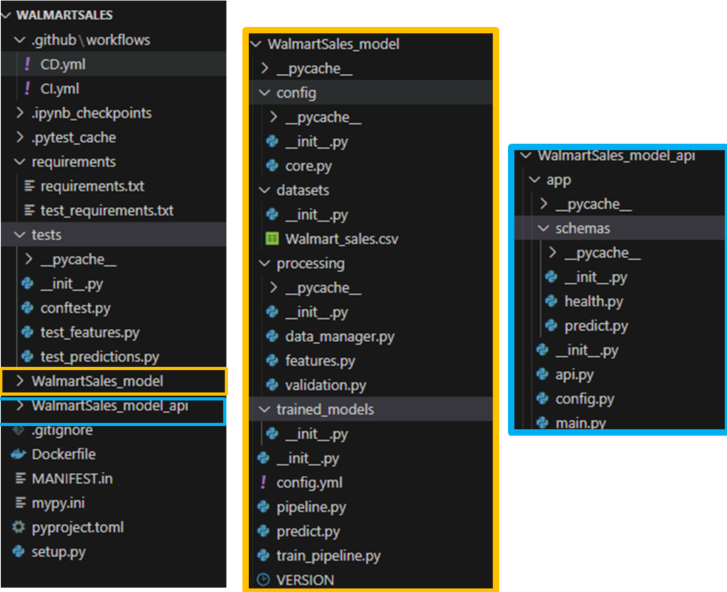
**Use case**: To predict the weekly sales of a given Walmart Store

**Input**: Below listed are the columns that is part of the dataset

1. Store Number – A unique ID for 45 odd stores
2. Date – Start date of weekly sales
3. Weekly Sales – This is the target column
4. Holiday Flag – Categorical column indicating whether it was a holiday or not
5. Temperature – Air Temperature in the region
6. Fuel Cost – Price of fuel in the region
7. CPI – Consumer Price Index
8. Unemployment – Rate of Unemployment

**Steps followed to build the application:**

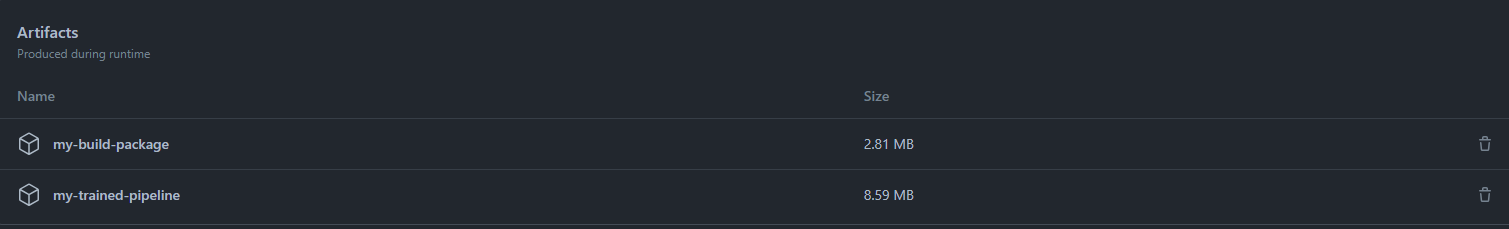
1. I started with exploratory data analysis to get some insights about the data, the notebook has been committed in the github repo
2. Functions were coded in the jupyter notebook for pre-processing, data transformation, training and prediction
   1. For ex: preprocess\_dataframe function processes the date and extracts the quarter
   2. Likewise scale\_columns function scales features namely Temperature, Fuel Price, CPI and Unemployment using the Scikit learn implementation of StandardScaler
   3. Train\_random\_forest\_regressor for training
      1. Two models were tried mainly, linear regression and random forest regressor out of which Random Forest Regressor prove to be superior interms of predicting the weekly sales at an RMSE of 147837.66 compared to linear regression which gave an RMSE of 522052.41.
      2. Hence Random forest regressor was chosen to be champion model for this use case
3. Next steps were about building pipelines and packaging.
   1. Here is the link to the repository, it is intended to be a public repo so that the mentors will be able to access <https://github.com/itzmemithul/WalmartSales> and take a look through the runs those were successful and failed ones☺
   2. This is an implementation of Fast API using EC2 instance with CI CD pipelines integrated in it hence this covers Mini Project 1 as well as Mini Project 2 for the same problem statement.
4. Listed below are some of the important snippets:
5. Structure of the directory in Github



