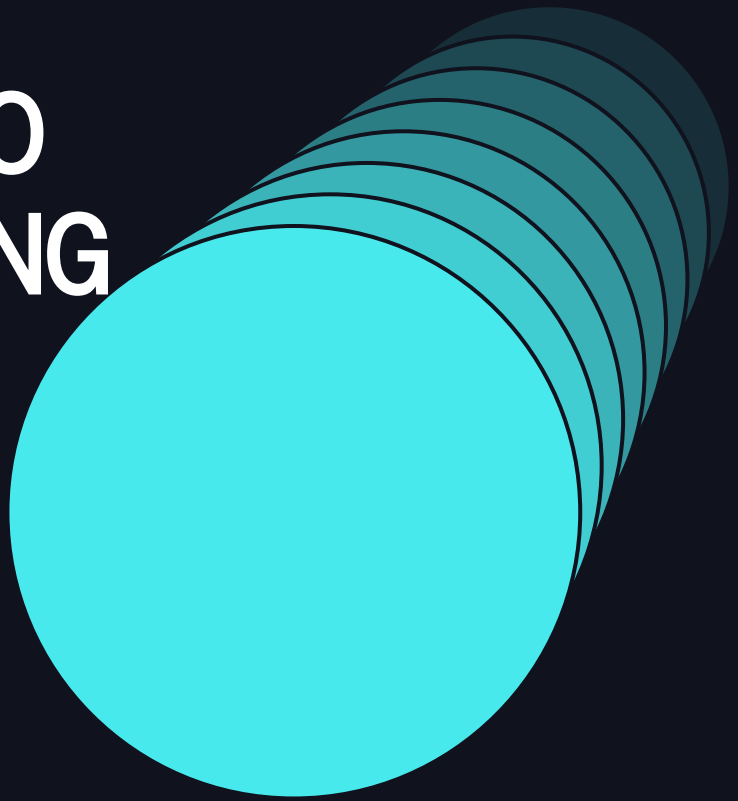
DATA+AI SUMMIT BY databricks

STOCK PORTFOLIO MANAGEMENT USING AI



Ayesha Nasim
AI & ML Engineer

ABOUT THE SPEAKER

- Experienced in **Developing** & **Operating** Technology Agnostic AI-Based Enterprise Scale Products

- Master's in Data Science and Working @ Allegis Group as MLOps Engineer

POWER OF DATA FUELED AI-BASED INVESTMENT DECISIONS

VISION

Portfolio Management Using Multi-Agent Reinforcement Learning

Developed an Automated Stock Portfolio Management System and overall net gain in terms of a dollar amount, using:

- Multi-Agent Reinforcement Learning
- Top 10 Stable & Volatile Stocks each from S&P
- Combination of Leading, Lagging, and Coincident Technical Indicators
- Sentiments from News Publications

