Support Vector machine

**Dataset Selection:**

For this assignment, we'll utilize the widely recognized Mushroom Dataset

**Task 1: Exploratory Data Analysis (EDA)**

1. Load the Mushroom dataset and perform fundamental data exploration.
2. Utilize histograms, box plots, or density plots to understand feature distributions.
3. Investigate feature correlations to discern relationships within the data.

**Task 2: Data Preprocessing**

1. Encode categorical variables if necessary.
2. Split the dataset into training and testing sets.

**Task 3: Data Visualization**

1. Employ scatter plots, pair plots, or relevant visualizations to comprehend feature distributions and relationships.
2. Visualize class distributions to gauge dataset balance or imbalance.

**Task 4: SVM Implementation**

1. Implement a basic SVM classifier using Python libraries like scikit-learn.
2. Train the SVM model on the training data.
3. Evaluate model performance on the testing data using appropriate metrics (e.g., accuracy, precision, recall, F1-score).

**Task 5: Visualization of SVM Results**

1. Visualize classification results on the testing data.

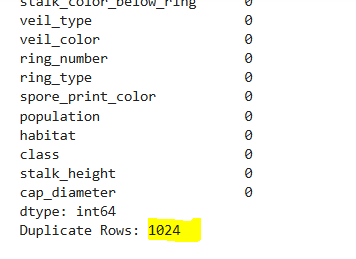
**Task 7: Comparison and Analysis**

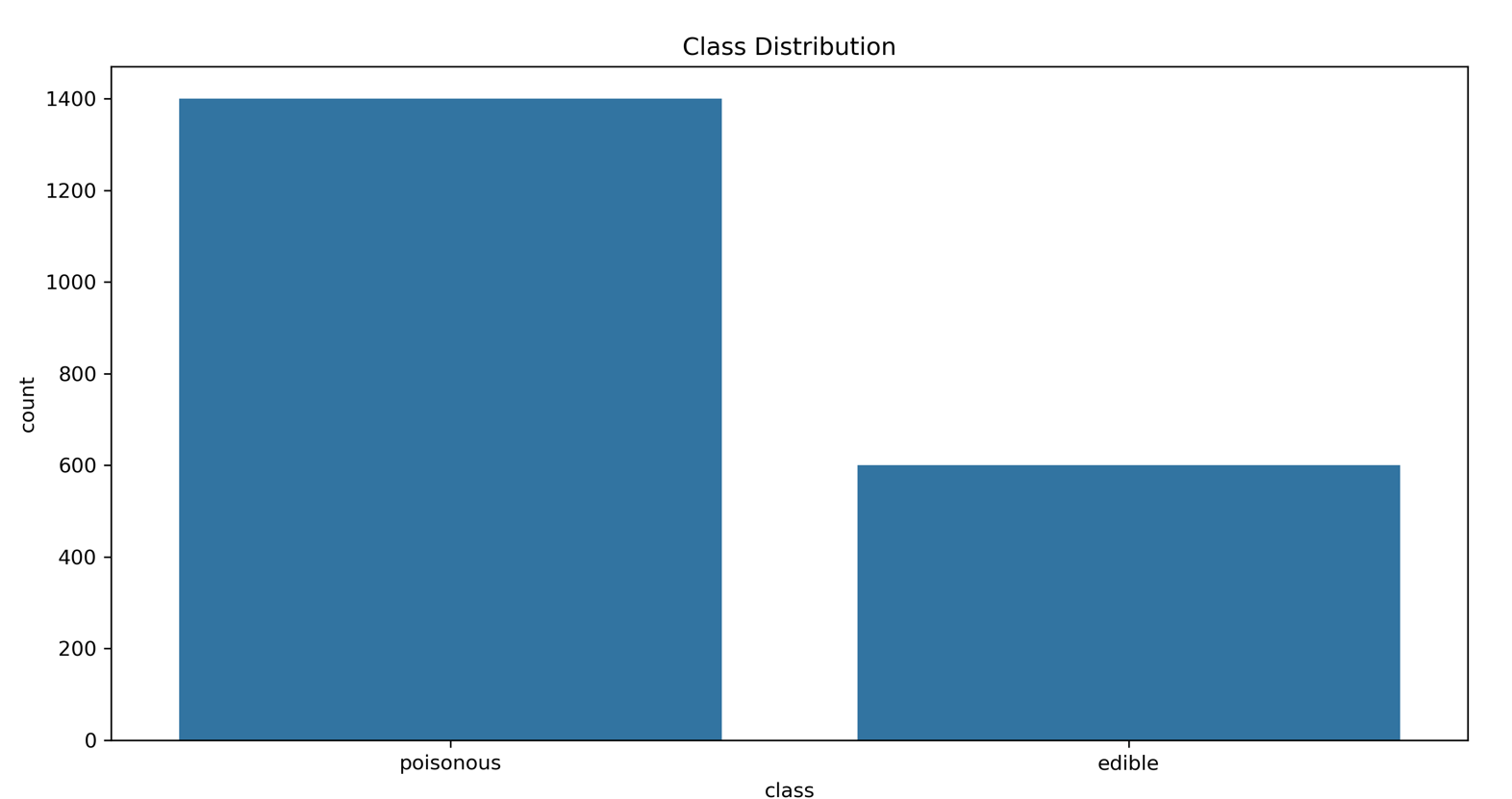
1. Compare SVM performance with various kernels (e.g., linear, polynomial, radial basis function).
2. Analyze SVM strengths and weaknesses for the Mushroom dataset based on EDA and visualization results.
3. Discuss practical implications of SVM in real-world classification tasks.

