Classifying Edibility of Mushrooms Using Machine Learning Algorithms

The aim is to develop machine learning models—using decision trees, random forest, and nearest neighbors—that can classify mushrooms as either edible or poisonous based on their features from the mushroom dataset.

2. Data Description:

The training data consisted of 50000 observations with 20 features, including 3 numerical and 17 nominal categorical features. Due to the limited size of the initial test set (7 observations), 15000 observations from the training dataset were reallocated to form an expanded test set. This resulted in a final training dataset of 30000 observations and 16 features, and a test dataset of 15007 observations.

The model was evaluated on the test set using the accuracy formula, F1-score, and accuracy metrics provided by GridSearchCV, ensuring performance consistency across all data subsets.

2.1 Preprocessing Stage:

1. Handling missing values:
    - The data consisted of null and non-null values. To handle missing values, all unknown (“?”) values were converted into NaN.
    - Features that had less than 20% of the total observations were removed
    - Mode imputations were used the other features with missing values

2. One-Hot encoding :
    - Categorical features were one-hot encoded
    - Edge case: all features were converted to float data type in case it was bool
3. Normalization: KNN specific
    - I addressed skewness of the numerical features before normalizing to make sure the outliers were taken into consideration.
        - All 3 features were right-skewed on various degrees.
        - For moderate skewness, square root transformation was applied to the values
        - For high skewness, log transformation was applied to the values
    - Numerical features were normalized using min-max scaling, MinMaxScaler() from sklearn, into a range between [0,1] because KNN is a distance-based algorithm.

3. Model Selection

The models were built using
- Nearest Neighbor, KNeighborsClassifier: U
- Decision Tree, DecisionTreeClassifier:
- Random Forest, RandomForestClassifier

All are methods in the Sci-Kit Learn Library.

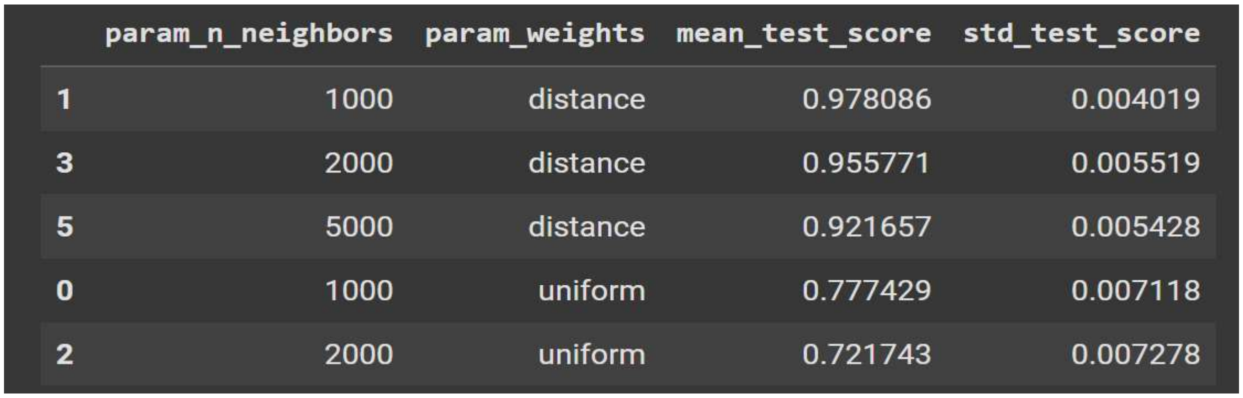
4. Hyperparameter Tuning (Grid Search)
The hyperparameter tuning was performed to optimize the performance of the models and ensure robust predictive accuracy. GridSearchCV from Sci-Kit Learn Library was used to tune the hyperparameters of choice.

The following parameters below were tuned for:
KNN model; the metric was fixed to euclidean
- K: [1000, 2000, 5000]
- Weights: uniform or distance

Decision Tree model; random state was fixed to 42 to allow for reproducibility
- Splitter: [best, random]
- Criterion: [gini, entropy]
- Min_samples_split: [5, 10]
- Max_depth: [5, 10]

Random Forest model; random state was fixed to 42 to allow for reproducibility
- Criterion: [gini, entropy]
- Min_samples_split: [5, 10]
- Max_depth: [5, 10]
- Ccp_alpha: [0.0, 0.2]

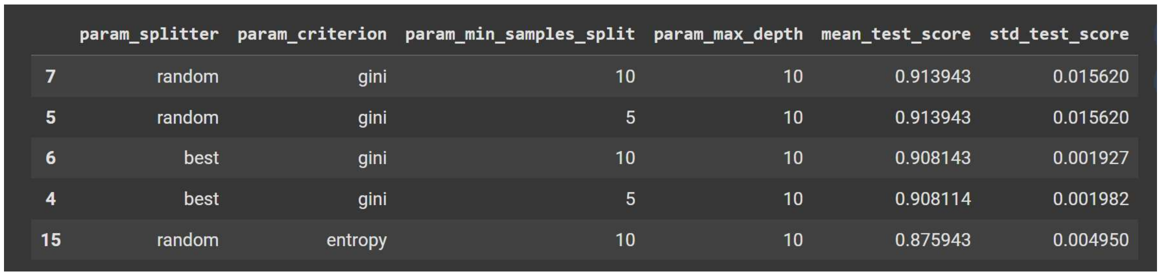
5. Cross-Validation Results
The scoring method used was accuracy metric and 5-fold cross validation was performed for each model to obtain the best estimator and ensure model's performance was consistent across all data subsets. The top 5 mean_test_scores for the parameters are provided in the cross-validation section.

