Mushroom Dataset Classifier

Kawya A.H.D. (E/20/197)
Department of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e20197@eng.pdn.ac.lk

***Abstract*—This study evaluates multiple machine learning models, including Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), and Random Forest, for binary classification on the Mushroom dataset. Feature engineering techniques such as Chi-Squared feature selection and Multiple Correspondence Analysis (MCA) were applied to enhance model performance. Experimental results demonstrate that while Logistic Regression and Naïve Bayes provided solid baseline performance, SVM with hyperparameter tuning and Random Forest significantly outperformed them. Random Forest achieved perfect accuracy, though with potential risk of overfitting, whereas tuned SVM provided robust and generalizable performance. The findings highlight the importance of feature engineering and model tuning in developing reliable predictive models.**

***Index Terms*—Machine Learning, Logistic Regression, Naïve Bayes, Support Vector Machine, Random Forest, Feature Selection, Chi-Squared Test, Multiple Correspondence Analysis (MCA), Model Evaluation**

I. Introduction and Dataset Description

The dataset selected for this analysis is the Mushroom dataset, obtained from the UCI Machine Learning Repository. It belongs to the biological domain and is commonly used for binary classification tasks.

This data contains 8124 records with 22 features such as cap-shape, cap-surface, cap-color and odor. The target variable is *poisonous*, which indicates whether the mushroom is poisonous (p) or edible (e).

The main objectives of this work are to:

- Explore and visualize the dataset to understand its structure.
- Preprocess the data, handling the missing values.
- Train multiple classification models, including Logistic Regression, SVM, Naïve Bayes and Random Forest.
- Compare the models based on accuracy, precision, recall, F1-score and confusion matrix to identify the best-performing model.

This dataset was chosen because it is clean, easy to interpret, and enables the exploration of machine learning methods such as handling categorical data, feature selection, and model evaluation.

II. Data Preprocessing

To ensure the data quality and prepare the data for modelling, several preprocessing steps were performed.

A. Data Cleaning

The dataset was checked for missing values. In the stalk-root column, 2480 records were missing, which is about 30% of the total number of instances. Since the feature is categorical, all the missing values were replaced with the most frequent value. This preserved distributional characteristics while avoiding unnecessary loss of data.

B. Data Transformation

Categorical variables were transformed into numerical representations to enable feature selection.

- **Binary Feature Transformation** – The binary feature "veil-type" was mapped manually, where 'p' was encoded as 1 and 'u' as 0.
- **Label Encoding of Categorical Features** – All other categorical features with multiple categories were encoded using Label Encoding.

These transformations were performed only for the application of the Chi-Squared feature selection method, since it requires input data to be in discrete numerical form.

III. Exploratory Data Analysis (EDA)

A. Class Distribution

The dataset has balanced classes, so oversampling or undersampling was not required. Since poisonous mushrooms represent the more safety-critical class, per-class recall was monitored to minimize false negatives.

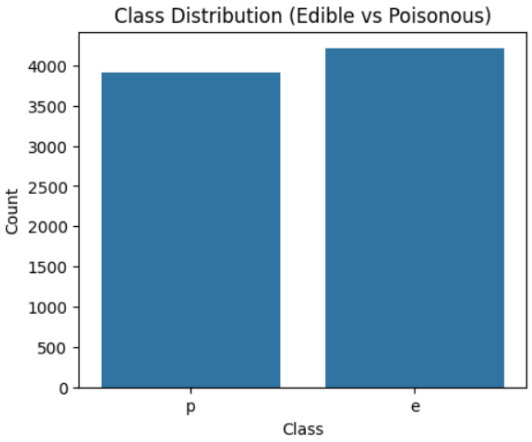

Fig. 1. Class Distribution of Mushrooms

B. Univariate Feature Distributions

Count plots revealed features with dominant categories and others with uniform distributions. Columns such as veil-type carried little to no variability and were excluded from further analysis.