Observations:

**Task 1: Exploratory Data Analysis (EDA)**

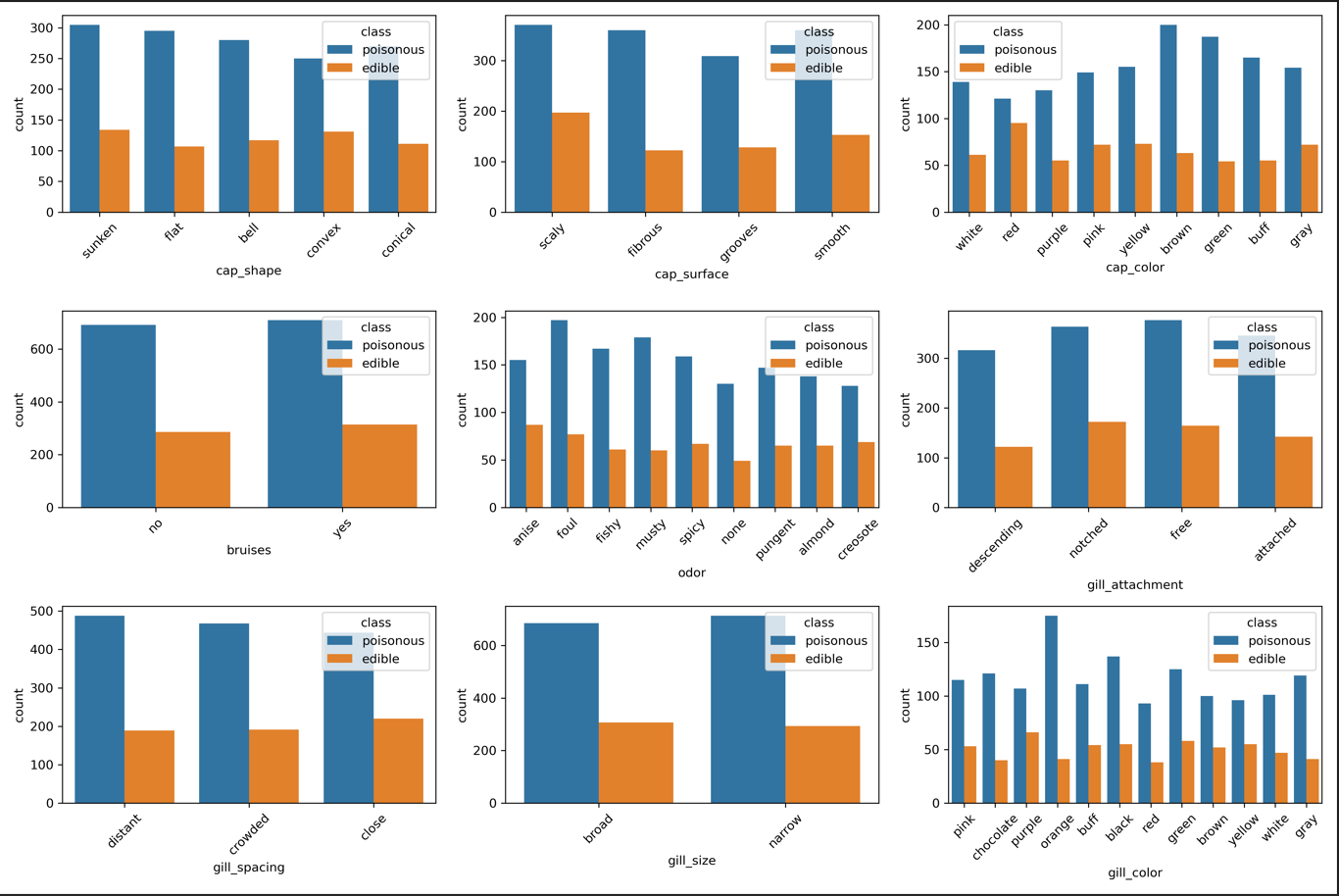
1. Load the Mushroom dataset and perform fundamental data exploration.
2. Utilize histograms, box plots, or density plots to understand feature distributions.
3. Investigate feature correlations to discern relationships within the data.