TOOLS & TECHNOLOGIES

Development Environment

- Google Collab
- VS Code

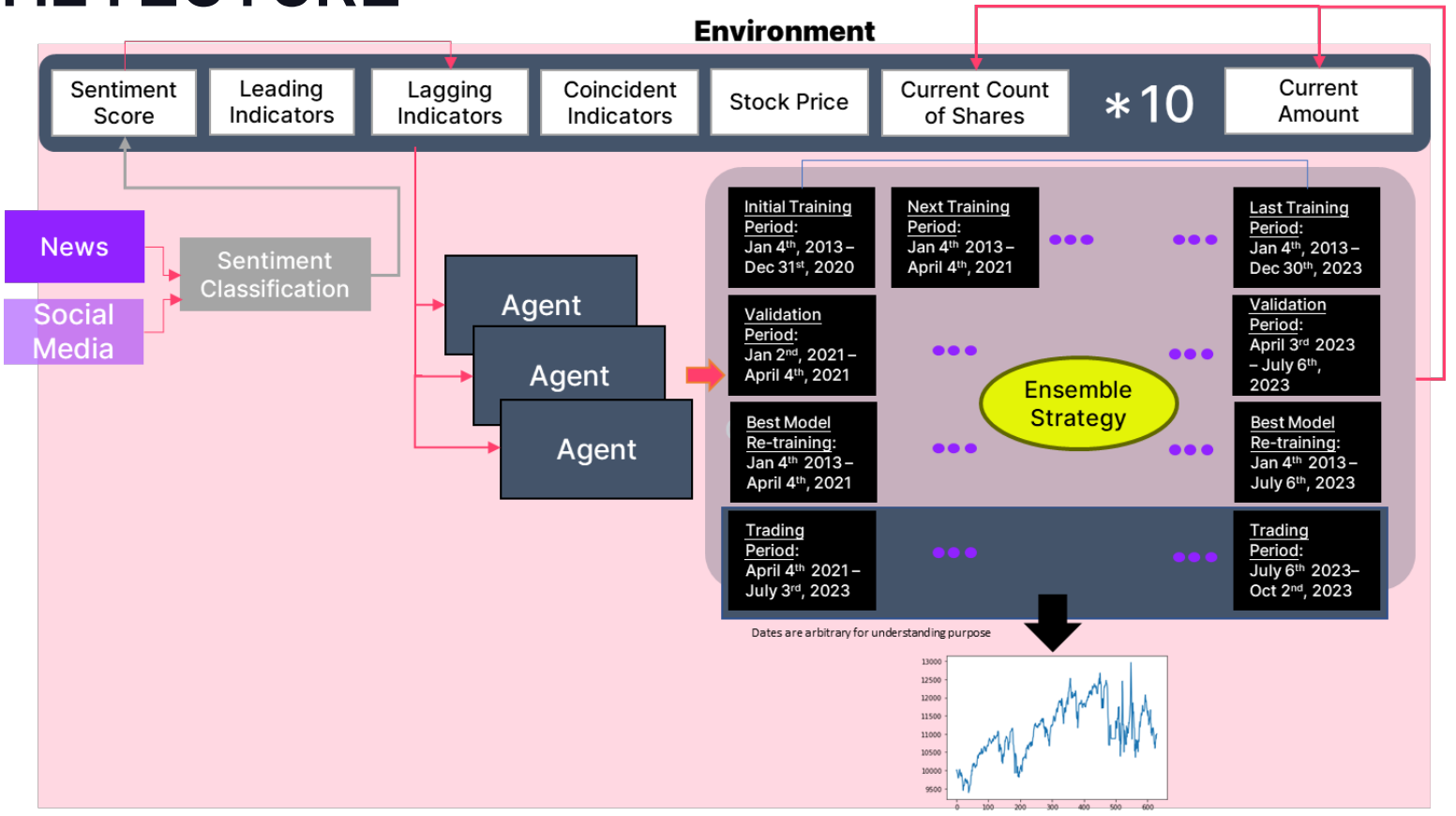
Data Source

- Yahoo Finance (Stock Prices)
- Kaggle (News)
- FRED

Libraries

- Pandas
- TexBlob
- OpenAI Gym
- Stable Baselines3

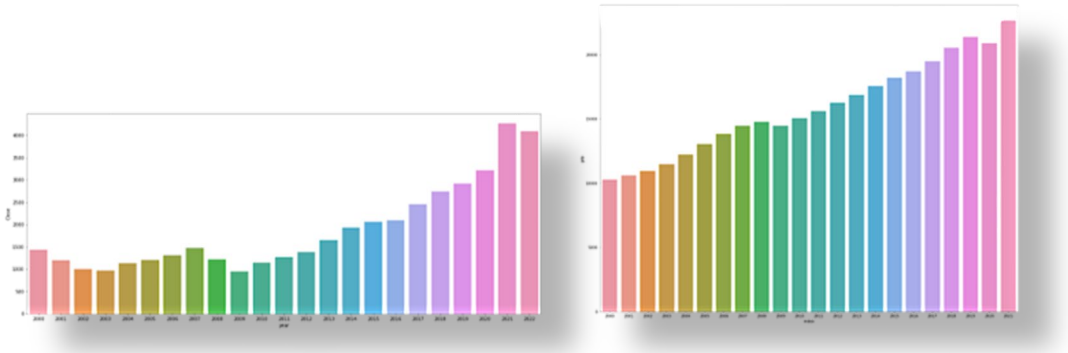
ARCHITECTURE



IMPLEMENTATION

Kick-off

- *Idea Inception*
- *Reviewed Pre-Existing Work*



How effective can a Multi-Agent Reinforcement Learning Algorithm be for Automated Stock Portfolio Management?