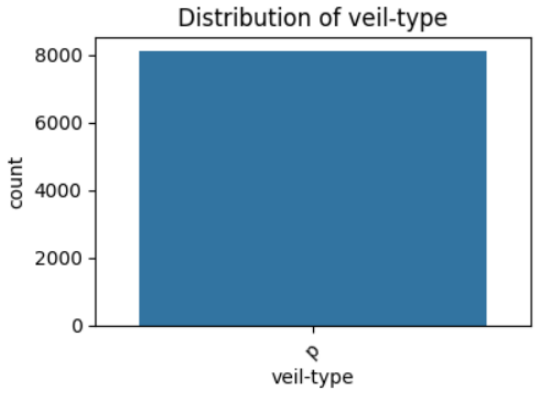

Fig. 2. Distribution of Veli-type

C. Correlation Analysis

A correlation heatmap was produced after encoding categorical variables. While Pearson correlation on encoded data does not perfectly capture categorical relationships, more robust methods such as Chi-Squared tests and MCA were applied later for feature evaluation and dimensionality reduction.

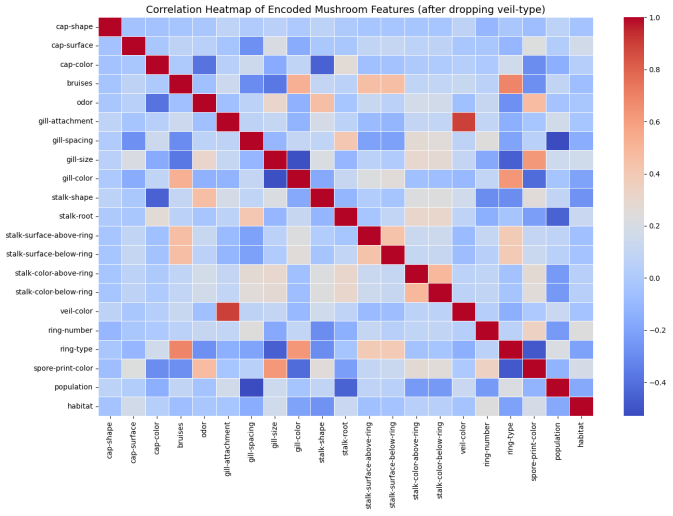

Fig. 3. Correlation Matrix

IV. Feature Engineering

A. Feature Selection – Chi-Squared Test

The Chi-squared (χ^2) statistical test was applied to measure each feature's association with the target label. The top 10 features by χ^2 score were selected for the next step.

The Chi-squared test is appropriate for evaluating independence between categorical features and a categorical target. It

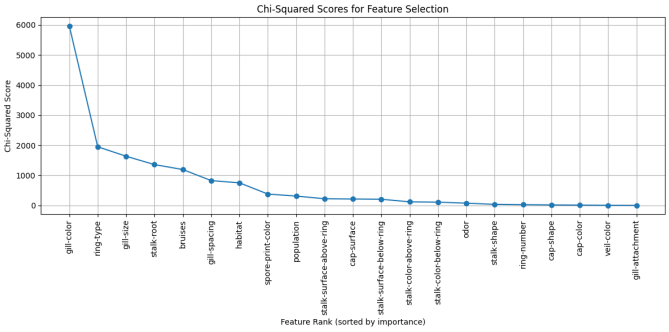

Fig. 4. Chi-Squared Score Plot

is computationally inexpensive and provides a straightforward ranking of features by strength of association.

B. Feature Extraction – Multiple Correspondence Analysis (MCA)

MCA was performed on the set of top features returned by χ^2 to transform the selected categorical attributes into a small number of continuous components. The MCA eigenvalues were inspected, and a number of components (four in this workflow) were chosen using the cumulative explained inertia. The resulting MCA components were used as inputs to downstream classifiers.

Multiple Correspondence Analysis (MCA) reduces the categorical features by creating a matrix with values of zero or one. It captures associations between levels across multiple categorical variables, not just per-feature variance. And because of that, this transformation was applied only on training data to avoid data leakage.

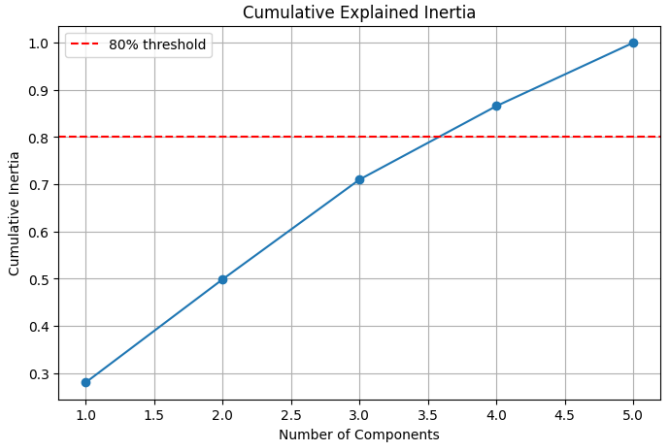

Fig. 5. Cumulative Explained Inertia

The extracted components from MCA are continuous, low-dimensional, and appropriate as inputs for models that expect numeric features.

