**Wine Quality Prediction using Machine Learning**

**Overview**

This project involves building a machine learning model to predict the quality of wine based on its chemical properties, using a dataset of wine characteristics. The solution leverages various machine learning techniques and is deployed through a web app that allows users to input data and receive predictions in real-time.

**Setup and Usage**

**1. Clone the Repository**

To get started, clone the repository to your local machine using the following command:

git clone https://github.com/yourusername/wine-quality-prediction.git

cd wine-quality-prediction

**2. Install Dependencies**

Ensure you have pip installed, then install the required dependencies using:

**3. Run the Django Web App Locally**

To run the web app locally, use the following command:

python app.py

This will start a Django development server, and you can visit http://127.0.0.1:8000/ in your browser to access the Wine Quality Predictor web app.

**4. Model Training and Deployment**

The machine learning model was trained using the **Wine Quality Dataset**. The dataset includes various features like acidity levels, residual sugar, alcohol content, and more. A **Random Forest Regressor** model was used to predict the wine quality based on these features.

To train the model, run the script train\_model.py:

python train\_model.py

5. Deployment on Heroku (or Render)

For cloud deployment, follow the steps to deploy the web app to Heroku (or Render):

Create a Heroku (or Render) account and install the Heroku CLI.

Push the app to the cloud using:

git push heroku master

Access the deployed web app via the provided Heroku (or Render) link.

Hugging Face Model

The model has been uploaded to Hugging Face for easy access and use in the future. Here is the link to the Hugging Face model: Wine Quality Prediction Model on Hugging Face

Links to the Project

GitHub Repository: https://github.com/yourusername/wine-quality-prediction

Deployed Web App: https://yourapp.heroku.com (Replace with actual deployed link)

Approach and Steps

1. Data Preprocessing

The first step was to load and clean the dataset. This included handling missing values (if any), scaling numerical features, and encoding categorical features (if any). The dataset was split into training and testing sets, with 80% for training and 20% for testing.

2. Model Training

We used a Random Forest Regressor for the prediction task. The model was trained on the training dataset, and hyperparameters such as the number of trees and maximum depth were optimized using grid search.

3. Model Evaluation

After training the model, we evaluated its performance using the test dataset. The key evaluation metric used was Mean Absolute Error (MAE), but you can also experiment with others such as R-squared and Root Mean Squared Error (RMSE).

4. Model Improvement

To improve model performance, I experimented with various machine learning algorithms, including Support Vector Regression (SVR) and Gradient Boosting Machines (GBM). Additionally, feature engineering and scaling were applied to optimize results.

5. Deployment

The trained model was serialized and deployed in a web app using Django, which allows users to input data through a form. This form sends the data to the backend, where the model processes the input and returns a predicted wine quality score.

Conclusion

This project demonstrates the use of machine learning for real-world applications like wine quality prediction. The model and web app are fully functional and deployed, making it easy for anyone to predict the quality of wine based on various chemical attributes.

python train\_model.py