Time Period Selection

Explored GDP & Stocks Historical Data

IMPLEMENTATION

Portfolio Selection

Select 5 most and 5 least volatile stocks

	date	open	high	low	close	volume	tic	day
0	2010-01-04	39.099998	39.419998	38.840000	22.922108	2175500	D	0
1	2010-01-04	11.220000	11.430000	10.950000	9.966131	14482500	DAL	0
2	2010-01-04	5.660000	5.970000	5.650000	4.164575	14901600	KEY	0
3	2010-01-04	25.320000	25.910000	24.930000	19.833185	3811400	LNC	0
4	2010-01-04	15.245000	15.350000	15.120000	9.583809	1332800	LNT	0
...	...	...	...	...	...	...	...	...
28549	2020-12-30	50.520000	51.000000	50.360001	47.424179	563800	LNT	2
28550	2020-12-30	147.470001	147.990005	147.009995	138.462997	2224900	PEP	2
28551	2020-12-30	138.600006	138.919998	137.550003	130.316620	3261400	PG	2
28552	2020-12-30	372.339996	373.100006	371.570007	359.862762	49455300	SPY	2
28553	2020-12-30	65.519997	65.849998	65.389999	61.628376	1296400	XEL	2

```
sorted_weekly_volatility.head()
```

	Name	Volatility
148	ENPH	4.283337
239	KEY	3.621506
118	DAL	3.496065
258	LNC	3.488239
156	ETSY	3.460588

Data Collection & Pre-Processing

- *Found News Publication Dataset & Conduct Sentiment Classification*
- *Figured Out Technical Indicators*
- *Refined & Pre-Processed Stocks Dataset*
- *Pre-Processed the Technical Indicators Data Set*

IMPLEMENTATION

Model Construction

Figured out the Reinforcement Learning Strategy, the existing Packages & Libraries

```
import pandas as pd

%matplotlib inline
from finrl.config_tickers import SP_500_TICKER
from finrl.meta.preprocessor.yahoodownloader import YahooDownloader
from finrl.meta.preprocessor.preprocessors import FeatureEngineer, data_split
from finrl.agents.stablebaselines3.models import DRLAgent, DRLEnsembleAgent
from finrl.plot import backtest_stats, backtest_plot, get_daily_return, get_baseline

import pandas_datareader.data as web

import os
from finrl.main import check_and_make_directories
from finrl.config import (
    DATA_SAVE_DIR,
    TRAINED_MODEL_DIR,
    TENSORBOARD_LOG_DIR,
    RESULTS_DIR,
    INDICATORS,
    TRAIN_START_DATE,
    TRAIN_END_DATE,
    TEST_START_DATE,
    TEST_END_DATE,
    TRADE_START_DATE,
    TRADE_END_DATE,
)

check_and_make_directories([DATA_SAVE_DIR, TRAINED_MODEL_DIR, TENSORBOARD_LOG_DIR, RESULTS_DIR])
```

IMPLEMENTATION

Environment Setup

Setup the Environment, Observation & Action Space, and associated methods

- Step 5.2: Define A2C model parameters

```
A2C_model_kwargs = {'n_steps': 5, 'ent_coef': 0.005, 'learning_rate': 0.0007}
```

- Step 5.3: Define PPO model parameters

```
PPO_model_kwargs = { "ent_coef": 0.01, "n_steps": 2048, "learning_rate": 0.00025, "batch_size": 128 }
```

- Step 5.4: Define DDPG model parameters

```
DDPG_model_kwargs = { "buffer_size": 10_000, "learning_rate": 0.0005, "batch_size":64 }
```

```
stock_dimension = len(df_processed.tic.unique())

udfs = 2 # Sentiment Scores, GDP

state_space = 1 + 2 * stock_dimension + len(technical_indicators)*stock_dimension + udfs*stock_dimension

print(f"Stock Dimension: {stock_dimension}, state Space: {state_space}")
```

```
Stock Dimension: 9, state Space: 73
```