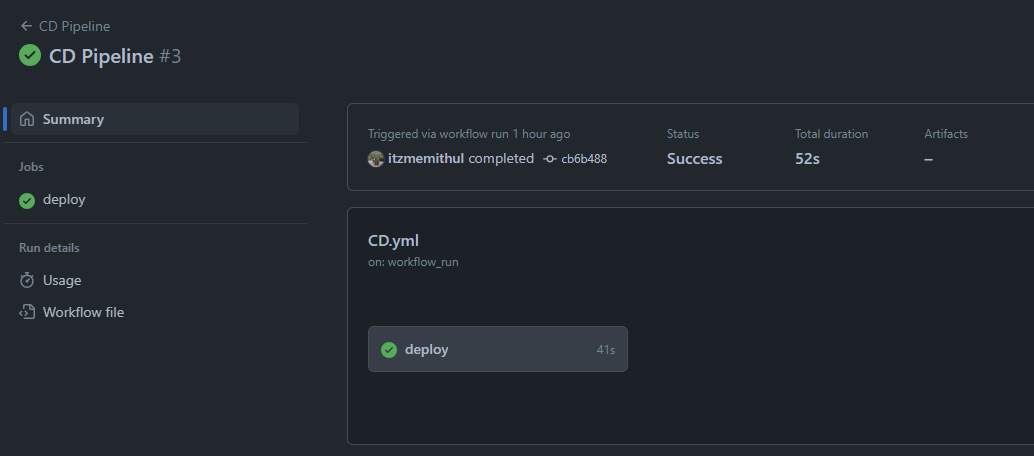
1. Completion of continuous integration workflow with the help of github actions



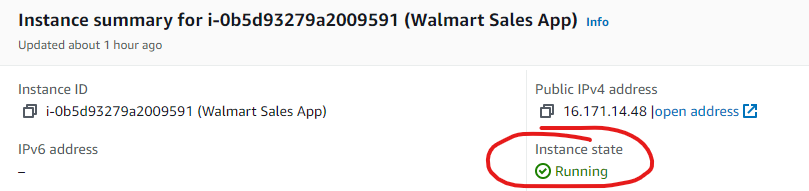
1. Here are the artifacts generated from the continuous integration pipeline.



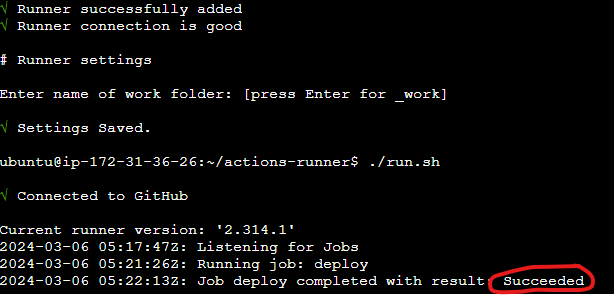
1. Completion of continuous deployment pipeline workflow with the help of github actions.



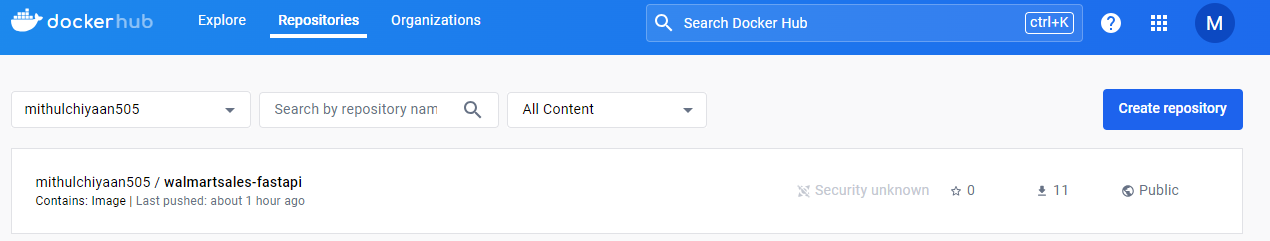
1. Here is the snippet of the EC2 instance running.