**Class Distribution:**

* There is a significant imbalance in the dataset, with **poisonous mushrooms outnumbering edible ones** by a large margin.
* This imbalance may affect model performance, and techniques like **oversampling (SMOTE) or under sampling** may be required to improve classification.



**Feature Distribution Across Classes:**

Each subplot shows the distribution of different categorical features for **poisonous (blue) vs. edible (orange) mushrooms**.

**Key Observations:**

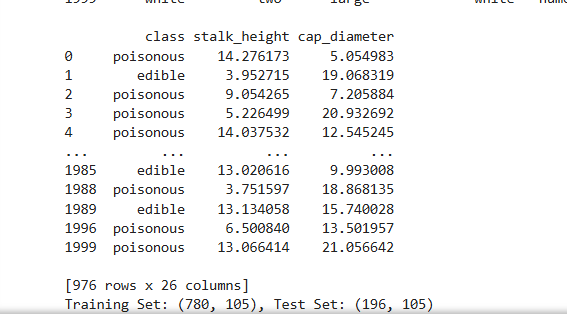
* **Odor Feature:**
  + Odor appears to be a strong distinguishing feature.
  + Certain odors (e.g., "foul", "fishy", "musty") are highly associated with poisonous mushrooms.
  + Other odors like "almond" and "anise" are more common in edible mushrooms.
  + **Insight:** This feature may be a critical predictor of edibility.
* **Bruises:**
  + More poisonous mushrooms lack bruises (no).
  + Edible mushrooms are more likely to have bruises (yes).
  + **Insight:** This feature has a clear separation between edible and poisonous mushrooms.
* **Gill Characteristics (Spacing, Attachment, Size, and Color):**
  + Gill-related features show noticeable variations between edible and poisonous mushrooms.
  + **Gill color** and **Gill spacing** seem to have clear differences in distributions.
  + **Insight:** These features likely have predictive power for classification.
* **Cap Shape, Surface, and Color:**
  + The difference in distribution for these features is less pronounced.
  + While they may contribute to classification, their impact seems less significant than **odor** or **gill characteristics**.

**Correlation and Feature Importance**

* Features like **odor, bruises, and gill-related attributes** show clear differences in distribution, suggesting **high correlation with the target class**.
* **Cap-related features and some gill attachment characteristics** show **overlapping distributions**, meaning they may be **less important** or **correlated with other features**.