Agents Selection

Select Deep RL Agents & their respective hyperparameters

IMPLEMENTATION

Ensemble Strategy Development

Development of Multi-agent strategy

	Iter	Val Start	Val End	Model Used	A2C Sharpe	PPO Sharpe	DDPG Sharpe
0	126	2018-01-02	2018-04-04	A2C	-0.18881	-0.419753	-0.200898
1	189	2018-04-04	2018-07-03	DDPG	-0.213949	-0.020959	0.197413
2	252	2018-07-03	2018-10-02	DDPG	0.163297	0.106038	0.400928
3	315	2018-10-02	2019-01-03	A2C	0.165592	-0.116504	-0.216793
4	378	2019-01-03	2019-04-04	DDPG	0.48494	0.296962	0.69472
5	441	2019-04-04	2019-07-05	DDPG	0.335411	0.31065	0.406546
6	504	2019-07-05	2019-10-03	DDPG	0.066773	-0.035344	0.10168
7	567	2019-10-03	2020-01-03	PPO	0.103287	0.110228	0.027765
8	630	2020-01-03	2020-04-03	DDPG	-0.306114	-0.329034	-0.078555
9	693	2020-04-03	2020-07-06	DDPG	0.06348	0.138119	0.233644

A Higher Sharpe Ratio Indicates Better Risk-Adjusted Performance When Comparing Similar Portfolios

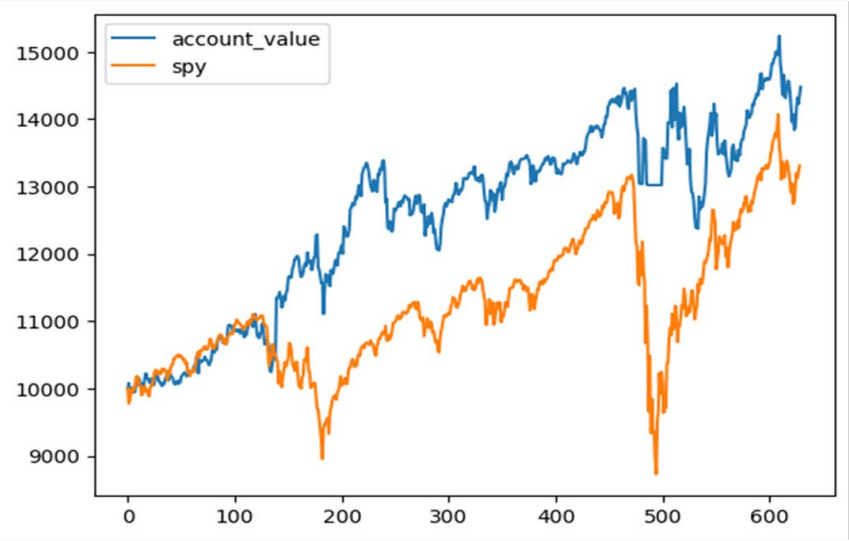
```
ensemble_agent = DRLEnsembleAgent(initial_amount = 10000,
                                  stock_dim = stock_dimension,
                                  df = df_processed,
                                  train_period=(TRAIN_START_DATE,TRAIN_END_DATE),
                                  val_test_period=(TEST_START_DATE,TEST_END_DATE),
                                  rebalance_window=rebalance_window,
                                  validation_window=validation_window,
                                  **env_variables)
```

RESULTS

5 Most & 5 Least Volatile Stocks

Backtest Stats & Account Summary

Cumulative return improved by approximately 35.34% compared to baseline



Annual return	0.159534
Cumulative returns	0.447801
Annual volatility	0.182268
Sharpe ratio	0.904378
Calmar ratio	1.076276
Stability	0.828840
Max drawdown	-0.148228
Omega ratio	1.195666
Sortino ratio	1.325742
Skew	NaN
Kurtosis	NaN
Tail ratio	0.968050
Daily value at risk	-0.022309

dtype: float64

Annual return	0.121354
Cumulative returns	0.330942
Annual volatility	0.236552
Sharpe ratio	0.604095
Calmar ratio	0.359916
Stability	0.554521
Max drawdown	-0.337173
Omega ratio	1.137069
Sortino ratio	0.825409
Skew	NaN
Kurtosis	NaN
Tail ratio	0.702057
Daily value at risk	-0.029236

