Data 144 Final Presentation: Identifying Poisonous Mushrooms

Presented by:
Saeyi Oh, Dustin Wallace, Marvin Hsin,
Ryan Joseph Finer, Jade Phay, Avinash Rao

Berkeley
UNIVERSITY OF CALIFORNIA

Introduction

Berkeley
UNIVERSITY OF CALIFORNIA

Research Question

Predicting Edibility of Mushrooms: A Machine Learning Approach

Project Description

We aim to predict whether simulated mushrooms are edible, poisonous, or classified as unsure (treated as poisonous).

What We’re Doing & Why It Matters

Mushroom poisoning is a significant issue, with approximately 704 major harms reported in the United States from 1999 to 2016.

Why it Matters:

- Unintentional ingestion of poisonous mushrooms is common.
- Prevention of mushroom poisoning is crucial.
- Key stakeholders may face risks in their activities.

Who Benefits

Biologists: Enhance their understanding of mushroom classification.

Hikers: Reduce the risk of accidental consumption during outdoor activities.

Foragers: Safeguard against unintentional poisoning while gathering mushrooms.

Why Machine Learning?

Simulated dataset: 20 features including one binary, 17 nominal, and 3 quantitative variables.

Ideal for feature engineering, Logistic Regression, and other ML methods.

Authentic dataset with real mushrooms available for testing models.

Dataset

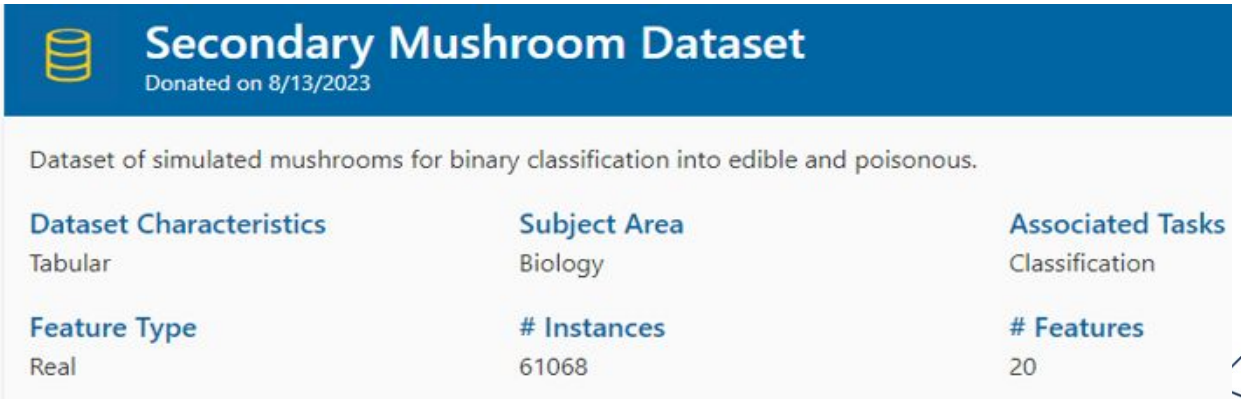
Berkeley
UNIVERSITY OF CALIFORNIA

Dataset

UC Irvine Machine Learning Repository

“Mushroom data creation, curation, and simulation to support classification tasks”

By Dennis Wagner, D. Heider, Georges Hattab. 2021, Published in Scientific Reports



Details

- 61069 hypothetical mushrooms
- 3 continuous features -

  cap-diameter, stem-height, stem-width

- 17 categorical features -

  such as cap-color, veil-type, gill-spacing, habitat, and season

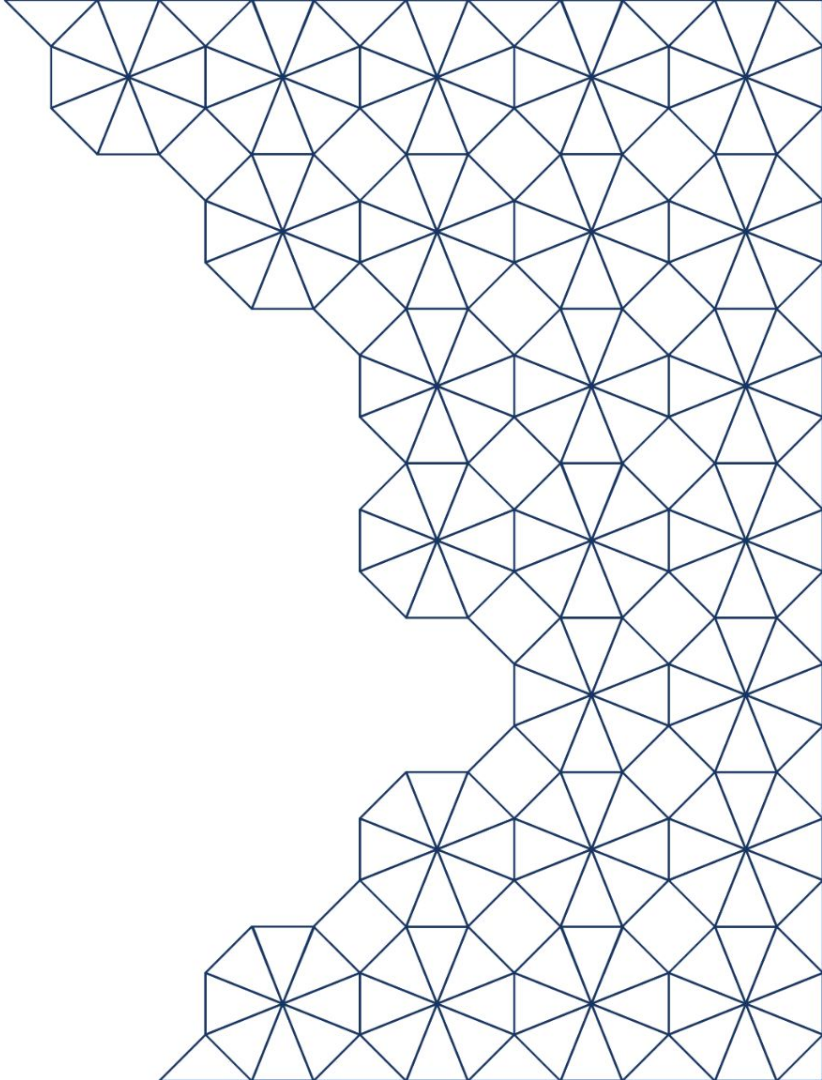
- each mushroom identified as edible or poisonous