KNN
Top 5 Mean 5-Fold Cross-Validation Scores:


	param_n_neighbors	param_weights	mean_test_score	std_test_score
1	1000	distance	0.978086	0.004019
3	2000	distance	0.955771	0.005519
5	5000	distance	0.921657	0.005428
0	1000	uniform	0.777429	0.007118
2	2000	uniform	0.721743	0.007278

Decision Tree

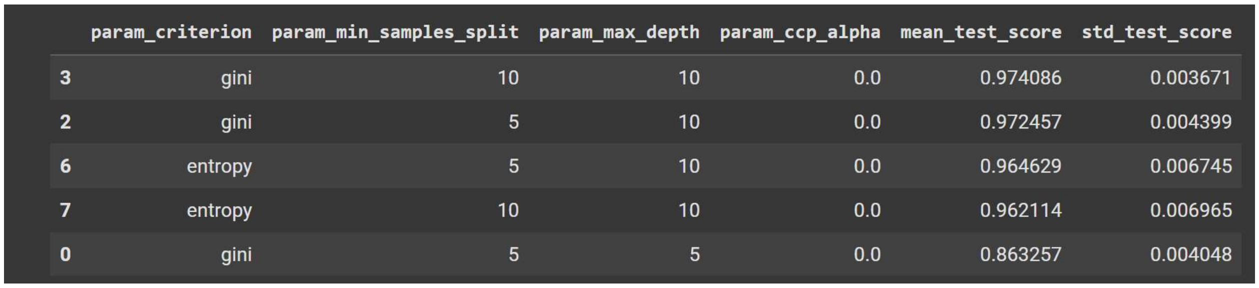
Top 5 Mean 5-Fold Cross-Validation Scores:



	param_splitter	param_criterion	param_min_samples_split	param_max_depth	mean_test_score	std_test_score
7	random	gini	10	10	0.913943	0.015620
5	random	gini	5	10	0.913943	0.015620
6	best	gini	10	10	0.908143	0.001927
4	best	gini	5	10	0.908114	0.001982
15	random	entropy	10	10	0.875943	0.004950

Random Forest

Top 5 Mean 5-Fold Cross-Validation Scores:



	param_criterion	param_min_samples_split	param_max_depth	param_ccp_alpha	mean_test_score	std_test_score
3	gini	10	10	0.0	0.974086	0.003671
2	gini	5	10	0.0	0.972457	0.004399
6	entropy	5	10	0.0	0.964629	0.006745
7	entropy	10	10	0.0	0.962114	0.006965
0	gini	5	5	0.0	0.863257	0.004048

6. Final Model Performances

KNN Model Performance:

```
Best parameters:  {'n_neighbors': 1000, 'weights': 'distance'}
Best score:  0.9781
```

Accuracy Method Given:

```
correct: 14753 , wrong: 254
Final Accuracy is  0.9831
```

Mean & Std of Errors:

```
Mean of Errors: 0.01693
Standard Deviation of Errors: 0.1290
```

Accuracy Metric of GridSearchCV:

```
test accuracy: 0.9831
```

F-1 Score Metric:

```
             precision    recall  f1-score   support
           e       0.99      0.97      0.98      6597
           p       0.98      0.99      0.99      8410
    accuracy                           0.98     15007
   macro avg       0.98      0.98      0.98     15007
weighted avg       0.98      0.98      0.98     15007
```

Decision Tree Model Performance

```
Best parameter:  {'criterion': 'gini', 'max_depth': 10,
                  'min_samples_split': 5, 'splitter': 'random'}
Best score:  0.9139
```

Accuracy Method Given:

```
correct: 13739 , wrong: 1268
Final Accuracy is  0.9155
```

Mean & Std of Errors:

```
Mean of Errors: 0.0845
Standard Deviation of Errors: 0.2781
```

Accuracy Metric of GridSearchCV:

```
test accuracy: 0.9155
```

F-1 Score Metric:

```
              precision    recall  f1-score   support
           e       0.89      0.92      0.91      6597
           p       0.93      0.91      0.92      8410
    accuracy                           0.92     15007
   macro avg       0.91      0.92      0.91     15007
weighted avg       0.92      0.92      0.92     15007
```

Random Forest Model Performance

```
Best parameter:  {'ccp_alpha': 0.0, 'criterion': 'gini',
                  'max_depth': 10, 'min_samples_split': 10}
Best score:  0.9741
```

Accuracy Method Given:

```
correct: 14664 , wrong: 343
Final Accuracy is  0.9771
```

Mean & Std of Errors:

```
Mean of Errors: 0.02286
Standard Deviation of Errors: 0.1494
```

Accuracy Metric of GridSearchCV:

```
test accuracy: 0.9771
```

F-1 Score Metric:

```
              precision    recall  f1-score   support
           e       0.98      0.97      0.97      6597
           p       0.98      0.98      0.98      8410
    accuracy                           0.98     15007
   macro avg       0.98      0.98      0.98     15007
weighted avg       0.98      0.98      0.98     15007
```

7. Conclusion

Among the three models, the KNN model achieved the highest accuracy (98.31%) and F1-score (macro avg: 0.98), making it the best-performing model for predicting mushroom edibility. Random Forest also performed well with slightly lower accuracy (97.71%), while the Decision Tree model had the lowest accuracy (91.55%), indicating it may not generalize as effectively as the others.