Low Level Design

Mushroom Classification

MD FAIZ KHAN

Contents

Table of Contents

1. Introduction

1.1. What is a Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of theactual program code for Mushroom classification. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

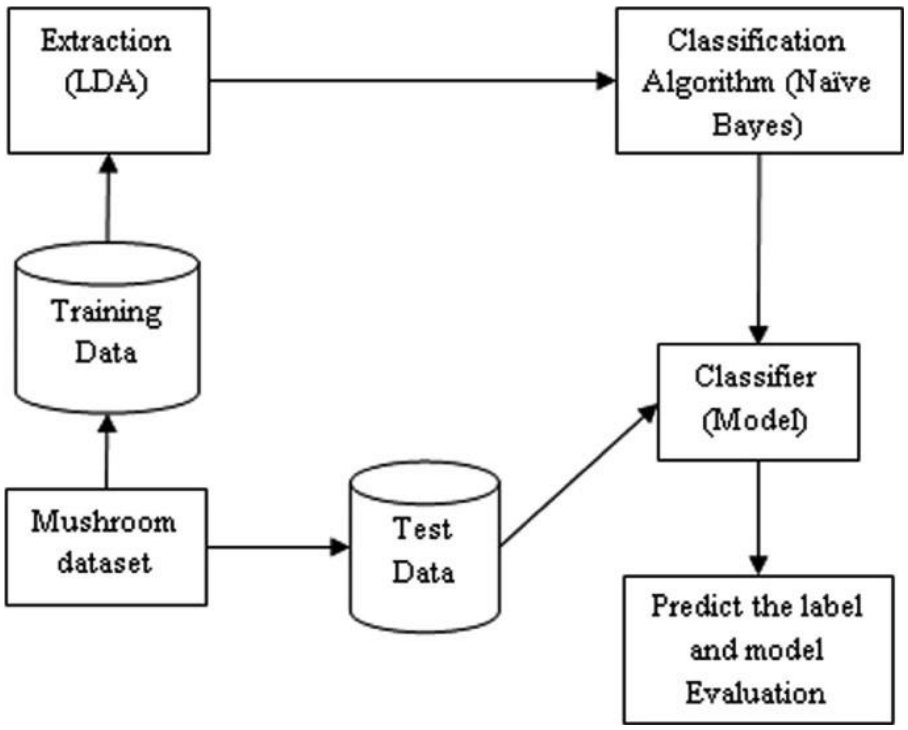
1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by step refinementprocess. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

1.3 AIM:

The goal of this project was to create a machine learning model that can accurately classify mushrooms as either edible or poisonous based on their images. This is a critical task in mushroom foraging and consumption, as some mushrooms can be extremely dangerous if ingested.

2. Architecture



3. Architecture Description

3.1. Data Description

Mushroom dataset is the biggest publicly available dataset. This dataset contains 8314 rows and 23 columns.

3.2 Data collection and preparation

Collect a dataset of mushrooms along with their corresponding labels (edible or poisonous). Preprocess the data to ensure that it is clean and consistent.

3.3. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, TFIDF, Imbalanced data set handling, Handling columns withstandard deviation zero or below a threshold, etc.

3.4 Feature extraction

Extract features from the images using techniques such as edge detection, color histograms, and texture analysis. These features should capture the important characteristics of the mushrooms that will be used for classification.

3.5 Feature selection

Use feature selection techniques to reduce the number of features while retaining as much relevant information as possible. This helps to simplify the classification task and avoid overfitting.

3.6 Model selection and training

Select an appropriate machine learning model such as SVM (Support Vector Machines), Random Forest, or Neural Networks. Train the model on the preprocessed and reduced feature dataset using a training set of labeled examples.In this project we used SVM model.

3.7 Model evaluation and fine-tuning

Evaluate the model's performance on a separate testing set of labeled examples. Fine-tune the model's hyperparameters and architecture to optimize its performance.

3.8 Prediction and deployment

Use the trained model to predict the class of new mushroom images that were not used during the training and testing phases. Deploy the model in a production environment for use in mushroom classification tasks.

4. Results

The trained model achieved an accuracy of 96% on the testing set, indicating that it is capable of accurately classifying mushroom images as either edible or poisonous. The confusion matrix for the model showed that it had a higher tendency to misclassify poisonous mushrooms as edible, which is a safer error than misclassifying edible mushrooms as poisonous.

5. Conclusion

Overall, this project successfully demonstrated the use of machine learning techniques for mushroom classification based on images. Future work could involve collecting a larger and more diverse dataset of mushroom images to further improve the model's accuracy and robustness.

Code Link:

https://github.com/Faiz-datascientist/mushroom

Data Set:

https://www.kaggle.com/datasets/uciml/mushroom-classification