CS5588
MarketPulse Project

Team members:
Odai Athamneh
Semir Hot
Devin Cline
Nami Lkhagvadorj

2024 • 09 • 23

Overview

- 🔄 **Project Recap**
- 📰 **Recent Developments**
- 👟 **Next Steps**
- 🎃 **Q&A**

🔄 Project Recap

- **Problem Statement**
  - The stock market is a complex environment and making the correct choice and decision is near impossible on a consistent basis.
  - Our product, MarketPulse, gives users a projected stock price based on market conditions and data surrounding a stock.

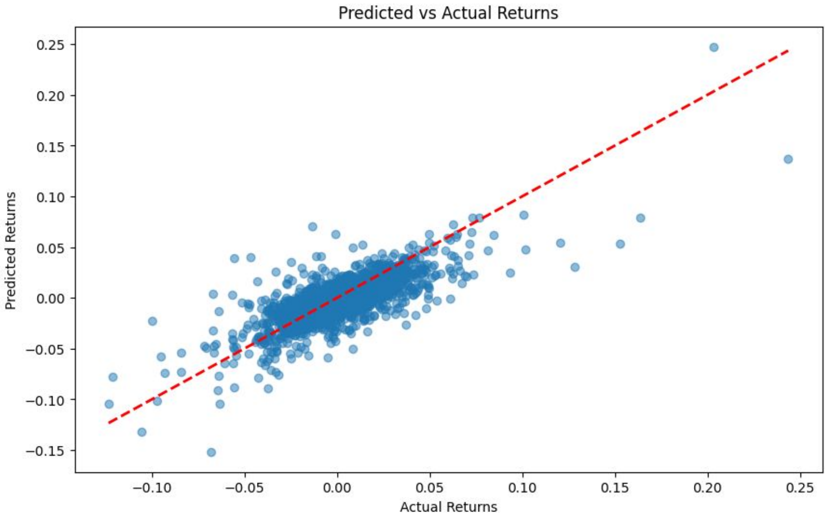
- **Our Proposed Solution**
  - Our team has experimented with graph neural networks and temporal graph neural networks.
  - We are prioritizing accuracy and precision with this model in order to have an effective product that can give the user a profitable tip.

📰 Recent Developments

- Our team has been hard at work developing a great base for data and understanding what data we will need to make an accuracy prediction
- Utilizing the yfinance API we were able to pull in data over a list of starting tickets that will be expanded later after our foundation is established. Deploying FinnHub and newsAPI, we could get the real time news dataset
- We have developed 2 different approaches as mentioned previously that have shown us promising results just based on initial testing (DEMO)
- Moreover the team has begun testing and planning how we will portray these predictions to a UI

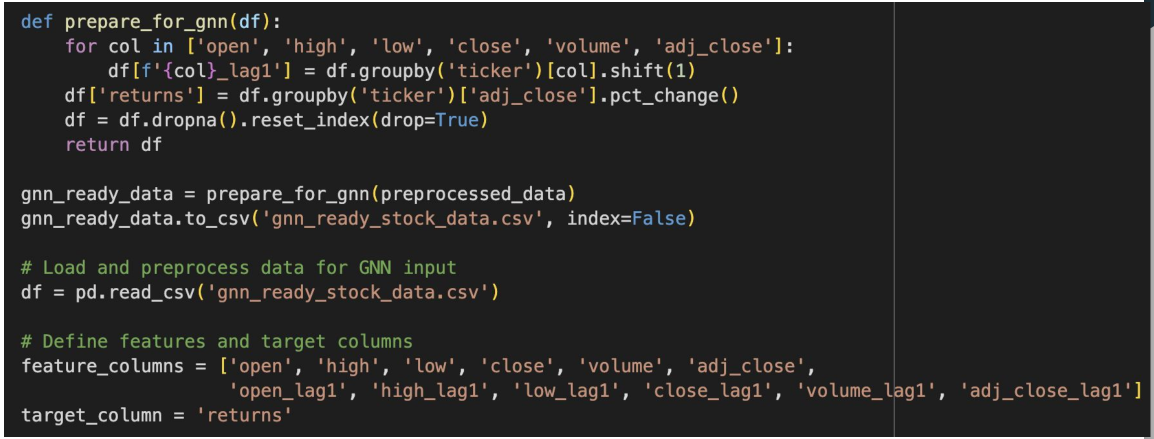
Model

- We first started out with a simple GNN that would take data from the yfinance API.
- With the first run though it we found that it was able to give us a predicted price but the accuracy was just not up to our expectations
- We shifted directions to a temporal GNN and found the results to be much better then previous experimentation



What's the next model?

- This was just a very basic build of the model that took in very limited information from a stock
- There is a whole world of information and features that can be added to this model from the yfinance api
- Moreover we still have yet to incorporate the sentiment data from either social media or the news api
- Moving forward we will be doing a good deal of feature exploration and engineering