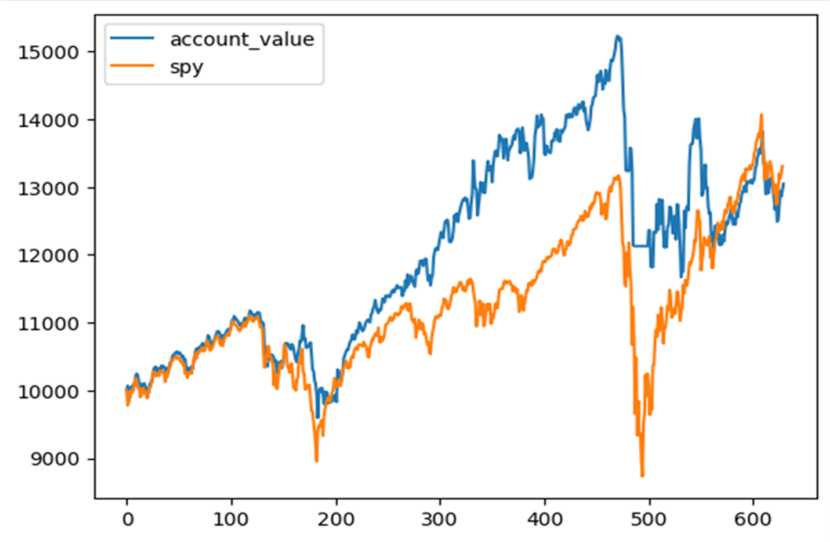
dtype: float64

RESULTS

5 Most Volatile Stocks

Backtest Stats & Account Summary

Cumulative return decreased by approximately 8.56% compared to baseline



Annual return	0.112297
Cumulative returns	0.304823
Annual volatility	0.195913
Sharpe ratio	0.642434
Calmar ratio	0.480721
Stability	0.636771
Max drawdown	-0.233601
Omega ratio	1.132554
Sortino ratio	0.884233
Skew	NaN
Kurtosis	NaN
Tail ratio	0.827437
Daily value at risk	-0.024183

dtype: float64

Annual return	0.121354
Cumulative returns	0.330942
Annual volatility	0.236552
Sharpe ratio	0.604095
Calmar ratio	0.359916
Stability	0.554521
Max drawdown	-0.337173
Omega ratio	1.137069
Sortino ratio	0.825409
Skew	NaN
Kurtosis	NaN
Tail ratio	0.702057
Daily value at risk	-0.029236

