**Range Prediction in a Ballistic System**

**Submitted by:**

**Dhruv Deora**

**PS No. 609590**

**Duration: 1/06/2024 to /07/2024**

**Official Documentation for the project**

**ACKNOWLEDGEMENT**

I would like to thank Mrs. Rohini Hegde (HR Personnel) for providing me the wonderful opportunity to enroll in an internship within the organization. It is my greatest pleasure to present this report. For me it was a unique experience to be a part of L&T and to study about Artificial Intelligence in the PTDC Department. It also encouraged me to gain more knowledge and spiked my interest in this field and helped me gain a broader perspective for my future career and aspirations. I express my sincere thanks to Mr. Anup Kadam, who guided me during my training period, regularly asked about our up-keep and helped in providing all the possible resources. I would also like to thank Mr. Harshit Goyal for his constant support and guidance across all fields through-out the internship period of my project, truly made me feel like home.

**Introduction:**

The range of a projectile is a crucial parameter in a ballistic system. While muzzle velocity, projectile weight and angle of elevation are also critical aspects without which the system will not function accurately or at all, predicting the range remains the core function of the said function. The given task was to integrate machine learning into the system to accurately predict projectile range.

**Project Overview:**

The main objective of this project was to integrate machine learning into the ballistic system to accurately predict the projectile range with respect to various parameters.

Reducing the margin of error to a minimum was at the core of the main objective which was achieved by:

* Extracting the given data and generating exponential amounts of data based on that.
* Followed by training different types of machine learning models based on the generated data.
* Which was validated through cross checking by selecting random columns from the given range table and calculating the margin of error.

**Team Structure**:

Team members are as follows:

1. Mr. Anup Kadam (Head of AICoE) – Responsible for our supervision and guided us about the applicability of the project. Ensured smooth workflow.

2. Mr. Harshit Goyal (Project Manager and Senior Engineer) – Responsible for the project foundation, assignment of tasks, regular team meetings and supervision of work. Guided us through difficult obstacles.

3. Dhruv Deora (Intern) – Responsible for working on the various machine learning models implemented and carrying out the tasks assigned.

4. Vedant Vasaikar (Intern) – Responsible for working on the various machine learning models implemented and carrying out the tasks assigned.

**Project Documentation:**

There are a large variety of machine learning methodologies available for implementation, each with their own structures and models. Testing various models for range prediction was crucial. By utilizing a range of technologies and tools, the project aims to streamline workflow efficiency, ensure precision and accuracy in outcomes, and potentially broaden its use-cases and application across other domains.

**Technologies Used:**

Python Libraries:

* Sci-kit Learn
* Pandas
* NumPy
* Matplotlib
* JobLib
* Seaborn
* Pickle

Machine Learning models used:

* Linear Regressor
* Polynomial Regressor (till 10th degree)
* Stochastic Gradient Descent
* Artificial Neural Networks
* Random Forrest Regressor (bootstrap sampling, grid search)
* Decision Tree Regressor

**Project Structure:**

The project consists of several Python scripts organized into, training and test scripts and trained models on which the data is computed on. Each model trained with increase in the number of parameters by which range is affected by.

The parameters are as follows:

1.Muzzle Velocity.

2. Head/tail wind.

3. Projectile weight.

4. Charge temperature.

5. Angle of elevation.

6. Air Temperature.

7. Air Density.

8. Change in bearing.

9. Cross wind.

**Components:**

**1. Data Generation:**

* Generating thousands of rows of data according to the logic based on the prediction proforma and from the data provided which was limited to about a hundred rows.

**2. Data Processing:**

Filtering out inaccurate data and handling missing or inconsistent values and restructuring and saving the data in a .csv file.

* Followed by normalizing the data for the training process.

**3. Model Training and Implementation:**

* Trained the machine learning models on newly generated data and saving the results into a .csv file.

**4. Testing:**

* Cross- validated the models trained by randomly selecting columns from the original data tables and calculated errors.
* Implementing Polynomial Regression of order 1-10
* Inculcating Random Forrest Regressor and decision tree Algorithms to reduce the error.

