iNeuron

HLD (HIGH LEVEL DESIGN)

MUSHROOM CLASSIFICATION

By BHUPESH PANDEY

Index Page no

Problem Statement :

To build a model to predict whether a mushroom is edible or poisonous , by looking at 23 species of gilled mushrooms we have to identify whether a mushroom is edible or poisonous

Abstract :

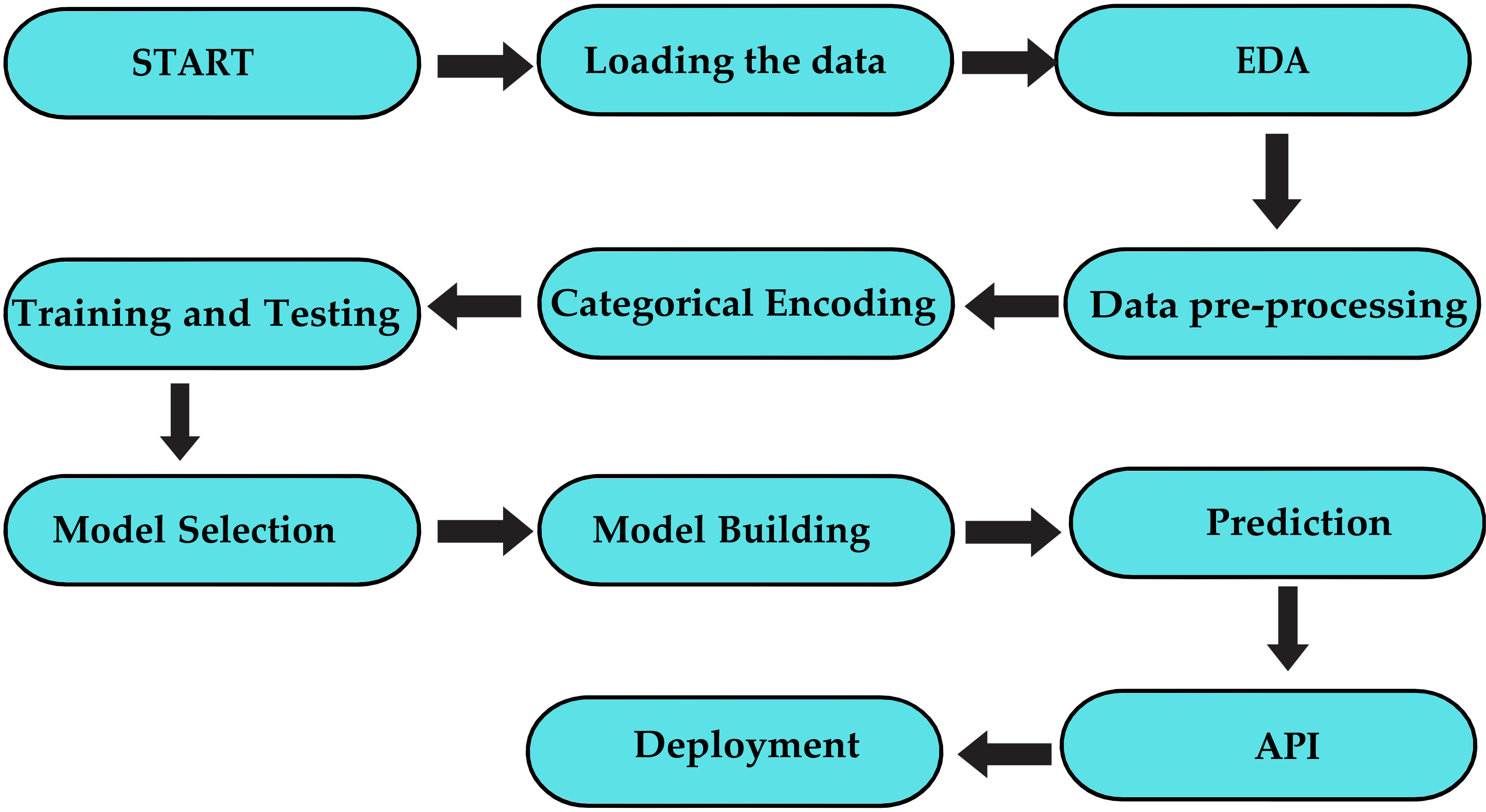
This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be'' for Poisonous Oak and Ivy.

Scope :

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system

The high-level design defines the project level architecture of the system. This architecture defines the sub-systems to be built, internal and external interfaces to be developed, and interface standards identified. The high level design is where the sub-system requirements are developed.

Architecture :



Tools Used :

python™

Pandas

matplotlib

scikit
learn

NumPy

PANDAS
PROFILING

GitHub

Streamlit

render

Python :

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991. Designed to be easy as well as fun, the name "Python" is a nod to the British comedy group Monty Python.

Pandas :

Pandas is defined as an open-source library that provides high-performance data manipulation in Python.

Pandas is a Python library. Pandas is used to analyze data.

Matplotlib :

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

scikit learn :

Scikit-learn is an open source data analysis library, and the gold standard for Machine Learning (ML) in the Python ecosystem. Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.

Numpy :

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

pandas-profiling :

pandas-profiling generates profile reports from a pandas DataFrame . The pandas df. describe() function is handy yet a little basic for exploratory data analysis. pandas-profiling extends pandas DataFrame with df.

GitHub :

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests.

Streamlit :

Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc

Render : render

"Render" is a cloud platform that offers application hosting services for developers. It provides a simplified and scalable infrastructure for deploying and managing web applications, APIs, and other cloud-based services.

Deployment :

To deploy your app to Render, use the git push command to push the code from your local repository's main branch to your render remote.

The Render platform uses Git as the primary means for deploying applications (there are other ways to transport your source code to render, including via an API). When you create an application on render, it associates a new Git remote, typically named render , with the local Git repository for your application.

Deploying an application in Render is a straightforward process. After creating an account, you can easily connect your code repository, configure deployment settings, and initiate the deployment. Render takes care of building and deploying your application, handling dependencies, and configuring the necessary infrastructure. You can monitor the deployment progress in real-time and test your application once it's deployed.