Timely Project Report

Problem Statement:

Build a classification model from the sample data provided to predict which time series model can be a best fit to the new csv data that is given as input and predict the output for a particular date given as input from the time series model chosen by our classifier model (Classifier Model should be build based on MAPE value of the time series models).

Data:

The sample dataset given has 4 categories Daily , Hourly , Weekly , Monthly . There are totally 36 datasets available in sample data . Each contains Two columns 'point_timestamp' , 'point_value' . point_timestamp is the timestamp of the data and the point_value is an integer . The data is preprocessed by removing any missing values and scaling it to have a zero mean and unit variance.

Models:

The time series models which were used here are ARIMA , SARIMA , XGboost.

ARIMA:The order of the ARIMA model is denoted as (p, d, q), where p is the order of the autoregressive component, d is the degree of differencing, and q is the order of the moving average component. The ARIMA model can be used to forecast future values of a time series, as well as to analyze the patterns and trends in the data.

SARIMA:

SARIMA, which stands for Seasonal Autoregressive Integrated Moving Average, is a time series forecasting model that extends the ARIMA model by incorporating seasonal components.

The SARIMA model can be written as ARIMA(p, d, q)(P, D, Q)s, where p, d, and q are the non-seasonal components of the ARIMA model, and (P, D, Q)s are the seasonal components.

XGboost:

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that can be used for time series forecasting. XGBoost is a tree-based ensemble method that builds a series of decision trees and combines their predictions to generate a final forecast.

For XGboost model various features are extracted from the given data such as

- Dayofweek
- Month
- Year
- Dayofyear
- Dayofmonth
- Weekofyear

These are the features that are used for training an xgboost model and also for the the training of the sample data lags are also used as feature for the building of xgboost model

Performance measure:

MAPE:MAPE stands for Mean Absolute Percentage Error. It is a commonly used metric for measuring the accuracy of predictions made by forecasting models. MAPE calculates the average absolute percentage difference between the predicted values and the actual values

Classifier Model Building:

We need to build a classifier model to choose the best model for the new input data given . For building the classifier model we need to extract various features from the sample data given . some of the features are

- Stationarity
- ACF
- PACF
- Ymean
- YSTD
- MAXy
- Miny
- MedianY
- Percentile

With these features as X and label should be done by taking the minimum of the MAPE value from all the 3 models and then the label will be assigned . With this we will get a dataframe of multiway classification

For the train data the accuracy was 100% and for the test data the accuracy is 75%

Now with the data available an classification model XGboost is done and save the model as a pickle file

Finding the Best Model:

After the building of a classification model ,with the help of that model our best model for forecasting is predicted by extracting the features from the input data and predict the best model using the pickle file which was already done

Forecasting:

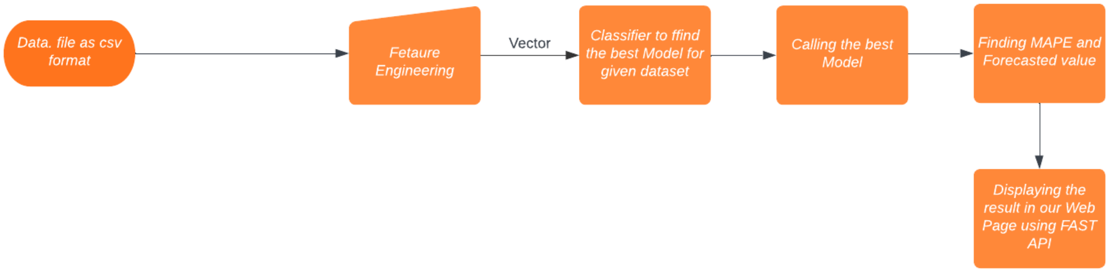
After finding the best model , the input data is trained in the model that was chosen as the best model and the MAPE value is calculated for that data and also the point_value is forecasted for the given particular timestamp

Fast API:

