Project Check-In: Mushroom Classification

Our Mushroom Classification project wants to develop a machine learning model that can classify mushrooms as either poisonous or edible based on their characteristics.

Project Evolution and Changes

The core concepts and methods of our project remain consistent with the original proposal. We use supervised learning techniques to train a model on a dataset labeled with mushroom features. We are now improving the gaussNb model to capture and analyze the information most relevant to whether a mushroom is poisonous or not.

Milestones and Progress

We have successfully completed several major implementation milestones:

1. Data Preprocessing: We have written code to preprocess the Kaggle dataset, handling missing values and encoding categorical features. The dataset is now clean and well-formatted, ready for training.
2. Model Architectures and Hyperparameters:
For the suggestions given to us by TA, we are trying to put various activation functions (e.g., ReLU, tanh) and output functions (e.g., sigmoid, softmax) into our code. The choice of loss function will be based on the problem formulation, with cross entropy being a common choice for classification tasks.
3. Model Implementation: We have implemented a Gaussian Naive Bayes model using the scikit-learn library. The model has been trained on the preprocessed dataset and is capable of classifying mushrooms as poisonous or edible based on their characteristics.
4. Training-test separation and model generalization:
To ensure that our classification model generalizes well to unseen data, we have split the dataset into a training set and a test set by 80-20. The test set will not be used during model training and will only be used for evaluation.

Challenges and Learning Moments

One of the main challenges we encountered was dealing with the categorical nature of the mushroom features. We learned about different encoding techniques and experimented with them to find the most suitable approach.

Next Steps and Anticipated Challenges

In the coming stages, we plan to focus on the following tasks:

1. Fine-tune the Gaussian Naive Bayes model and optimize its hyperparameters.
2. Explore other machine learning algorithms and compare their performance with the current model.
3. Create a user-friendly interface for users to input mushroom characteristics and receive a classification prediction.

Contribution & Work Division

Zirui Zeng: Data preprocessing, report writing
Weichen Zhang: Model implementation, encoding categorical features.
Lily Hu: Model evaluation, documentation
Hansheng Huang: Integration,model testing
Xiangchen Kong: Project management, report review

Sample Code

Here's partial code of our Gaussian Naive Bayes model adapted for the mushroom classification project:

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the mushroom dataset
data = pd.read_csv('mushroom_dataset.csv')

# Preprocess the data
# Perform feature selection and engineering
X = data[['cap_shape', 'cap_color', 'gill_size', 'gill_color', 'stalk_shape', 'stalk_root', 'spore_print_color', 'population', 'habitat']]
y = data['class']

# Encode categorical features
# ...

# Build the Gaussian Naive Bayes model
model = GaussianNB()
model.fit(X, y)

# Make predictions on new data
Xnew = [...] # New mushroom data
ynew = model.predict(Xnew)

# Evaluate the model
accuracy = accuracy_score(y, model.predict(X))
print("Accuracy:", accuracy)
print(classification_report(y, model.predict(X)))

# Print the confusion matrix
mat = confusion_matrix(y, model.predict(X))
# Visualize the confusion matrix
# ...
```

We have successfully preprocessed the dataset, performed feature selection and engineering, and implemented a Gaussian Naive Bayes model. We are excited to continue refining the model and creating a user-friendly interface.