# Stock Price Predictor

## Project Overview

This project provides a stock price prediction application that utilizes an XGBoost machine learning model to estimate the closing price of a stock. The prediction is based on key fundamental and technical financial indicators, and the application features a user-friendly web interface built with Streamlit.

In addition to the prediction tool, this repository includes a comprehensive Jupyter Notebook that details the entire data science workflow—from initial data exploration and cleaning to feature engineering and model tuning.

## Table of Contents

* [Features](https://www.google.com/search?q=%23features)
* [Data Science Workflow](https://www.google.com/search?q=%23data-science-workflow-stock_pred_xgboostipynb)
* [Installation](https://www.google.com/search?q=%23installation)
* [Usage](https://www.google.com/search?q=%23usage)
* [Model Information](https://www.google.com/search?q=%23model-information)
* [Files in this Repository](https://www.google.com/search?q=%23files-in-this-repository)

## Features

* **Stock Price Prediction**: Estimates the closing price of a stock using a tuned XGBoost model.
* **Interactive Web App**: A simple and intuitive user interface created with Streamlit for easy interaction.
* **Key Predictive Indicators**: Utilizes a combination of essential financial metrics for prediction:
  + Total Revenue (in billions)
  + Earnings Per Share (EPS)
  + Price-to-Earnings (P/E) Ratio
  + GICS Sector
  + *Technical indicators (used in the model, with placeholders in the app)*

## Data Science Workflow (stock\_pred\_xgboost.ipynb)

The Jupyter Notebook provides a detailed, step-by-step guide through the project's data science process:

1. **Data Loading and Initial Exploration**: Importing the necessary datasets (prices-split-adjusted.csv, fundamentals.csv, securities.csv) and performing an initial assessment of their structure and content.
2. **Data Cleaning and Preparation**: Handling missing values, standardizing column names, and identifying and addressing outliers to ensure data quality.
3. **Exploratory Data Analysis (EDA)**: Conducting univariate, bivariate, and multivariate analysis to uncover patterns, correlations, and key trends in the data.
4. **Feature Engineering**: Creating new, valuable features from the existing data. This includes technical indicators such as:
   * Moving Averages (MA)
   * Relative Strength Index (RSI)
   * Moving Average Convergence Divergence (MACD)
5. **Model Building and Evaluation**: Training and evaluating both a baseline Linear Regression model and a more advanced XGBoost Regressor.
6. **Model Tuning**: Using GridSearchCV to find the optimal hyperparameters for the XGBoost model, thereby improving its predictive performance.

## Installation

To run this project on your local machine, please follow these steps:

1. **Prerequisites**: Ensure you have Python 3.x installed.
2. **Install Libraries**: Open your terminal or command prompt and install the required libraries using pip:  
   pip install streamlit pandas numpy scikit-learn xgboost pmdarima joblib

## Usage

1. **Clone the Repository**:  
   git clone <repository-url>  
   cd <repository-directory>
2. Run the Streamlit Application:  
   Before running, make sure the following files are in the same directory as app.py:
   * xgb\_stock\_predictor.joblib
   * scaler.joblib
   * model\_columns.joblib

Execute the following command in your terminal:streamlit run app.py

1. **Interact with the App**:
   * A new tab should open in your web browser with the application running.
   * Use the sidebar to input the financial indicators for the stock you want to analyze.
   * Click the **"Predict"** button to view the estimated closing price.

## Model Information

* **Prediction App (app.py)**: The web application is powered by a pre-trained **XGBoost** model (xgb\_stock\_predictor.joblib). This model was trained on a rich set of fundamental and technical indicators to provide robust and reliable stock price estimations.
* **Model Development (stock\_pred\_xgboost.ipynb)**: The notebook provides a detailed account of the development and tuning process for the XGBoost model. It was chosen for its superior ability to handle complex, non-linear relationships commonly found in financial data.

## Files in this Repository

* app.py: The main Python script for the Streamlit web application.
* stock\_pred\_xgboost.ipynb: The Jupyter Notebook containing the complete data analysis and model development process.
* xgb\_stock\_predictor.joblib: The serialized, pre-trained XGBoost model.
* scaler.joblib: The saved scaler object for data normalization during preprocessing.
* model\_columns.joblib: A list of the feature columns that the model was trained on.