**AI-Based Stock Market Advisor**

**Team Name: Machine Learning Maverick's**

**Team Members:**  
1. M. Lakshmi Ganesh (231FA04955)  
2. G. Pavan Kumar (231FA04768)  
3. Y. Madhu Veera Sankar (231FA04A36)  
4. P. Praveen Kumar (231FA04A01)

# 1. Abstract

This project presents an AI-powered stock market insight and investment recommendation system. The system uses real-time and historical market data to analyze trends, determine user risk profiles, suggest diversified portfolios, and predict future stock prices. Built using Python and Gradio, the solution ensures a seamless UI/UX while adhering to data privacy and compliance standards.

# 2. Introduction

Stock market investments require informed decisions based on complex and dynamic data. This project aims to simplify the process by offering a smart assistant that recommends investment portfolios and predicts stock prices based on historical data, machine learning models, and user-specific profiles.

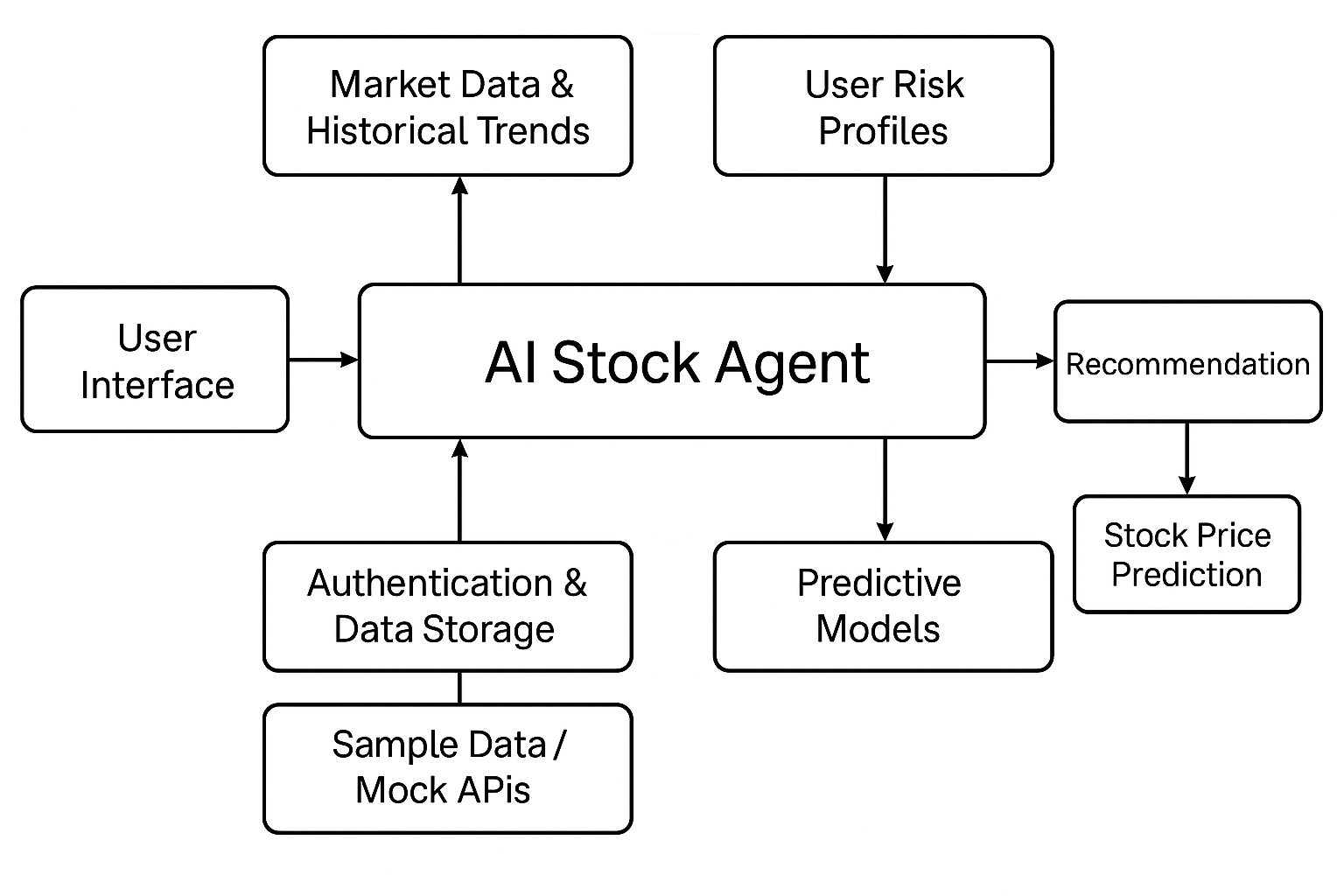
# 3. Literature Review

Various machine learning models, including LSTM and Random Forest, have been employed in previous studies to predict stock trends. Portfolio optimization techniques are also used widely for suggesting diversified investment opportunities. This project integrates both approaches into a unified platform.