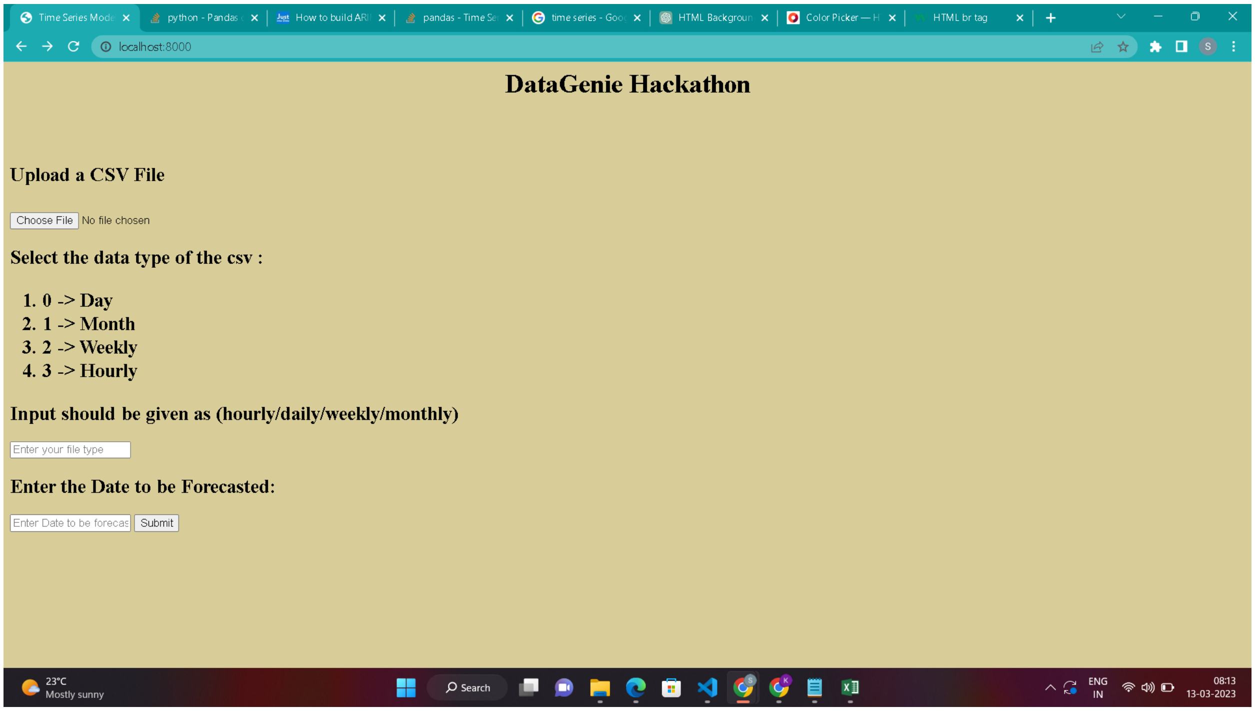
FastAPI is a modern, fast (hence the name) web framework for building APIs with Python. It is built on top of the Starlette framework, which is a lightweight ASGI (Asynchronous Server Gateway Interface) framework, and uses Python 3.6+ type hints to provide fast, efficient, and easy-to-use routing and data validation.

User interface:HTML (Hypertext Markup Language) is used to create the structure and content of web pages. HTML provides a set of tags that are used to define the various elements of a web page, including text, images, links, forms, and other interactive elements. Here is a brief overview of how to create a simple user interface using HTML

Project Workflow:


Data. file as csv format
Fetaure Engineering
Vector
Classifier to ffind the best Model for given dataset
Calling the best Model
Finding MAPE and Forecasted value
Displaying the result in our Web Page using FAST API

HOME PAGE



DataGenie Hackathon

Upload a CSV File

Choose File No file chosen

Select the data type of the csv :

1. 0 -> Day
2. 1 -> Month
3. 2 -> Weekly
4. 3 -> Hourly

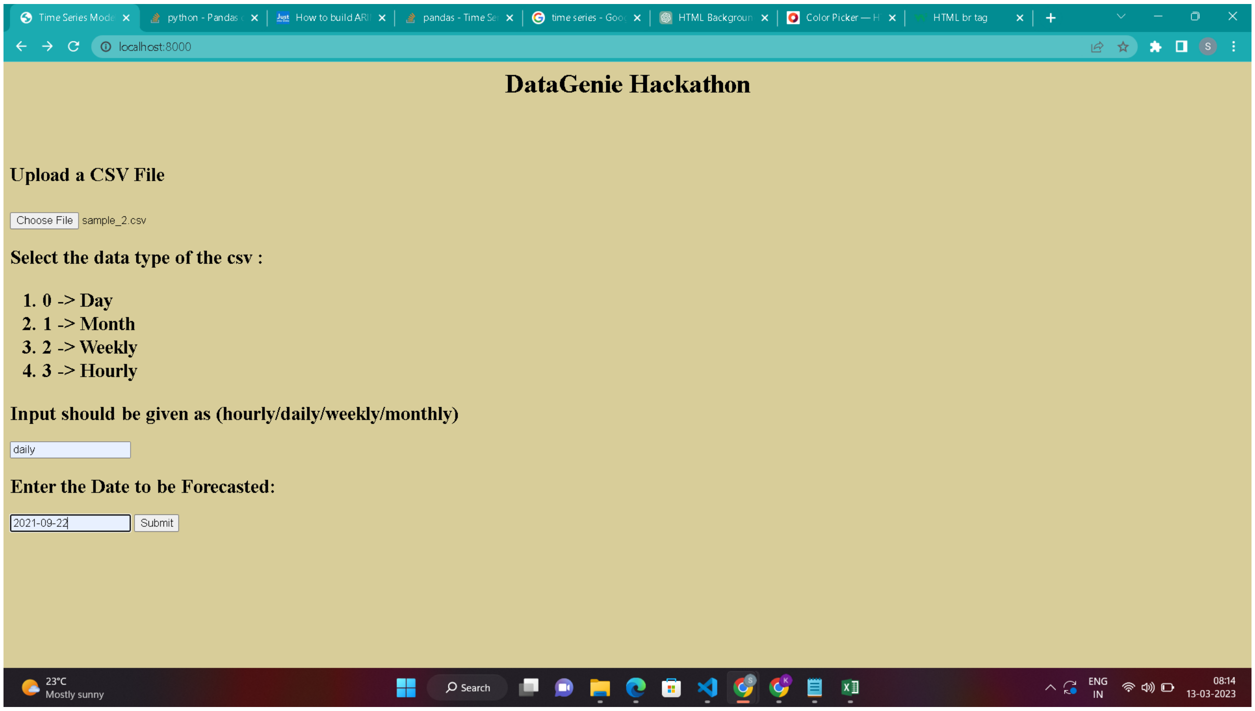
Input should be given as (hourly/daily/weekly/monthly)

