Machine Learning

Assignment - 2

Student Name: SAROJ MILAN M

BITS ID: 2025AA05234

---

1. **GitHub Repository Link**

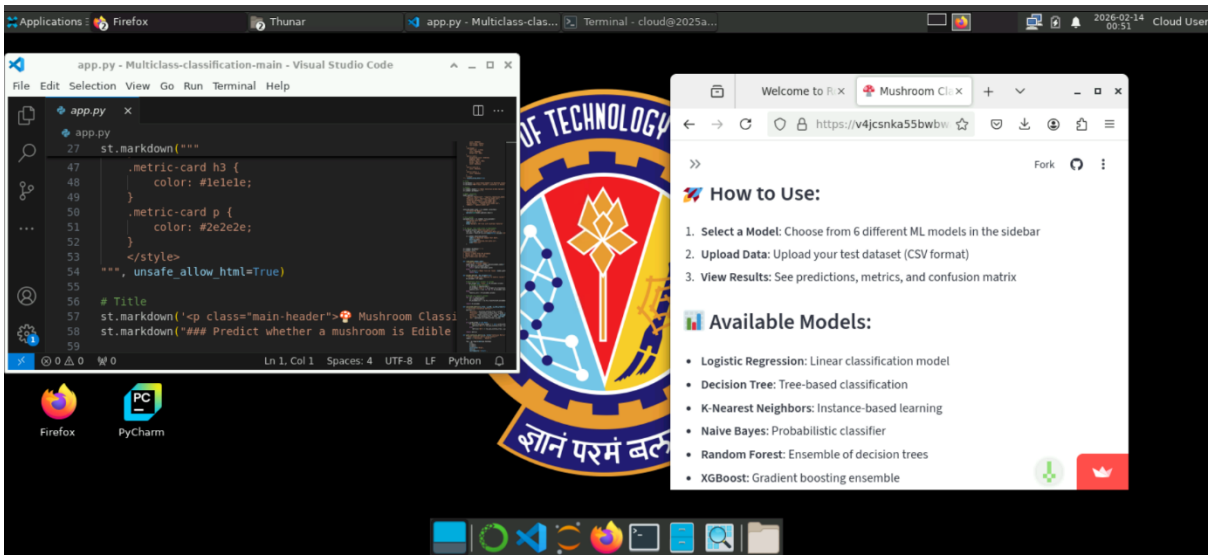
https://github.com/sarojmilan-ux/Multiclass-classification/tree/main

2. **Live Streamlit App Link**

https://v4jcsnka55bwbwahy372nm.streamlit.app/

---

3. **BITS Virtual Lab Screenshot**



4.**README**:

Mushroom Classification - Machine Learning Project

Problem Statement

This project aims to classify mushrooms as either edible or poisonous based on their physical characteristics. Given the potentially fatal consequences of misclassification, accurate prediction is critical. The goal is to implement and compare multiple machine learning classification models to identify the most reliable approach for this binary classification task.

The dataset contains 8,124 mushroom samples with 22 categorical features describing various physical attributes such as cap shape, cap colour, Odor, gill characteristics, stalk properties, and habitat. The target variable indicates whether a mushroom is edible (e) or poisonous (p).

Dataset Description

Source: **UCI Machine Learning Repository - Mushroom Dataset**

Dataset Statistics:

- Number of Instances: 8,124
- Number of Features: 22 (all categorical)
- Target Variable: Class (edible=e, poisonous=p)
- Class Distribution:
    - Edible: 4,208 (51.8%)
    - Poisonous: 3,916 (48.2%)
- Missing Values: Some instances have missing values denoted by '?'