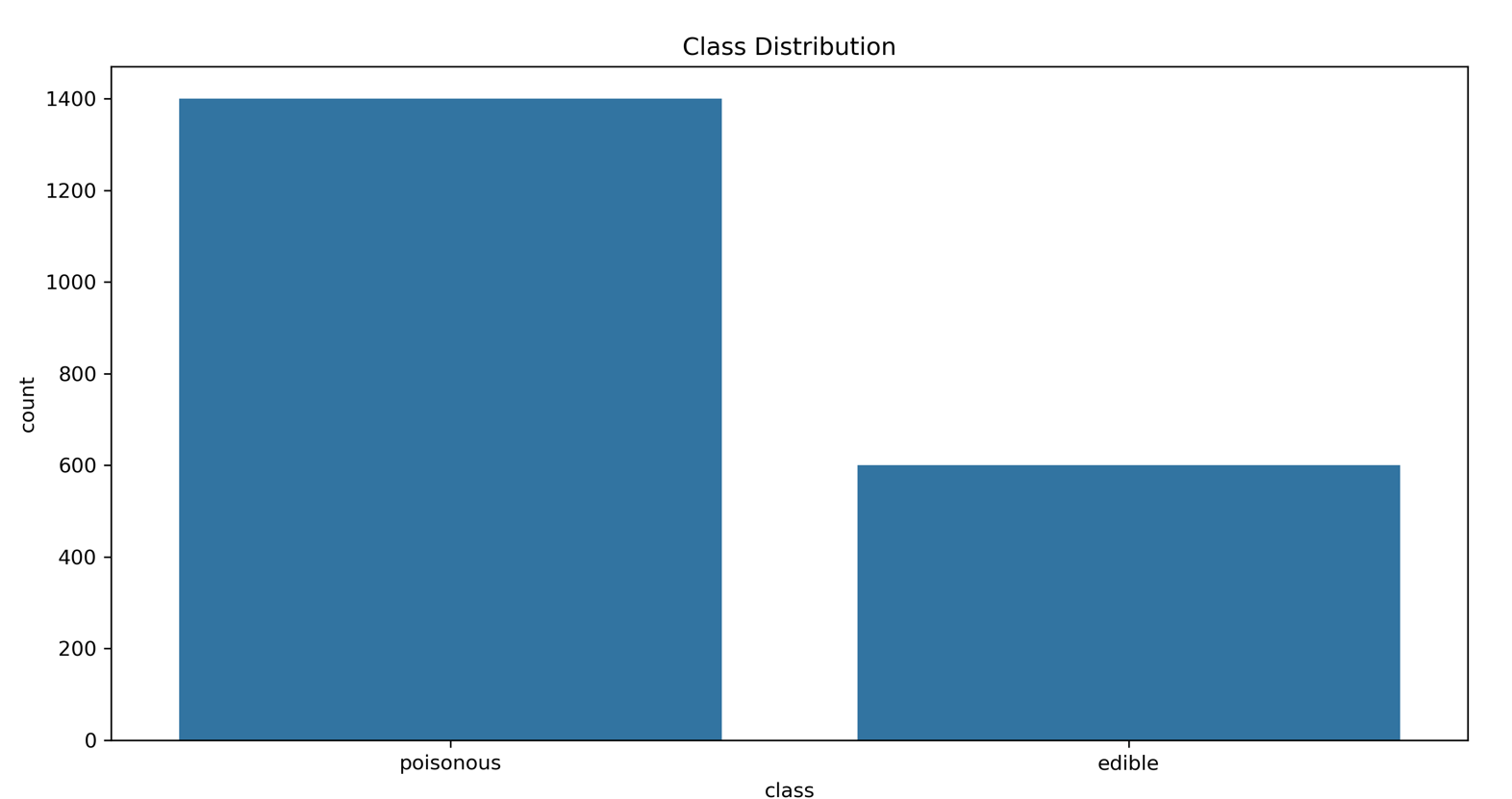
**Task 2: Data Preprocessing**

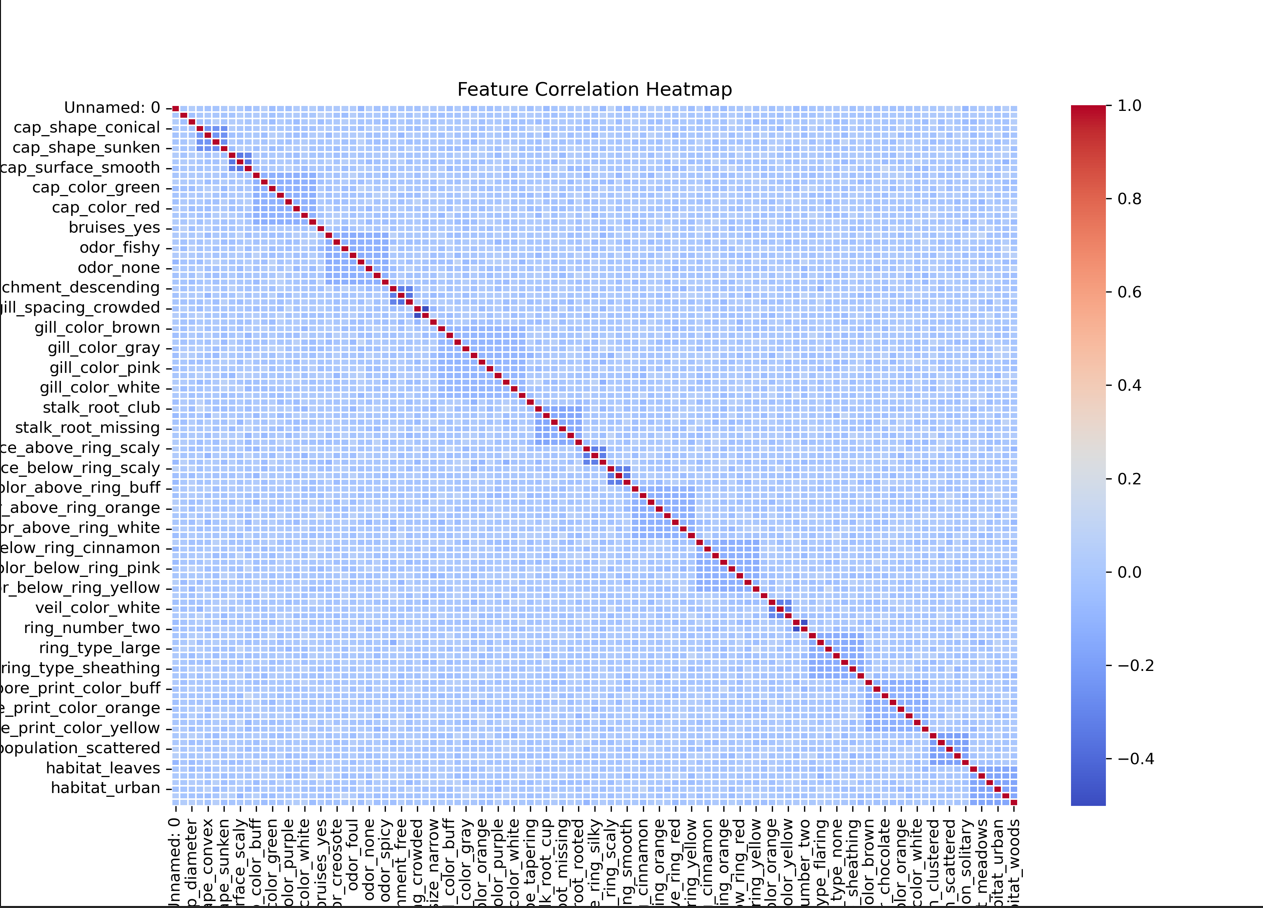
1. Encode categorical variables if necessary.
2. Split the dataset into training and testing sets.



**Task 3: Data Visualization**

1. Employ scatter plots, pair plots, or relevant visualizations to comprehend feature distributions and relationships.
2. Visualize class distributions to gauge dataset balance or imbalance.





