Virginia Tech

Bradley Department of Electrical and Computer Engineering

ECE 5984 Data Engineering Project
Fall 2024

Assignment 3
Machine Learning Pipeline

Please note the following:

- Solutions must be clear and presented in the order assigned. Solutions must show the work needed, as appropriate, to derive your answers.
- Data being processed in both the lab and homework should be in the correct format and location and any other deliverables should also be completed as mentioned in the modules.
- For the Homework, the submission process requires that all code files should be uploaded to Canvas before the deadline.
- Submit your Homework using the respective area of the class website by 11:55 p.m. on the due date.
- When a PDF file must be submitted, include at the top of the first page: your name (as recorded by the university), email address, and the assignment name (e.g., "**ECE 5984, Homework/Project 1**"). Submit a single file unless an additional file is explicitly requested.

Lab 3

3.1 Perform feature extraction on the cleaned data from Lab 2.2

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data. In our example, we will build towards making a prediction model based on the LSTM architecture.

The first step is to import the libraries required to perform feature extraction on our stock data. In our case we will be using the scikit-learn library. In order to use the libraries, install scikit-learn on both your local machine using PyCharm and onto your docker container on the cloud. On your local machine the procedure to add scikit-learn is the same as talked about in Lab 1. To install the scikit-learn library onto you docker container, spin up your container and run the following command.

```
pip install -U scikit-learn
```

The following steps are then done to perform feature extraction. These steps are executed in code in the featureExtraction.py script. You should follow along with the code to understand the reasoning behind some of the operations performed.

1. The cleaned data from lab 2.2 is fetched from the S3 bucket (data warehouse) and loaded as dataframes.
2. The output column is assigned to the target variable. It is the 'adj close' value of each company in question. Furthermore, we pick the features that serve as the independent variables to the target variable (dependent variable). We choose five characteristics to account for training purposes:
    a. 'Close'
    b. 'High'
    c. 'Low'
    d. 'Open'
    e. 'Volume'

3. To decrease the computational cost of the data in the table, we will scale the stock values to values between 0 and 1. As a result, all the data in large numbers is reduced, and therefore memory consumption is decreased. Also, because the data is not spread out in huge values, we can achieve greater precision by scaling down. To perform this, we will be using the MinMaxScaler class of the scikit-learn library.
4. Next, the entire dataset is divided into training and test sets before feeding it into any training model. The Machine Learning LSTM model will be trained on the data in the training set and tested for accuracy and backpropagation on the test set. The scikit-learn library's TimeSeriesSplit class will be used for this. We set the number of splits to 10, indicating that 10% of the data will be used as the test set and 90% of the data will be used to train the LSTM model. The advantage of utilizing this Time Series split is that the split time series data samples are examined at regular time intervals.
5. The final X_train, X_test, y_train and y_test features are then pushed to the cloud infrastructure.

Steps to run feature extraction on the pipeline:

1. Save the dag.py, batch_ingest.py ,transform.py and featureExtraction.py scripts to your local machine
2. Get into your docker container
3. Navigate to the airflow/dags folder
4. Create a new dag.py and copy all the code from your local machine dag.py onto it using sudo nano dag.py and copy from your local machine clipboard using right click.
5. Save by pressing ctrl+x and then hit Enter
6. Similarly create a batch_ingest.py, featureExtraction.py and transform.py file and copy code from local machine batch_ingest.py and transform.py onto it using the same steps
7. Access your airflow GUI
8. You should see the same **batch_ingest_dag**. Click it and go to the graph. On the right side click the play button and press trigger DAG
9. This triggers the dag we just uploaded which intern runs the ingest_data function from batch_ingest.py followed by the transform_data() function from transform.py and then ends with the feature extraction and you are able to see the whole process
10. After the DAG finishes running you should be able to navigate to your S3 bucket directory and check to see the data now there in a pickle file format

3.2 Build and train a machine learning model on the stock market data set being used throughout the previous labs

In this Module we will be building and training an LSTM prediction model that can predict the 'adj close' price of the stocks data we feed it. The point of the lab is not to provide a state-of-the-art method to build such a model but rather to give an idea on how to implement a pipeline that can train almost any kind of machine learning model.

After performing feature selection and extraction, the extracted data is then reshaped and fed into the LSTM model according to the tensor required. Once the training and test sets are retrieved from our data warehouse, we will input the data into the LSTM model. Before we can do that, we must transform the training and test set data into a format that the LSTM model can interpret. As the LSTM needs that the data to be provided in the 3D form, we first transform the training and test data to NumPy arrays and then restructure them to match the format.

Next, we construct the LSTM Model. In this step, we'll build a Sequential Keras model with one LSTM layer. The LSTM layer has 32 units and is followed by one Dense Layer of one neuron. We compile the model using Adam Optimizer and the Mean Squared Error as the loss function. For an LSTM model, this is the most preferred combination.

The last step is to use the fit function to train the LSTM model created on the training data for each company for 25 epochs with a batch size of 8. The trained model is then saved and pushed to our s3 bucket in a .h5 file format.

In order to perform the lab, follow these steps:

1. Open a new PyCharm project onto your local machine
2. Copy all the provided .py scripts onto your new project folder
3. Read through the code along with the comments of featureExtraction.py and build_train_model.py and complete the code where needed.
4. Get into your docker container
5. Navigate to the airflow/dags folder
6. Uncomment the commented parts of your dag.py in your local machine (including the model_etl in the last line)
7. Create a new dag.py and copy all the code from your local machine dag.py onto it using sudo nano dag.py and copy from your local machine clipboard using right click.
8. Save by pressing ctrl+x and then hit Enter
9. Similarly add all the other .py scripts to the same location
10. Access your airflow GUI
11. You should see the same **batch_ingest_dag**. Click it and go to graph. On the right side click the play button and press trigger DAG
12. This triggers the dag we just uploaded which intern runs all the coded py scripts in a pacific order and you are able to see the whole process
13. After the DAG finishes running you should be able to navigate to your S3 bucket directory and check to see the data now there in a pickle file format along with the trained models in .h5 format
14. You can download the test data set and the trained model from the S3 bucket into your local machine and test your model on the test data set (Optional)

Deliverables

3.1 Perform similar feature extraction as in Lab 3.1

3.2 Perform a similar machine learning training task as in Lab 3.2

1. The dag.py file, batch_ingest.py, featureExytraction.py, build_train_model.py and the transform.py file for the lab should be uploaded to Canvas.

2. Data from the lab should be in the appropriate S3 buckets (data lake and data warehouse). Namely the following files:
   a. data.pkl (batch ingest data)
   b. clean_aapl.pkl, clean_amzn.pkl, clean_googl.pkl (transformed data)
   c. X_train_aapl.pkl, X_test_aapl.pkl, y_train_aapl.pkl, y_test_aapl.pkl (feature selection data for Apple)
   d. X_train_amzn.pkl, X_test_amzn.pkl, y_train_amzn.pkl, y_test_amzn.pkl (feature selection data for Amazon)
   e. X_train_googl.pkl, X_test_ googl.pkl, y_train_ googl.pkl, y_test_ googl.pkl (feature selection data for Google)
3. Saved model should be in .h5 format in the appropriate location within the S3 bucket. Namely:
   a. lstm_aapl.h5, lstm_amzn.h5, lstm_googl.h5