Methods

Berkeley
UNIVERSITY OF CALIFORNIA

Methods

- Preprocessing
- Baseline *for comparison*
- MLP Classifier
- Boosted Tree
- Random Forest Classifier
- Logistic Regression
- Vanilla Bagging



Preprocessing

- **One-Hot Encoding**: 10+ features were categorical with 1-3 categories each
- **Test set**: For most models we used train-test split to have a validation set
  - Avoids overfitting
- **EDA w/ Pandas**: cast string columns to float, drop/rename columns,

Baseline

- **Simple Classifier**: Predict all mushrooms are edible
  - **Accuracy**: NOT useful here because 99% of dataset is labeled 'edible’
  - Use Recall Instead
- **Recall**: 56%

1. MLP Classifier

- Model specs:
  - Hidden Layer Size: (10,)
  - Activation: Logistic
  - Max Iterations: 200
  - Solver: lbfgs
- Used all numeric columns
  - Including one-hotted columns

1. MLP Classifier

- *Advantages*
  - Good for complex feature set (20 + features)
  - Good for large input size (60K + records)
- *Disadvantages*
  - Potential overfitting (but mitigated with train/test split)

2. Boosted Tree

```
model2 = GradientBoostingClassifier(random_state=42)
model2.fit(X_train, y_train)
y_test_pred = model2.predict(X_test)
```

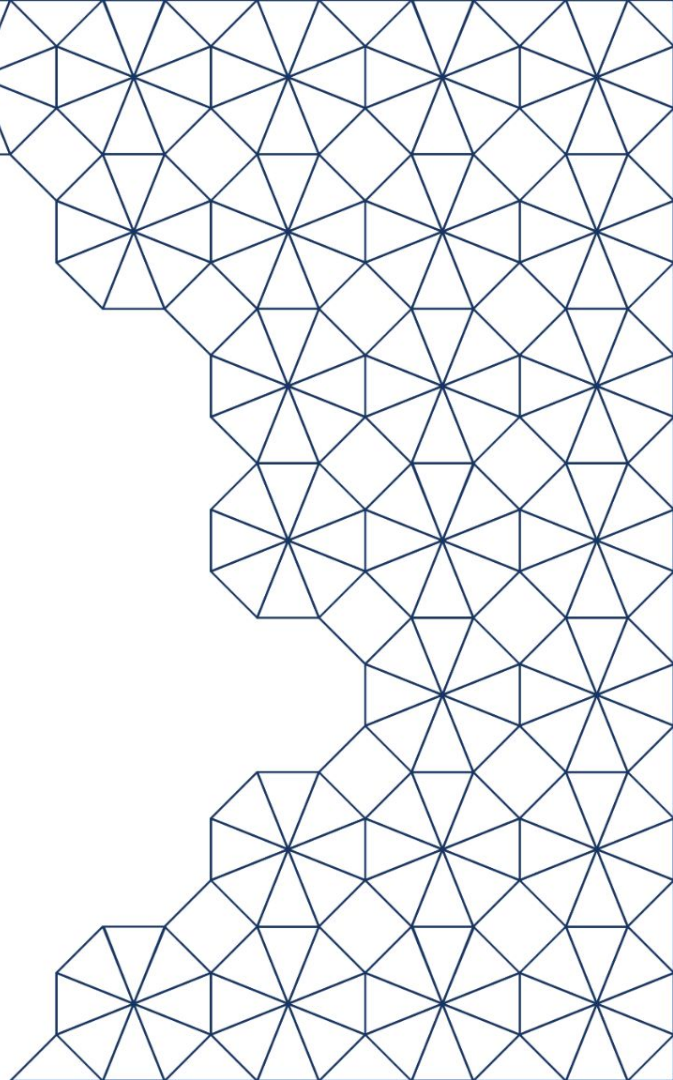
```
Model 2: Boosted Tree Performance on Test Set
Accuracy: 0.9320451940396267
Classification Report:
              precision    recall  f1-score   support

           e       0.92      0.92      0.92      5374
           p       0.94      0.94      0.94      6840

    accuracy                           0.93     12214
   macro avg       0.93      0.93      0.93     12214
weighted avg       0.93      0.93      0.93     12214
```

