**Earthquake Magnitude predictive model**

**Project Overview**

**Objectives**

The primary objective of this project was to develop a machine learning model capable of predicting earthquake magnitudes based on relevant features. The project involved data gathering, cleaning, feature engineering, and the implementation of a predictive model. The model was deployed using FastAPI to serve predictions to a Flutter mobile application.

**Data Collection and Cleaning**

**Data Source**

The earthquake dataset used in this project was obtained from [https://earthquake.usgs.gov/earthquakes/search/]. It contained records of seismic events for the last 12 years along with various attributes such as location, depth, and magnitude.

**Data Cleaning**

using python libraries for data manipulation and memory efficient computations

**Pandas:** was utilized for data manipulation, and a crucial step involved handling missing values. **From Sklearn:**

The **KNN imputer** was employed to impute missing values based on the similarity of feature vectors. This approach ensured a robust handling of missing data, enhancing the quality of the dataset for subsequent analysis.

**Feature Selection**

**Random Forest Regressor**

To identify the most effective features for earthquake magnitude prediction, a Random Forest Regressor was employed. This algorithm provides insights into feature importance, aiding in the selection of the most influential variables. The results guided the feature selection process, contributing to the model's accuracy.

A diagram of a tree

Description automatically generated

**Model Development**

**Voting Regressor**

The machine learning model for earthquake magnitude prediction was constructed using the Voting Regressor ensemble technique. **This model combined the predictions of multiple base estimators, enhancing the overall predictive performance**. The choice of base estimators and their hyperparameters was determined through a systematic evaluation process.

**A diagram of a output

Description automatically generated**

**Model Evaluation**

**Performance Metrics**

The performance of the earthquake magnitude prediction model was evaluated using appropriate metrics, such as **Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) and R2 score.** These metrics provided a quantitative measure of the model's accuracy, enabling a comprehensive assessment of its predictive capabilities.

A black background with white text

Description automatically generated

**Training model scrores:**

RMSE = 0.03670422062728734

R2-Score = 0.9909718299743652

**Testing model scores:**

RMSE = 0.19458402427156873

R2-Score = 0.7776838577313727

**Model Deployment**

**Collaborative with app developer**

**FastAPI Integration**

The trained model was deployed using FastAPI, a modern, fast web framework for building APIs with Python 3.7+. FastAPI facilitated the creation of a robust API endpoint, allowing seamless integration with the Flutter mobile application.

A diagram of a flowchart

Description automatically generated

A diagram of a model serving

Description automatically generated

**Conclusion**

This project successfully addressed the challenge of earthquake magnitude prediction through a comprehensive approach encompassing data cleaning, feature selection, and model development. The ensemble technique employed demonstrated the effectiveness of combining diverse predictive models to enhance overall performance. The deployment of the model through FastAPI ensures accessibility and scalability for real-time predictions.

**Future Considerations**

* Potential enhancements to the model, such as incorporating additional features or experimenting with different ensemble methods.
* Continuous monitoring and retraining of the model to adapt to changing patterns in seismic data.

**Acknowledgments**

We acknowledge the contributions of all team members and express gratitude for the support received throughout the project.