.

This paper introduces a clearly structured and evenly distributed apple feature data set and verifies its effectiveness in classification tasks through classification experiments. The random forest model performed well on this dataset, providing a powerful classification tool expected to distinguish apples of different qualities in real-world applications. This research result lays the foundation for further data analysis and model optimization.

Reference

Cramer, G. M., Ford, R. A., & Hall, R. L. (1976). Estimation of toxic hazard—a decision tree approach. *Food and cosmetics toxicology*, *16*(3), 255-276.

Greenhalgh, T. (1997). How to read a paper: Statistics for the non-statistician. II: "Significant" relations and their pitfalls. *BMJ*, *315*(7105), 422-425.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.