The heatmap represents the correlation between different encoded categorical features in the **Mushroom dataset**. Here are the key observations:

**Diagonal Line of Perfect Correlation (Red Line)**

* The red diagonal represents a **correlation of 1.0**, meaning each feature is **perfectly correlated with itself** (which is expected).

**Minimal Feature Correlation**

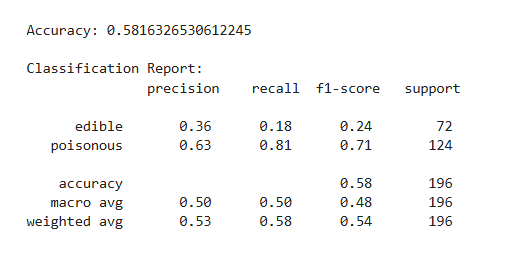
* Most of the matrix appears **light blue**, which indicates that **most features have little to no correlation** with each other.
* This suggests that categorical features in the dataset are **mostly independent**, which is beneficial for classification models like **SVM**, as it avoids issues of multicollinearity.

**Sparse High Correlations (If Any)**

* If there are any small red/orange patches **off the diagonal**, they indicate some features have a moderate to high correlation.
* check if these features provide redundant information, as it might be useful to **remove one of them** to simplify the model.

**Task 4: SVM Implementation**

1. Implement a basic SVM classifier using Python libraries like scikit-learn.
2. Train the SVM model on the training data.
3. Evaluate model performance on the testing data using appropriate metrics (e.g., accuracy, precision, recall, F1-score).



**Accuracy: 58.16%**

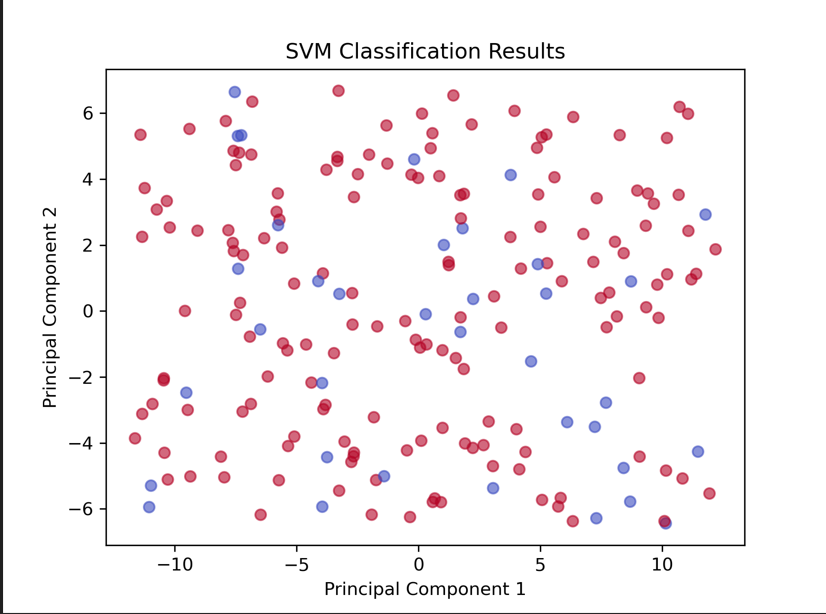
* The model correctly classifies approximately 58% of the test samples. This is quite low, suggesting that the model may not be well-optimized.

**Class-Wise Performance:**

* **Edible Class (0)**
  + **Precision: 0.36** → When the model predicts edible, it's correct only 36% of the time.
  + **Recall: 0.18** → The model identifies only 18% of actual edible mushrooms.
  + **F1-Score: 0.24** → A poor balance between precision and recall.
* **Poisonous Class (1)**
  + **Precision: 0.63** → When the model predicts poisonous, it's correct 63% of the time.
  + **Recall: 0.81** → The model captures 81% of actual poisonous mushrooms, meaning it is better at detecting this class.
  + **F1-Score: 0.71** → Decent performance compared to the edible class