# 4. Methodology & Technology Involved

The backend is developed in Python using libraries such as pandas, scikit-learn, and Keras for modeling. Gradio is used for the frontend interface. The platform analyzes user input (age, experience) to generate risk profiles and provide portfolio recommendations. It also reads historical CSV data and predicts future stock prices using machine learning models.

# 5. Block Diagram

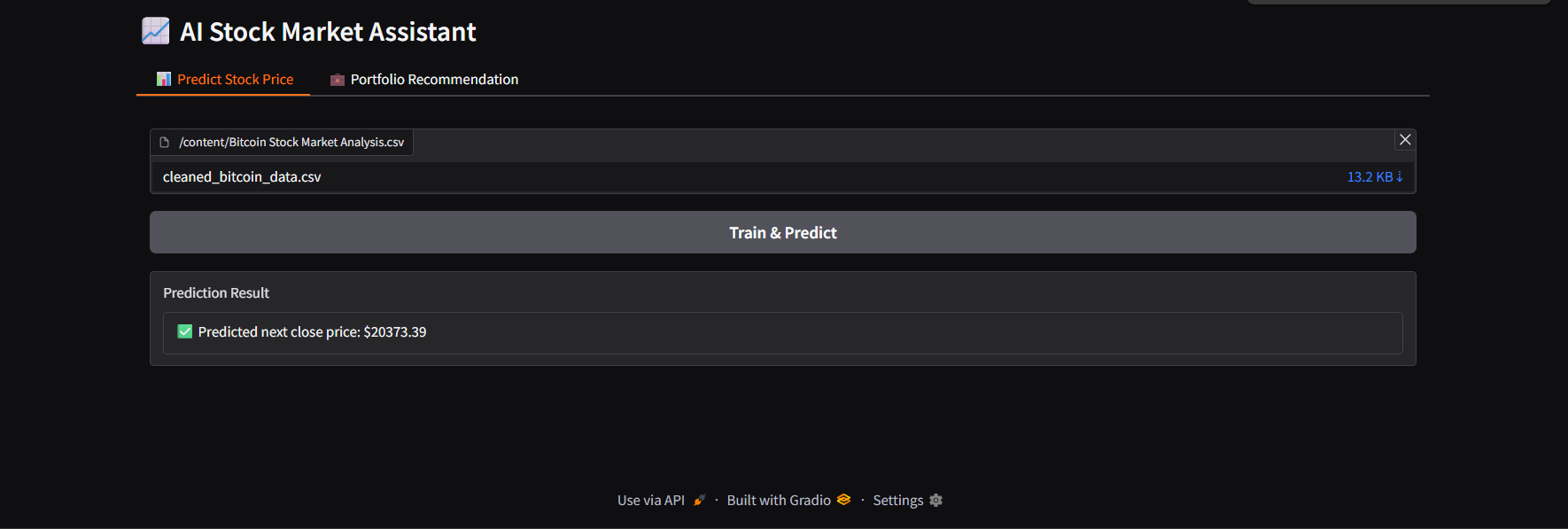


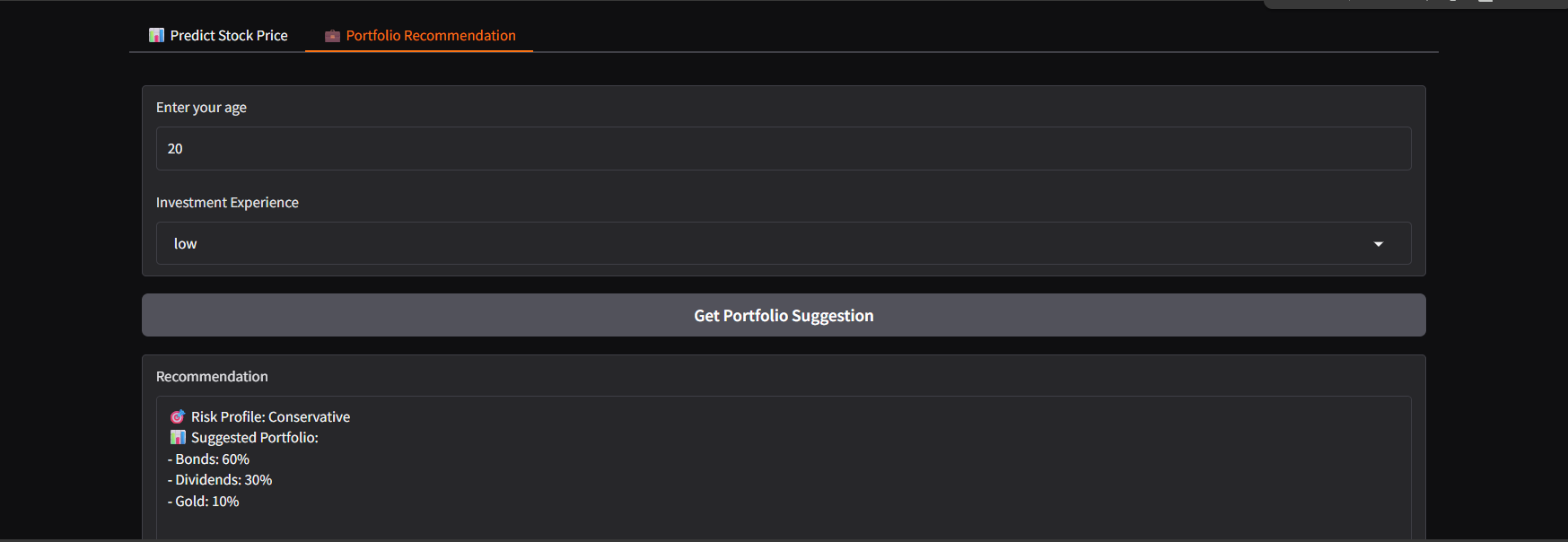
# 6. Complete Code (Python Backend)

import pandas as pd  
import numpy as np  
from keras.models import Sequential  
from keras.layers import LSTM, Dense  
import gradio as gr  
  
def create\_model(input\_shape):  
 model = Sequential()  
 model.add(LSTM(units=50, return\_sequences=True, input\_shape=input\_shape))  
 model.add(LSTM(units=50))  
 model.add(Dense(1))  
 model.compile(optimizer='adam', loss='mean\_squared\_error')  
 return model  
  
def preprocess\_data(df):  
 df = df[['Close']]  
 df = df.dropna()  
 data = df.values  
 scaled\_data = (data - data.min()) / (data.max() - data.min())  
 X, y = [], []  
 for i in range(60, len(scaled\_data)):  
 X.append(scaled\_data[i-60:i, 0])  
 y.append(scaled\_data[i, 0])  
 return np.array(X), np.array(y), data.min(), data.max()  
  
def predict\_price(file):  
 df = pd.read\_csv(file.name)  
 X, y, min\_val, max\_val = preprocess\_data(df)  
 X = X.reshape((X.shape[0], X.shape[1], 1))  
 model = create\_model((X.shape[1], 1))  
 model.fit(X, y, epochs=10, batch\_size=32, verbose=0)  
 last\_60\_days = X[-1].reshape(1, 60, 1)  