For this case I have not used PCA, because it is designed for continuous, quantitatively measured variables. Applying PCA directly to arbitrary integer encodings of categorical variables

is not theoretically correct, because PCA would treat those numeric codes as continuous quantities, potentially producing misleading directions.

V. Modeling

Three supervised classification algorithms were chosen for evaluation:

- **Logistic Regression** – A linear classifier that is interpretable and effective for categorical data.
- **Support Vector Machine (SVM, RBF Kernel)** – A powerful non-linear classifier that separates edible vs. poisonous mushrooms effectively.
- **Naïve Bayes (GaussianNB)** – A probabilistic model that assumes conditional independence, useful as a strong baseline.

VI. Evaluation

SVM with RBF kernel outperformed Logistic Regression and Naïve Bayes, reaching 96% accuracy. Hyperparameter tuning using GridSearchCV further improved SVM performance to 99%.

Random Forest, an ensemble method, achieved 100% accuracy with perfect precision, recall, and F1-score. However, such perfect performance raises concerns of overfitting if unseen data contains more variability.

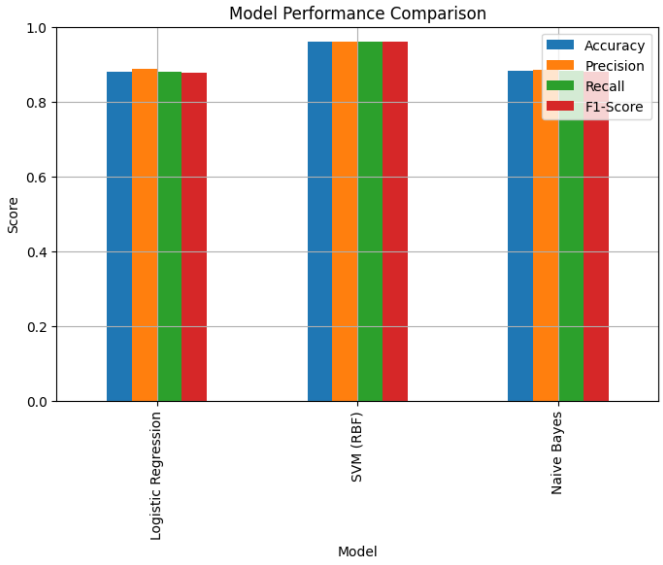

Fig. 6. Model Performance Comparison

VII. Model Improvements

A. Hyperparameter Tuning of the SVM Model

After identifying SVM with RBF kernel as the best-performing baseline model, further improvement was carried out using hyperparameter tuning. A GridSearchCV approach with 5-fold cross-validation was employed to systematically search for the optimal set of parameters.

The following hyperparameters were tuned:

- C (Regularization Parameter): Controls the trade-off between a smooth decision boundary and correct classification of training examples. A smaller C encourages a wider margin but allows more misclassifications, while a larger C aims to classify all training points correctly, risking overfitting.
- Gamma (Kernel Coefficient): Defines how far the influence of a single training example reaches. A low gamma means smoother decision boundaries, while a high gamma leads to more complex boundaries, risk of overfitting.
- Kernel: Restricted to RBF (Radial Basis Function), as it showed the strongest performance during baseline evaluation.

The best parameters identified were: C = 100, gamma = scale, kernel = 'rbf' The best cross-validation score achieved was 0.9852274530704094 which indicates strong generalization ability.

B. Evaluation of Tuned SVM

The tuned SVM model was then evaluated on the test set. Results indicated a clear improvement compared to the baseline SVM.