2. Boosted Tree

- *Advantages*
  - Uses ensemble method of weak learners and averages
  - Assigning weight to incorrect predictions fixes errors
- *Disadvantages*
  - Expensive to train
  - Could be prone to overfitting



3. Random Forest Classifier

- **Model Parameters**:
  - n_estimators = 800
  - max_depth = 100
  - random_state = 42

```
model = RandomForestClassifier(max_features=1, n_estimators=800, max_depth=100, verbose=2, random_state=42)

model.fit(X_train, y_train)

pred = model.predict(X_test)
```

3. Random Forest Classifier

- *Advantages*
  - Can handle linear and non-linear relationships well.
  - Less prone to overfitting compared to normal decision tree
    - (Takes average)
- *Disadvantages*
  - Are not easily interpretable

4. Logistic Regression

- **Model Parameters**:
  - random_state = 42
  - max_iter = 10,000

```
from sklearn.linear_model import LogisticRegression

model_lr = LogisticRegression(max_iter=10000, random_state=42)

model_lr.fit(X_train, y_train)

pred = model_lr.predict(X_test)
```

4. Logistic Regression

- *Advantages*
  - Easy to implement and very efficient to train
  - Easy to interpret
- *Disadvantages*
  - Assumes the linear relationship between dependent and independent variables

5. Vanilla Bagging

- **Model Parameters**:
  - random_state = 42
  - Max_features = num_feature

```
#Vanilla Bagging
from sklearn.ensemble import RandomForestClassifier

#num_feature will be equal to the # of features in the train df
num_feature = len(X_train.columns)

model3 = RandomForestClassifier(random_state=42, max_features = num_feature)
model3.fit(X_train, y_train)
```

5. Vanilla Bagging

- *Advantages*
  - Lower the chance of overfitting by taking average
- *Disadvantages*
  - Extreme Case: If there is one strong predictive feature, it will always select this as first predictor
    - Have all same trees

Results & Conclusions

Berkeley
UNIVERSITY OF CALIFORNIA

Model Performance

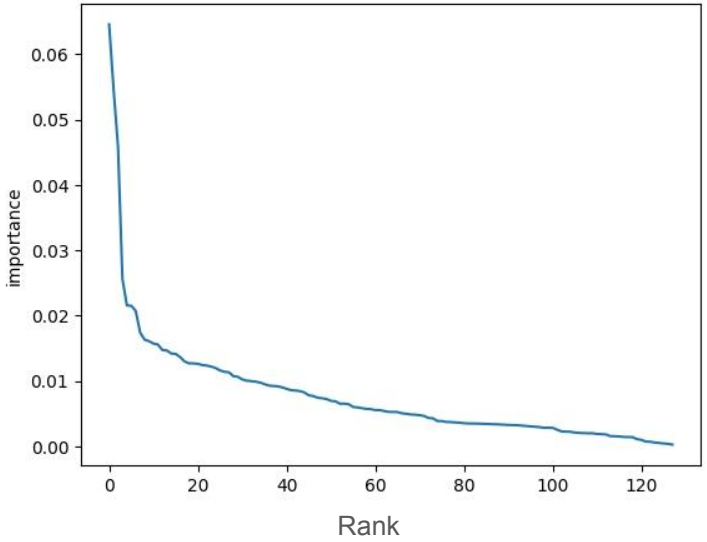
Goal: Minimize false negatives for poisonous mushrooms

- **MLP Classifier**
  - 98% accuracy, 98% recall
- **Gradient Boosting**
  - 93% accuracy, 94% recall
- **Random Forest Classifier**
  - 100% accuracy, 100% recall
- **Logistic Regression**
  - 86% accuracy, 87% recall
- **Vanilla Bagging**
  - 100% accuracy, 100% recall

Bagging Models like Random Forest perform well on the dataset

What's good about Random Forest?

Interpretability!



Feature	Importance
cap-diameter	0.054609
stem-height	0.045532
stem-color_w	0.025480
gill-spacing_c	0.021237
stem-surface_	0.021215
gill-spacing_	0.020742
gill-color_w	0.018048
cap-shape_x	0.016393
gill-spacing_d	0.016110

Top 10 feature importances

Limitation and Improvement

- Collinearity of Variables and PCA
- Cross Validation
- Technical Limitations
- Plant Species and the problem of sampling
    - Dilemma: What do we do when we encounter new mushrooms, will the models hold?

Implications

- Forest Like Data Models are good for finding species and classifying them as correctly poisonous or safe.

- However, with high cost of false negatives and unknown qualities of new species these models would not be great at predicting if a newly found species is poisonous or not.