6. MergeDataset

September 6, 2023

[ ]:
```
import pandas as pd
import os
```

[ ]:
```
# Load CSV Data

# Stocks :- AAPL, MSFT, AMZN, NVDA, TSLA, GOOGL
# Sector Indices :- SSINFT (^SP500-45)

ticker = "GOOGL"
# indirectory = "PreProcessedArticles"
indirectory = "PreProcessedContextArticles"

if indirectory == "PreProcessedContextArticles":
  outdirectory = "MergedContextDataset"
else:
  outdirectory = "MergedDataset"

# Load the news artcile file
df = pd.read_csv(f"{indirectory}/{ticker}_news_data.csv")
```

[ ]:
```
# 6. Aggregate the sentument score on a given day and calculate the overall␣
 ↪sentiment by taking each days positive and negative score sum and dividing␣
 ↪by total number of articels on that day
# Fill NaN values in the Summary column
df['Summary'].fillna("", inplace=True)

# Convert all values in 'Headline' and 'Summary' to strings
df['Headline'] = df['Headline'].astype(str)
df['Summary'] = df['Summary'].astype(str)

aggregations = {
    'Headline': ' '.join,
    'Summary': ' '.join,
}

# Group by Date and aggregate
agg_df = df.groupby('Date').agg(aggregations).reset_index()
```

[ ]:
```
# Convert the 'Date' column to datetime dtype (if it's not already)
agg_df['Date'] = pd.to_datetime(agg_df['Date'], format='%Y-%m-%d')
```

[ ]:
```
# 6. Load stock market data with price trend
stock_df = pd.read_excel(f"PreProcessedStocks/{ticker}_stock_data.xlsx",␣
↪engine='openpyxl')
```

[ ]:
```
# Convert the 'Date' column to datetime dtype
stock_df['Date'] = pd.to_datetime(stock_df['Date'], format='%d/%m/%Y')
```

[ ]:
```
# 7. Compare the the sentiment value to the following days price trend and get␣
↪the accuracy
merged_df = pd.merge(agg_df,stock_df,  on="Date", how='inner')
```

[ ]:
```
# Use next day price trend to check the effect of news sentiment
merged_df['next_day_price_trend'] = merged_df['price_trend'].shift(-1)
```

[ ]:
```
# Remove days with nuetral value for sentiment_label to simulate not trading on␣
↪those days since no clear directional sentiment was found.
merged_df = merged_df[~merged_df['next_day_price_trend'].isin(['neutral',␣
↪'None'])]

# Drop all rows without a "price_trend" value (removing non trading days)
merged_df = merged_df.dropna(subset=["price_trend", "next_day_price_trend"])
```

[ ]:
```
# Convert sentiments to binary
merged_df['price_trend'] = merged_df['price_trend'].replace({'positive': 1,␣
↪'negative': 0})
merged_df['next_day_price_trend'] = merged_df['next_day_price_trend'].
↪replace({'positive': 1, 'negative': 0})
```

[ ]:
```
merged_df
```

[ ]:
```
# 5. Output Sentiment Results with stock price trend
merged_df.to_csv(f"{outdirectory}/{ticker}_agg_news_stock_trend_output.csv",␣
↪index=False)z
```

[ ]:
```

```