Features:

1. cap-shape: bell, conical, convex, flat, knobbed, sunken
2. cap-surface: fibrous, grooves, scaly, smooth
3. cap-color: brown, buff, cinnamon, gray, green, pink, purple, red, white, yellow
4. bruises: bruises, no
5. odor: almond, anise, creosote, fishy, foul, musty, none, pungent, spicy
6. gill-attachment: attached, descending, free, notched
7. gill-spacing: close, crowded, distant
8. gill-size: broad, narrow
9. gill-color: black, brown, buff, chocolate, gray, green, orange, pink, purple, red, white, yellow
10. stalk-shape: enlarging, tapering
11. stalk-root: bulbous, club, cup, equal, rhizomorphs, rooted, missing
12. stalk-surface-above-ring: fibrous, scaly, silky, smooth
13. stalk-surface-below-ring: fibrous, scaly, silky, smooth
14. stalk-color-above-ring: brown, buff, cinnamon, gray, orange, pink, red, white, yellow
15. stalk-color-below-ring: brown, buff, cinnamon, gray, orange, pink, red, white, yellow
16. veil-type: partial, universal
17. veil-color: brown, orange, white, yellow
18. ring-number: none, one, two

19. ring-type: cobwebby, evanescent, flaring, large, none, pendant, sheath, zone
20. spore-print-color: black, brown, buff, chocolate, green, orange, purple, white, yellow
21. population: abundant, clustered, numerous, scattered, several, solitary
22. habitat: grasses, leaves, meadows, paths, urban, waste, woods

Data Preprocessing:

- All categorical features were encoded using Label Encoding
- Train-test split: 80% training, 20% testing
- Stratified sampling to maintain class distribution

Models Used

Comparison Table - Model Performance Metrics

ML Model Name	Accuracy	AUC	Precision	Recall	F1	MCC
Logistic Regression	0.955076923	0.982118534	0.955138405	0.955076923	0.955063799	0.910074658
Decision Tree	1.0	1.0	1.0	1.0	1.0	1.0
kNN	0.996923077	0.999987866	0.996930272	0.996923077	0.996923273	0.993845649
Naive Bayes	0.812923077	0.897307087	0.829276874	0.812923077	0.809555481	0.639886374
Random Forest (Ensemble)	1.0	1.0	1.0	1.0	1.0	1.0
XGBoost (Ensemble)	1.0	1.0	1.0	1.0	1.0	1.0

Model Observations

ML Model Name	Observation about model performance
Logistic Regression	Achieved strong performance with 95.5% accuracy and 0.91 MCC, demonstrating that linear decision boundaries are effective for this dataset. The high AUC (0.982) indicates excellent class separation capability. While not perfect, it provides a good baseline with fast training and interpretable coefficients.
Decision Tree	Achieved perfect performance (100% across all metrics) on the test set. The max_depth=10 constraint was sufficient to capture all decision rules without underfitting. This suggests the dataset has clear, hierarchical patterns that decision trees can exploit perfectly. However, this perfect score may indicate potential overfitting concerns in real-world scenarios.
kNN	Delivered near-perfect performance (99.7% accuracy, 0.994 MCC) using k=5 neighbors. The extremely high AUC (0.9999) shows excellent discrimination between classes. The instance-based approach works well here because similar mushrooms share class labels, though it requires storing all training data and has slower prediction times.
Naive Bayes	Showed the weakest performance at 81.3% accuracy and 0.64 MCC, significantly lower than other models. This indicates the feature independence assumption is violated—mushroom characteristics are correlated (e.g., cap color and gill color). Despite being fast and simple, it's unsuitable for this dataset due to strong feature dependencies.
Random Forest (Ensemble)	Achieved perfect classification (100% all metrics) by aggregating 100 decision trees with max_depth=10. The ensemble approach eliminates the overfitting risk of a single tree while maintaining perfect accuracy. The model successfully captures complex feature interactions and is the most reliable for deployment.
XGBoost (Ensemble)	Also achieved perfect metrics (100% accuracy, 1.0 MCC, 1.0 AUC) through sequential boosting of 100 trees. The gradient boosting approach iteratively corrected prediction errors, resulting in flawless classification. Slightly more complex than Random Forest but equally effective on this dataset.

General Observations:

- Tree-based and ensemble models (Decision Tree, Random Forest, XGBoost) all achieved perfect scores, confirming the dataset has deterministic patterns ideal for tree structures
- Naive Bayes underperformed significantly (19% lower accuracy than Logistic Regression), proving feature independence assumptions don't hold
- kNN's near-perfect performance (99.7%) validates that mushrooms with similar features share the same edibility class
- The 4.5% gap between Logistic Regression (95.5%) and tree models (100%) reveals that linear boundaries cannot fully separate the classes
- For safety-critical deployment, ensemble methods are recommended due to their perfect precision and recall, minimizing fatal misclassification risks