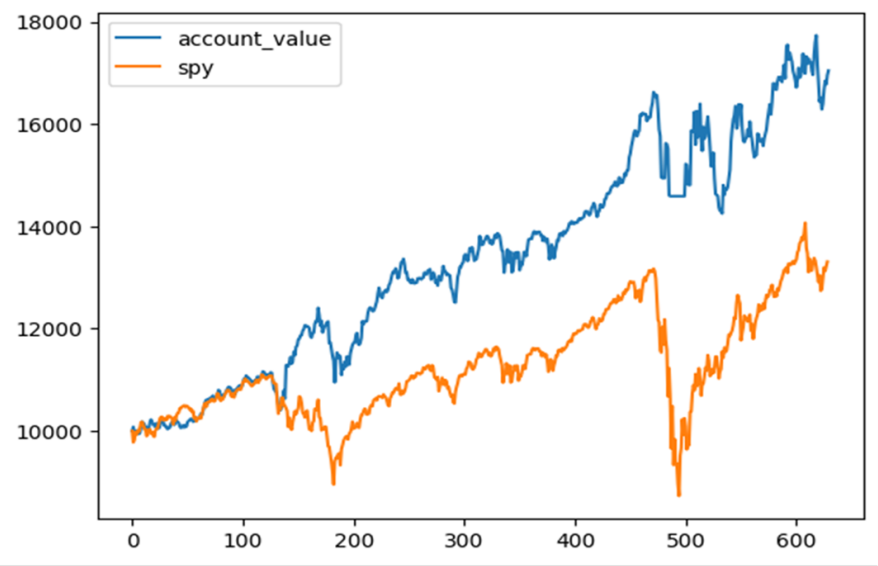
dtype: float64

RESULTS

5 Least Volatile Stocks

Backtest Stats & Account Summary

Cumulative return improved by approximately 53% compared to baseline



Annual return	0.237728
Cumulative returns	0.704367
Annual volatility	0.179996
Sharpe ratio	1.276987
Calmar ratio	1.666635
Stability	0.955641
Max drawdown	-0.142640
Omega ratio	1.281791
Sortino ratio	1.908398
Skew	NaN
Kurtosis	NaN
Tail ratio	1.046383
Daily value at risk	-0.021765
dtype: float64	

Annual return	0.121354
Cumulative returns	0.330942
Annual volatility	0.236552
Sharpe ratio	0.604095
Calmar ratio	0.359916
Stability	0.554521
Max drawdown	-0.337173
Omega ratio	1.137069
Sortino ratio	0.825409
Skew	NaN
Kurtosis	NaN
Tail ratio	0.702057
Daily value at risk	-0.029236
dtype: float64	