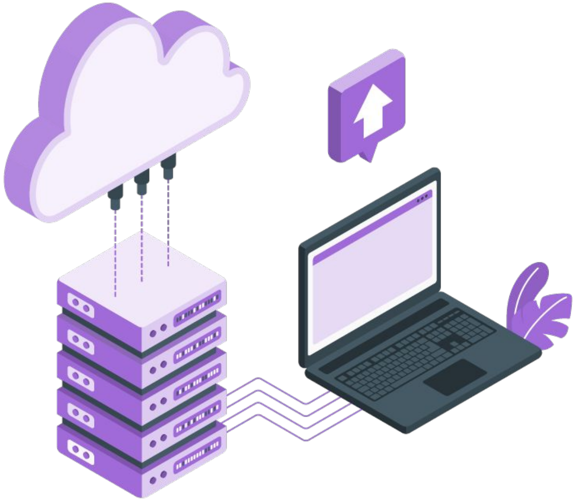


```python
def prepare_for_gnn(df):
    for col in ['open', 'high', 'low', 'close', 'volume', 'adj_close']:
        df[f'{col}_lag1'] = df.groupby('ticker')[col].shift(1)
    df['returns'] = df.groupby('ticker')['adj_close'].pct_change()
    df = df.dropna().reset_index(drop=True)
    return df

gnn_ready_data = prepare_for_gnn(preprocessed_data)
gnn_ready_data.to_csv('gnn_ready_stock_data.csv', index=False)

# Load and preprocess data for GNN input
df = pd.read_csv('gnn_ready_stock_data.csv')

# Define features and target columns
feature_columns = ['open', 'high', 'low', 'close', 'volume', 'adj_close',
                   'open_lag1', 'high_lag1', 'low_lag1', 'close_lag1', 'volume_lag1', 'adj_close_lag1']
target_column = 'returns'
```

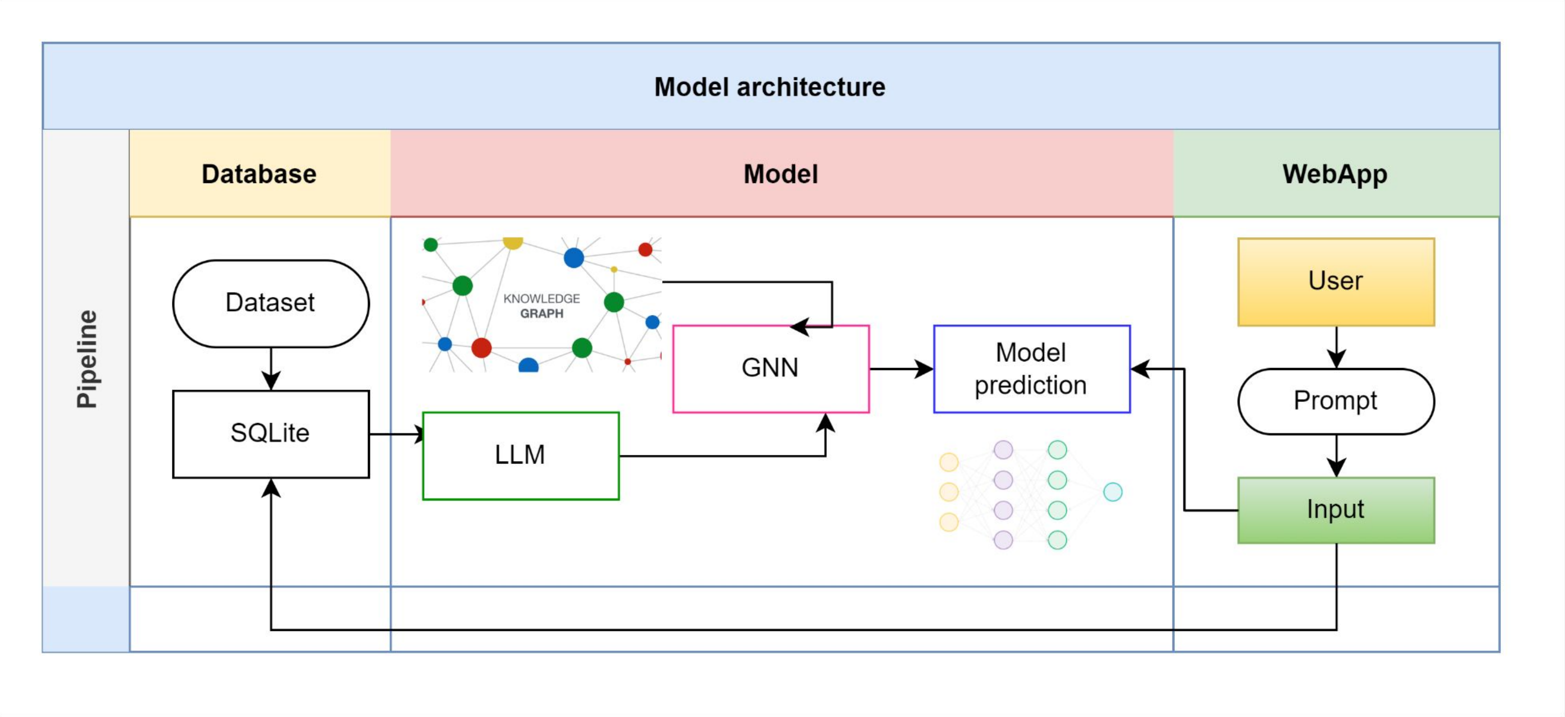
Dataset

News dataset - NewsAPI, Finnhub API, yfinance API

- As touched in in the previous slide we have yet to fully incorporate news or social media data to our model
- We have a dataset that we have begun experimenting with and are exploring the use of NewsAPI and FinnHub API
  - The limitation for NewsAPI is it has articles from the last 30 days in free tier while FinnHub is available for 1 year dataset for free
- This text data we will then take and process with some forum of a LLM and will add that sentiment score onto our DF for our temporal GNN to further tighten that gap as seen previously



Model Architecture



👟 Next Steps

- While we have a great base currently there is still a ton of work to be done in order to have a product that we are proud to show
- We will have to
  - Develop a front end that will be simple for the user to understand the data and easy to navigate
  - Find a method to connect our Database to our front end and have the data readily available and current for each run of the temporal GNN
  - Bring in the text data as previously mentioned and pass it though an LLM

Any questions?