High Level Document(HLD)

End to End Oil Price Predictor

Contents:

1. Introduction
2. Project Overview
3. Technologies and Tools Used
4. Architecture of the model
5. Data Collection and Preparation
6. Model Development
7. Web Application Development
8. Deployment
9. Usage Instructions

10.Advantages of the Model

11. Conclusion

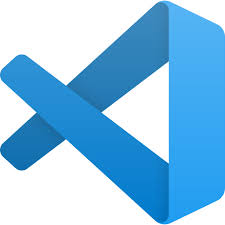
1. Introduction

The Oil Price Prediction project is an end-to-end solution designed to forecast the price of oil in USD based on specific user inputs, including the date, day, and month. The project is implemented using Python and several libraries, with a user-friendly interface built using Streamlit. The application is deployed on Streamlit Cloud, making it accessible globally through the link: (https://pwskills-oilpriceprediction.streamlit.app/).

2. Project Overview

This project aims to predict oil prices based on temporal features. Users provide input data (date, day, month), and the model predicts the corresponding oil price. The solution integrates data processing, model training, and deployment, providing a seamless user experience from input to prediction.

3. Technologies and Tools Used

**- Python**: The primary programming language used for data manipulation, model training, and application development.

**- VS Code**: The integrated development environment (IDE) used for writing, debugging, and running the code.

**- Pandas**: A powerful data manipulation and analysis library used for handling the dataset.

**- NumPy**: A library for numerical computing, essential for array operations and efficient data handling.

- **Seaborn**: A visualization library built on top of Matplotlib, used for creating informative visualizations.

**- Streamlit**: A framework for building web applications in Python, used to create the user interface for the oil price prediction model.

**- Firebase**: Integrated for user authentication and management.

**- Streamlit Cloud**: The platform used to deploy the web application, making it accessible globally.

4. Architecture of model:



5. Data Collection and Preparation

Data is collected from relevant sources, focusing on historical oil prices and temporal data (dates, days, months). The dataset is processed using Pandas to handle missing values, normalize data, and engineer features that improve the model's predictive power.

Steps Involved:

**- Data Cleaning**: Handling missing or inconsistent data.

**- Feature Engineering**: Creating additional features such as day of the week, month, etc., to improve model accuracy.

**- Data Splitting**: Dividing the data into training and testing sets for model evaluation.

6. Model Development

The predictive model is developed using Python’s machine learning libraries. Various models are evaluated, and the one with the best performance is selected. The final model is trained on the entire dataset and saved as a `model.pkl` file for deployment.

**Key Steps:**

**- Model Selection:** Choosing the best algorithm for prediction (e.g., Linear Regression, Random Forest).

**- Training:** The model is trained on historical data to learn the relationship between input features and oil prices.

**- Evaluation**: The model’s performance is evaluated using metrics such as Mean Absolute Error (MAE).

**- Saving the Model:** The trained model is serialized using `pickle` for deployment.

7. Web Application Development

The user interface for the model is built using Streamlit. Users can input the date, day, and month, and the application returns the predicted oil price in USD. The interface is designed to be intuitive and responsive, allowing for easy interaction.

**Key Features:**

**- Input Fields**: Users can enter the date, day, and month.

**- Prediction Display:** The predicted price is displayed clearly on the screen.

**- User Authentication:** Integrated Firebase for user login and registration.

8. Deployment

The application is deployed on Streamlit Cloud, making it accessible worldwide. Deployment ensures that the model is available to users in real-time, with the capability to handle multiple requests simultaneously.

**Deployment Steps:**

- Configure `requirements.txt` to include all dependencies.

- Deploy the Streamlit app on Streamlit Cloud.

- Ensure that the model file (`model.pkl`) and necessary assets are included in the deployment.

9. Usage Instructions

1. Access the application via (https://pwskills-oilpriceprediction.streamlit.app/).

2. Register or log in to your account.

3. Enter the desired date, day, and month.

4. Click on the "Predict Price" button to view the predicted oil price in USD.

5. The prediction will be displayed on the same page.

10. Advantages of the Model

**- User-Friendly Interface:** The Streamlit-based interface is simple to use, making it accessible to users with minimal technical knowledge.

**- Real-Time Prediction**s**:** The model provides instantaneous predictions, enabling quick decision-making.

**- Global Accessibility:** Deployed on Streamlit Cloud, the application is accessible to users worldwide.

**- Scalability:** The application can handle a large number of users simultaneously, thanks to the robust deployment infrastructure.

**- Security:** User authentication and data protection are ensured through Firebase integration.

11. Conclusion

The Oil Price Prediction project is a comprehensive solution that leverages machine learning to predict future oil prices based on user inputs. The project demonstrates the power of integrating data science with web technologies to create impactful tools that can be accessed by users globally. With its deployment on Streamlit Cloud, this application is a testament to the potential of cloud-based machine learning solutions.

This high-level document provides an overview of the project, detailing the steps involved in development, deployment, and usage, as well as highlighting the benefits of using this model.

Thank You……..