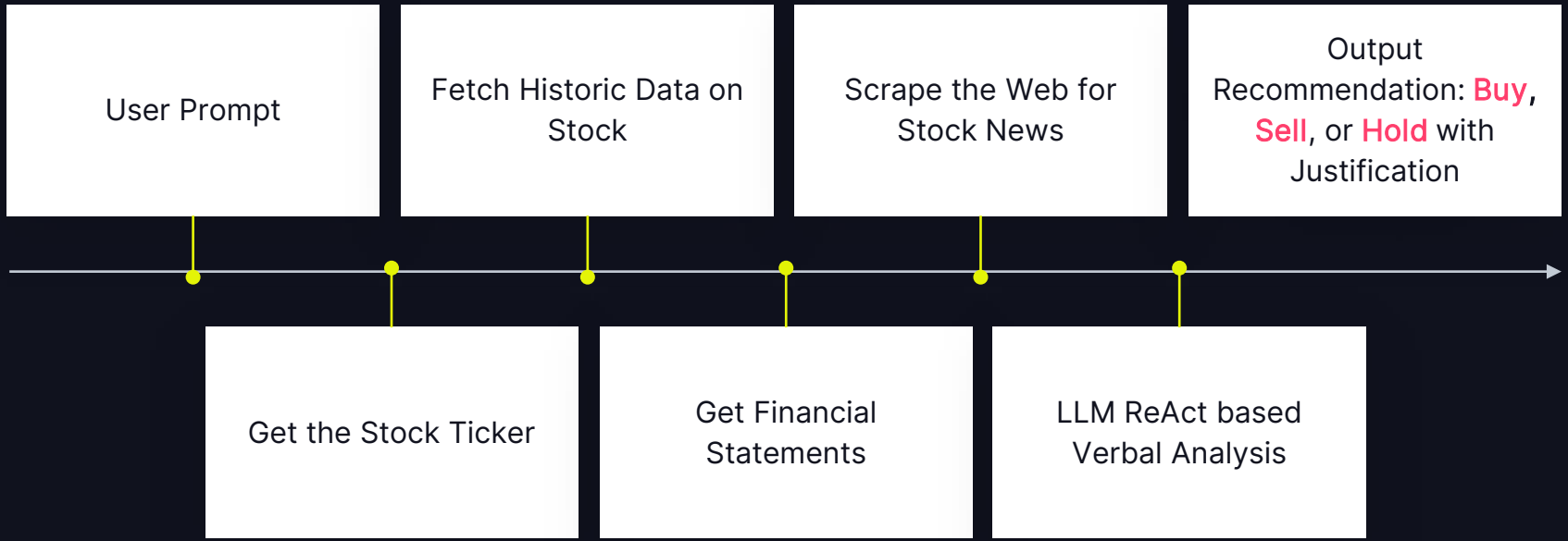
GENERATIVE AI BASED STOCK RECOMMENDATION SYSTEM

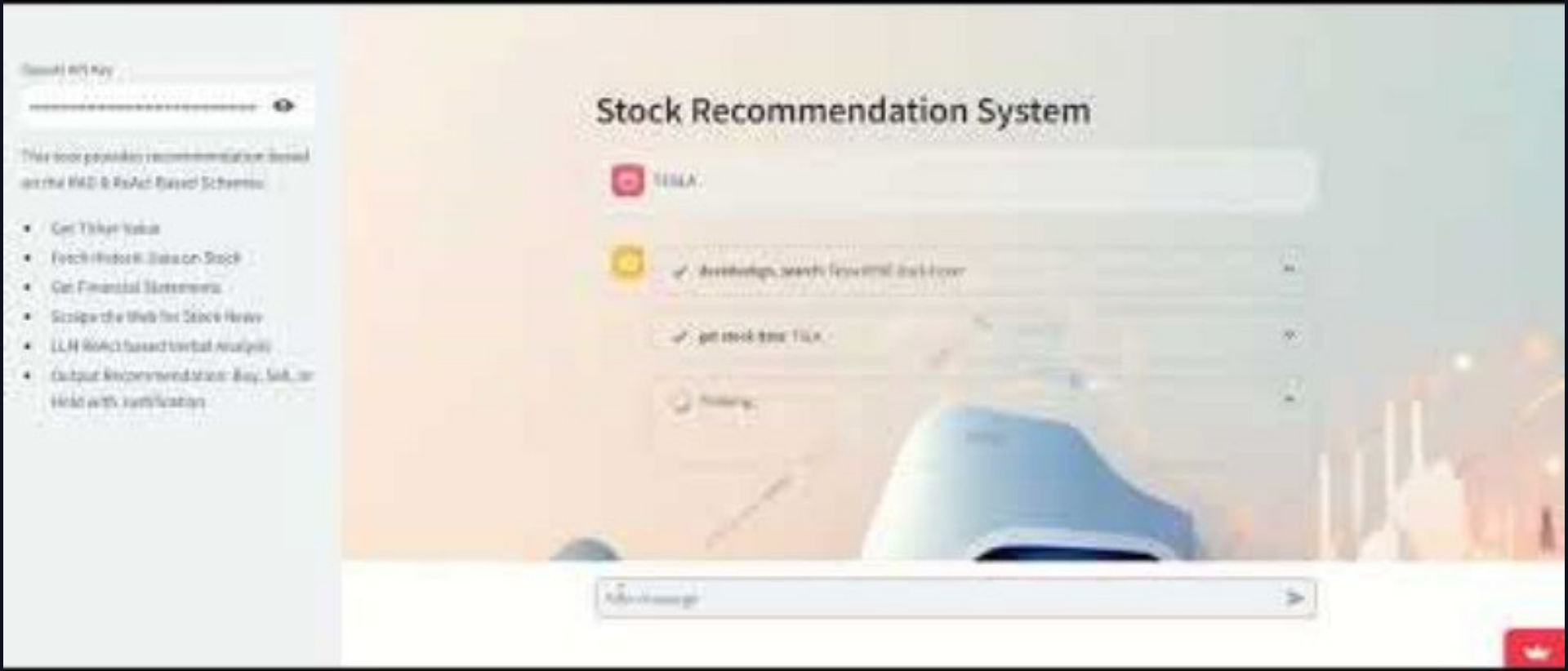
VISION

Generative AI Based Stock Recommendation System

- To utilize Generative AI for Stock Investment Decisions
- Developed a Recommendation System Constituting Multiple Modules Chained Together by Langchain
- RAG & ReAct Based Recommendations to eliminate hallucinations and provide quality outputs
- UI/UX was developed on Streamlit

WORKFLOW




Stock Recommendation System
TESLA

CONCLUSION

- There are multiple AI-Powered Techniques for Stock Portfolio Management, Both Predictive & Generative.
- Combination of **Predictive** & **Generative AI** models can be used to leverage strengths of both in the Autonomous Stock Portfolio Management System
- **Databricks** offers a robust suite of tools and capabilities e.g., **ML Runtime & MLflow** to empower the development and scaling of such **AI products.**

Q/A

Follow Me On:


linkedin.com/in/ayesha-nasim-b31819b5/


github.com/anasim1


medium.com/@ayeshanasim6914