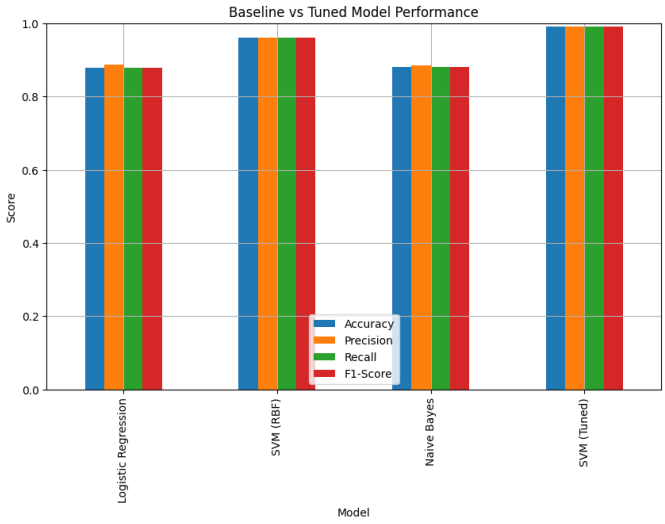

Fig. 7. Model Performance Comparison

A comparison plot between the baseline and tuned models further illustrates the improvement across all metrics.

VIII. Ensemble Learning

As part of the model evaluation process, an ensemble learning approach was explored using the Random Forest Classifier. Random Forest is a popular and powerful machine learning algorithm that combines the predictions of multiple decision trees to improve accuracy and control overfitting.

The Random Forest model achieved an accuracy of 1.0 on the test dataset. There is a balanced performance across precision, recall, and F1-score, showing the model's ability to handle class distributions effectively.

The heatmap highlighted that all the predictions were correctly classified.

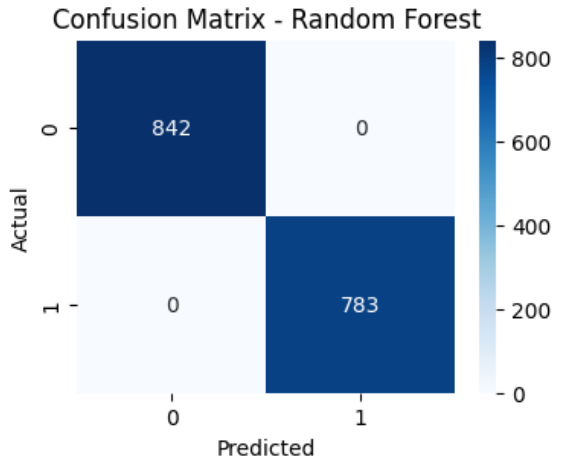

Fig. 8. Confusion Matrix

IX. Discussion and Conclusion

Logistic Regression achieved an overall accuracy of 88% with reasonably balanced precision and recall. While effective as a baseline linear model, it struggled with more complex, nonlinear decision boundaries in the data.

Naïve Bayes also achieved an accuracy of 88%, with strong recall for class 0 but lower recall for class 1. This reflects the algorithm's simplifying assumption of feature independence, which may not be held in this dataset.

Support Vector Machine (SVM, RBF kernel) outperformed the linear models, reaching 96% accuracy with well-balanced precision and recall across both classes. This highlights the effectiveness of nonlinear kernels in capturing complex feature interactions.

Tuned SVM (C=100, kernel=RBF, gamma=scale) further improved performance to 99% accuracy, demonstrating the importance of hyperparameter optimization in SVMs. This model showed near-perfect classification with minimal misclassifications.

Random Forest (Ensemble Learning) achieved 100% accuracy, precision, recall, and F1-score. This suggests that the ensemble method effectively captured all underlying patterns in the dataset. However, such perfect performance raises the possibility of overfitting, particularly if the test set does not fully represent unseen real-world scenarios.

A. Key Insights

- Ensemble methods (Random Forest) provided the highest accuracy.
- SVM with RBF kernel (tuned) offered the best trade-off between accuracy and generalization.
- Logistic Regression and Naïve Bayes, while effective as baselines, underperformed compared to non-linear models.

References

[1] S. Shariati, "Tune reduction techniques, PCA and MCA, to build a model on a mixed data," Medium, Oct. 20, 2020. [Online]. Available: https://sinashariati.medium.com/tune-the-feature-reductions-pca-and-mca-to-build-a-model-on-a-categorical-and-numerical-data-7c27310607b8

[2] GeeksforGeeks, "Support Vector Machine (SVM) Algorithm," Jan. 20, 2021. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/