**Task 5: Visualization of SVM Results**

1. Visualize classification results on the testing data.

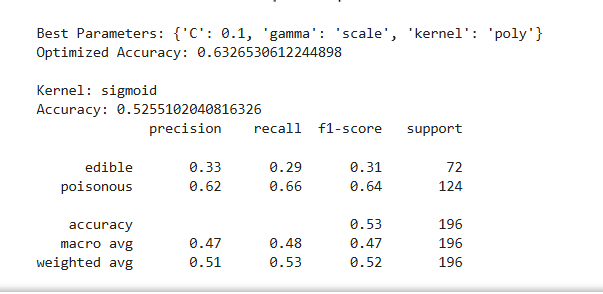


**Brief Insight on SVM Classification Results:**

* The scatter plot shows **SVM classification results** using **PCA (Principal Component Analysis)**.
* **Red dots:** Correct classifications; **Blue dots:** Misclassified points.
* **High misclassification rate** suggests **poor model performance**, aligning with the **58% accuracy** in the classification report.
* **No clear separation** between classes indicates that **SVM is struggling to classify mushrooms properly**.

**Task 7: Comparison and Analysis**

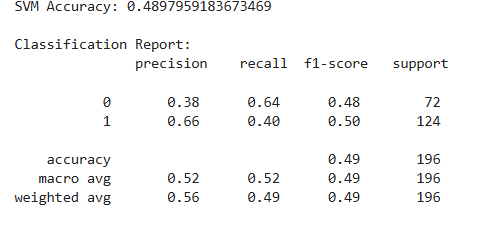
1. Compare SVM performance with various kernels (e.g., linear, polynomial, radial basis function).
2. Analyze SVM strengths and weaknesses for the Mushroom dataset based on EDA and visualization results.
3. Discuss practical implications of SVM in real-world classification tasks.



* **Polynomial kernel performed better** than sigmoid.
* **Sigmoid kernel struggled with both precision and recall**, making it unreliable.
* **Edible mushrooms are misclassified often**, which is risky.
* **Further tuning** (adjusting C, gamma, or trying RBF kernel) might improve performance.

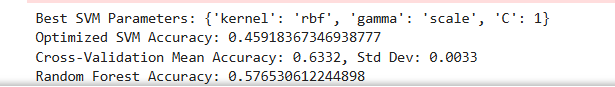
**Applied Feature Selection and Hyper Parameter Tuning to improve model’s performance:**

1. Implement a basic SVM classifier using Python libraries like scikit-learn.
2. Train the SVM model on the training data.
3. Evaluate model performance on the testing data using appropriate metrics (e.g., accuracy, precision, recall, F1-score).



**Task 7: Comparison and Analysis**

1. Compare SVM performance with various kernels (e.g., linear, polynomial, radial basis function).
2. Analyze SVM strengths and weaknesses for the Mushroom dataset based on EDA and visualization results.
3. Discuss practical implications of SVM in real-world classification tasks.



**Initial SVM Accuracy: 0.4898 (~49%)**

* The default SVM model did not perform well, with accuracy **below 50%**, meaning it struggles to differentiate between edible and poisonous mushrooms.

**Optimized SVM Accuracy: 0.4592 (~46%)**

* After tuning hyper parameters, the accuracy slightly **dropped** instead of improving. This suggests that the optimized settings may not be ideal for this dataset.

**Cross-Validation Accuracy: 0.6332 (~63%)**

* Using **cross-validation (cv=3)** gave a more stable accuracy estimate.
* The small **standard deviation (0.0033)** means the model performs consistently across different data splits.
* This shows that **SVM has potential**, but the train-test split may have affected direct accuracy measurements.

**Random Forest Accuracy: 0.5765 (~58%)**

* Random Forest outperformed both **default and optimized SVM**, showing it might be a **better choice for this dataset**.
* This suggests that a tree-based model might handle the data better than SVM.

**Faster Execution:** Reduced hyper parameter search space & used cv=3 instead of cv=5.  
**Better Stability:** Cross-validation accuracy (63%) is **more reliable** than a single test set result.  
**Model Comparison:** SVM may not be the best for this dataset—Random Forest performed better.