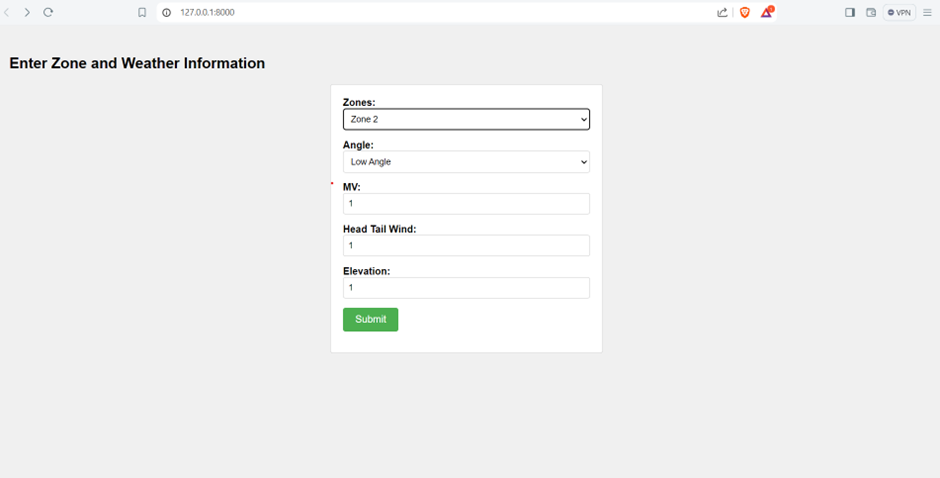
**5. File Structure:**

* Constructing the file structure and the nomenclature in such a manner that it is easier for future developers to understand the process and code.

**UI using Fast API and MongoDB:**

Apart from the assigned project I Created a prototype of a user interface to complement the project by learning Fast API to connect our backend code to frontend which was made using basic HTML for the time being. Also learnt basics MongoDB to potentially store our trained models.

Users will be able to select their parameters as per their requirement and through the integration of MongoDB in Fast API, which is connected to the local host server, the predicted range will be provided as output.