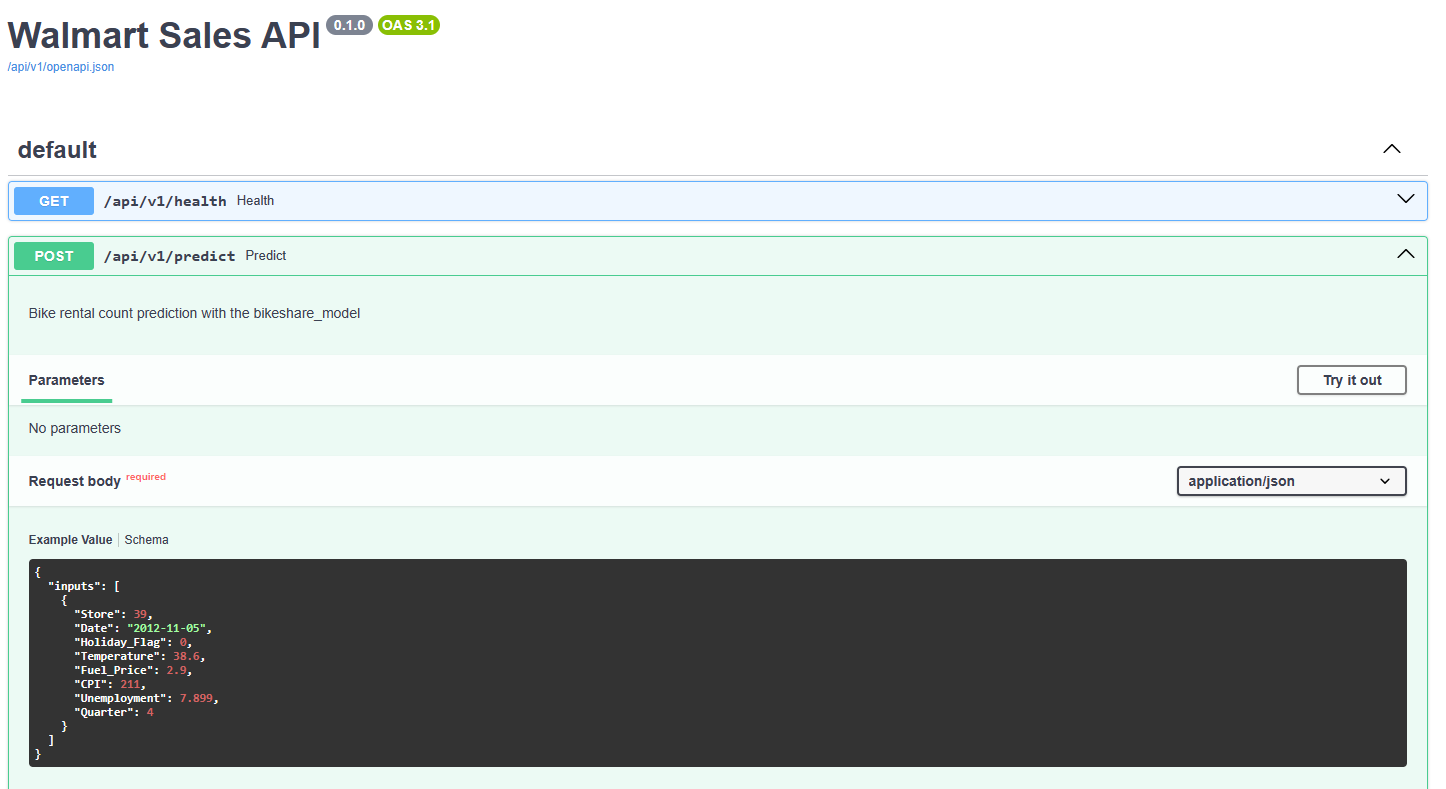
1. The docker was installed on the EC2 instance and it was added as self-hosted runner to the github.



1. The image was pushed as a part of CI, CD pipeline to the docker hub which is captured in the snippet shown below.



1. Finally the application where the sales for store #39 was predicted for 2012-11-05.



1. What we see in the prediction is the actual sales value for the given store on a given day

