finbert_min_example

December 17, 2023

[ ]:

```python
import torch

from tqdm import tqdm

import pandas as pd
import numpy as np
from transformers import AutoTokenizer, AutoModelForSequenceClassification

import os
os.environ["TRANSFORMER_CACHE"] = "./.crypto_bot/"
```

```
c:\Users\tomsr\Documents\School\aidi\student
projects\crypto_bot\.crypto_bot\Lib\site-packages\tqdm\auto.py:21: TqdmWarning:
IProgress not found. Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

[ ]:

```python
device = "cuda:0" if torch.cuda.is_available() else "cpu"
tokenizer = AutoTokenizer.from_pretrained("ProsusAI/finbert")
model = AutoModelForSequenceClassification.from_pretrained("ProsusAI/finbert").
  ↪to(
    device
)

stock_data_df = pd.read_csv("./data/stock_data.csv")

stock_data_df.head()

X = stock_data_df["Text"]
y = stock_data_df["Sentiment"]
```

[ ]:

```python
y_test = np.array([])

for text in tqdm(X, desc="tokenizing"):
    token = tokenizer(text, return_tensors="pt", truncation=True).to(device)
    y_test = np.append(y_test, {
        "input_ids": token["input_ids"],
        "token_type_ids": token["token_type_ids"],
        "attention_mask": token["attention_mask"]
    })
```

```
tokenizing: 100%|      | 5791/5791 [00:01<00:00, 2968.51it/s]
```

```
[ ]: y_pred = np.array([])

     with torch.no_grad(), tqdm(total=len(y_test), desc="Predicting") as pbar:
         for index, token in enumerate(y_test):
             output = model(**token)
             prediction = torch.argmax(output.logits, dim=1).item()
             y_pred = np.append(y_pred, prediction)
             pbar.update(1)
```

```
Predicting: 100%|      | 5791/5791 [01:15<00:00, 76.40it/s]
```

```
[ ]: if "Predictions" in stock_data_df:
         stock_data_df.drop("Predictions", axis=1)
     stock_data_df["Predictions"] = y_pred
```

```
[ ]: stock_data_df.shape
```

```
[ ]: (5791, 3)
```

```
[ ]: stock_data_df = stock_data_df[stock_data_df["Predictions"] != 0]
```

```
[ ]: stock_data_df.loc[:, "Predictions"] = np.where(stock_data_df["Predictions"] >=␣
      ↪2, stock_data_df["Sentiment"], -1)
```

```
[ ]: stock_data_df.head(10)
```

```
[ ]:                                                 Text  Sentiment  Predictions
     0  Kickers on my watchlist XIDE TIT SOQ PNK CPW B…          1          1.0
     1  user: AAP MOVIE. 55% return for the FEA/GEED i…          1          1.0
     2  user I'd be afraid to short AMZN - they are lo…          1          1.0
     3                                  MNTA Over 12.00          1          1.0
     4                                   OI  Over 21.37          1          1.0
     5                                  PGNX  Over 3.04          1          1.0
     6  AAP - user if so then the current downtrend wi…         -1         -1.0
     7  Monday's relative weakness. NYX WIN TIE TAP IC…         -1         -1.0
     8  GOOG - ower trend line channel test & volume s…          1          1.0
     9              AAP will watch tomorrow for ONG entry.          1          1.0
```

```
[ ]: # Calculate accuracy
     correct_predictions = stock_data_df["Sentiment"] == stock_data_df["Predictions"]
     accuracy = correct_predictions.sum() / len(correct_predictions)

     print(f'Accuracy: {accuracy * 100:.2f}%')
```

```
Accuracy: 93.67%
```