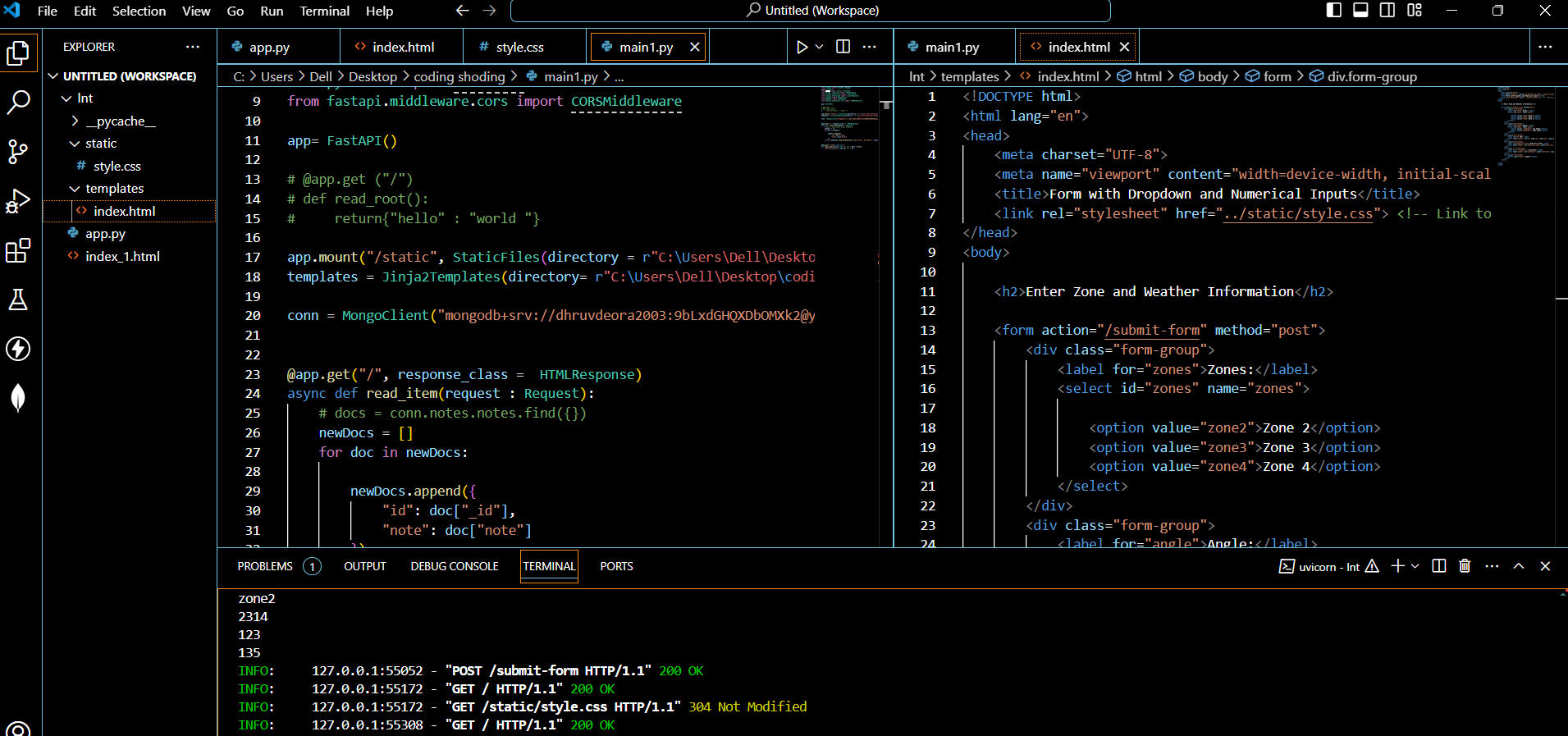
**Libraries used:**

1. FastAPI

2. Uvicorn

3. Pydantic

4. Pymongo



**Langchain:**

Furthermore I researched upon LangChain to load data from a PDF and store it in a vector database.

LangChain is a framework to build with LLMs by chaining interoperable components. LangGraph is the framework for building controllable agentic workflows.

**PDF Loading:**

Langchain uses PyPdf2 for Loading PDF documents and loads PDF using PyPdf into list of documents. Loader chunks the data by ‘page’ and stores page numbers in metadata.

Code:

from langchain\_community.document\_loaders import PyPDFLoader

file\_path = (

    "../../docs/integrations/document\_loaders/example\_data/layout-parser-paper.pdf"

)

loader = PyPDFLoader(file\_path)

pages = loader.load\_and\_split()

pages[0]

PyPdf2 loads the pdf I document format

To store the exracted data we use Vector Store(FAISS)

**FAISS:**

Faiss is a specialized library for vector operations, including similarity search, clustering, and more. It is in integration with LangChain and enhances the capabilities of applications that rely on efficient management and retrieval of high-dimensional vector data.

Code :

from langchain\_community.embeddings.openai import OpenAIEmbeddings

from langchain\_community.vectorstores import FAISS

embeddings = OpenAIEmbeddings()

texts = ["FAISS is an important library", "LangChain supports FAISS"]

faiss = FAISS.from\_texts(texts, embeddings)

**Challenges Tackled:**

* Whilst it was challenging, it was interesting to learn how to interpret the encoded data in the MET tables.
* The interpretation of the corrections in Range Table based on changing values of the parameters.
* Understanding the concept of calculating the angle of elevation using Mathematical interpolation.
* One peculiar problem we faced was the difference in error between High and Low Angle of elevation.
* While the elevation was at a low angle, a simple linear regression model would bring the errors within the required range.
* When the same model was applied at High angle elevation the errors had risen exponentially.
* Solved the crisis of low computational memory for heavier models by enabling CUDA and additionally shifted onto Ubuntu and used Linux.