 pred = model.predict(last\_60\_days)  
 return f"Predicted next close price: ${pred[0][0] \* (max\_val - min\_val) + min\_val:.2f}"  
  
def get\_portfolio(age, experience):  
 if age < 30 or experience == 'low':  
 return "🧭 Risk Profile: Conservative  
📊 Suggested Portfolio:  
- Bonds: 60%  
- Dividends: 30%  
- Gold: 10%"  
 elif age < 50 or experience == 'medium':  
 return "🧭 Risk Profile: Moderate  
📊 Suggested Portfolio:  
- Stocks: 50%  
- Bonds: 30%  
- Gold: 20%"  
 else:  
 return "🧭 Risk Profile: Aggressive  
📊 Suggested Portfolio:  
- Stocks: 70%  
- Crypto: 20%  
- Gold: 10%"  
  
ui1 = gr.Interface(fn=predict\_price, inputs=["file"], outputs="text", title="Predict Stock Price")  
ui2 = gr.Interface(fn=get\_portfolio, inputs=["number", gr.Dropdown(['low', 'medium', 'high'])], outputs="text", title="Portfolio Recommendation")  
gr.TabbedInterface([ui1, ui2], ["Predict Stock Price", "Portfolio Recommendation"]).launch()

# 7. Results & Discussion

The following screenshots show the functional output of both modules:





# 8. Conclusion & Future Scope

This AI-based stock market assistant successfully provides portfolio recommendations and stock predictions. In the future, real-time data integration, sentiment analysis from financial news, and personalized investment tracking could enhance the tool's effectiveness and reach.

# 9. References

- Scikit-learn documentation  
- Keras documentation  
- Gradio documentation  
- Yahoo Finance historical datasets  
- Research papers on stock market forecasting using LSTM and risk profiling methods