Enter your file type

Enter the Date to be Forecasted:

Enter Date to be forecas Submit

Reading a Sample data and input date was given



DataGenie Hackathon

Upload a CSV File

Choose File sample_2.csv

Select the data type of the csv :

1. 0 -> Day
2. 1 -> Month
3. 2 -> Weekly
4. 3 -> Hourly

Input should be given as (hourly/daily/weekly/monthly)

daily

Enter the Date to be Forecasted:

2021-09-22 Submit

Output: Best Model is ARIMA ,its MAPE value and Predicted point value is displayed

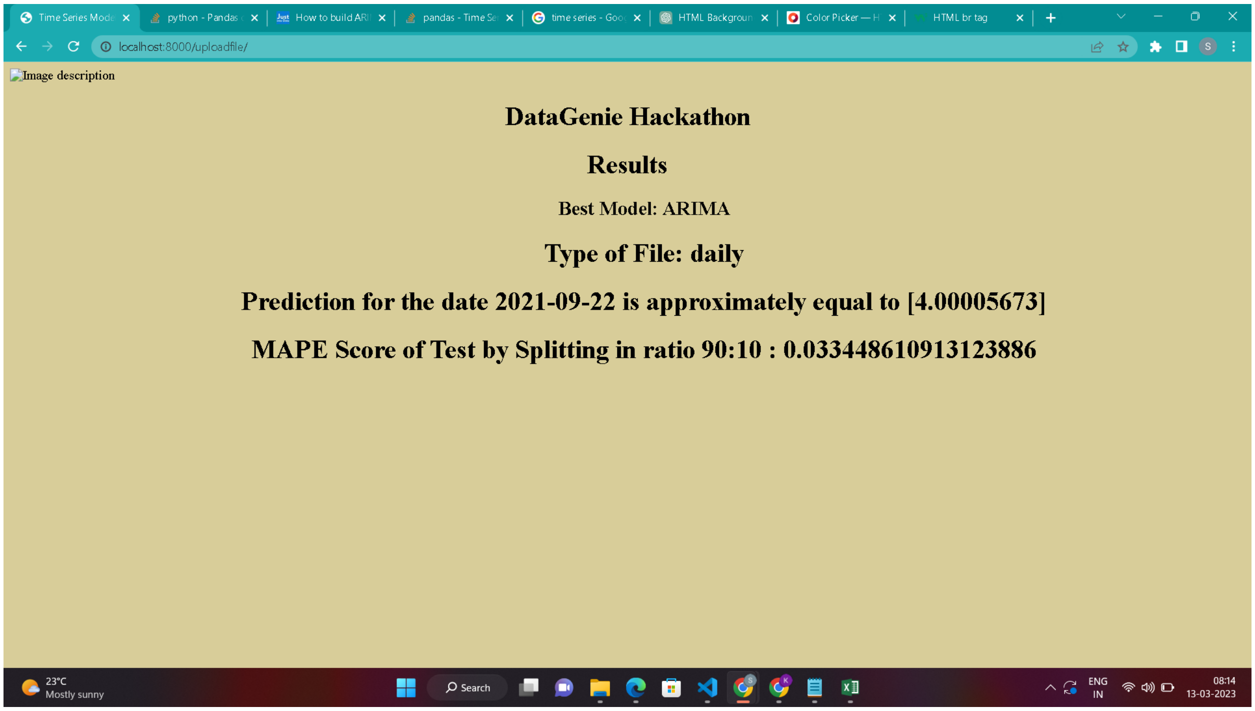


Image description

DataGenie Hackathon

Results

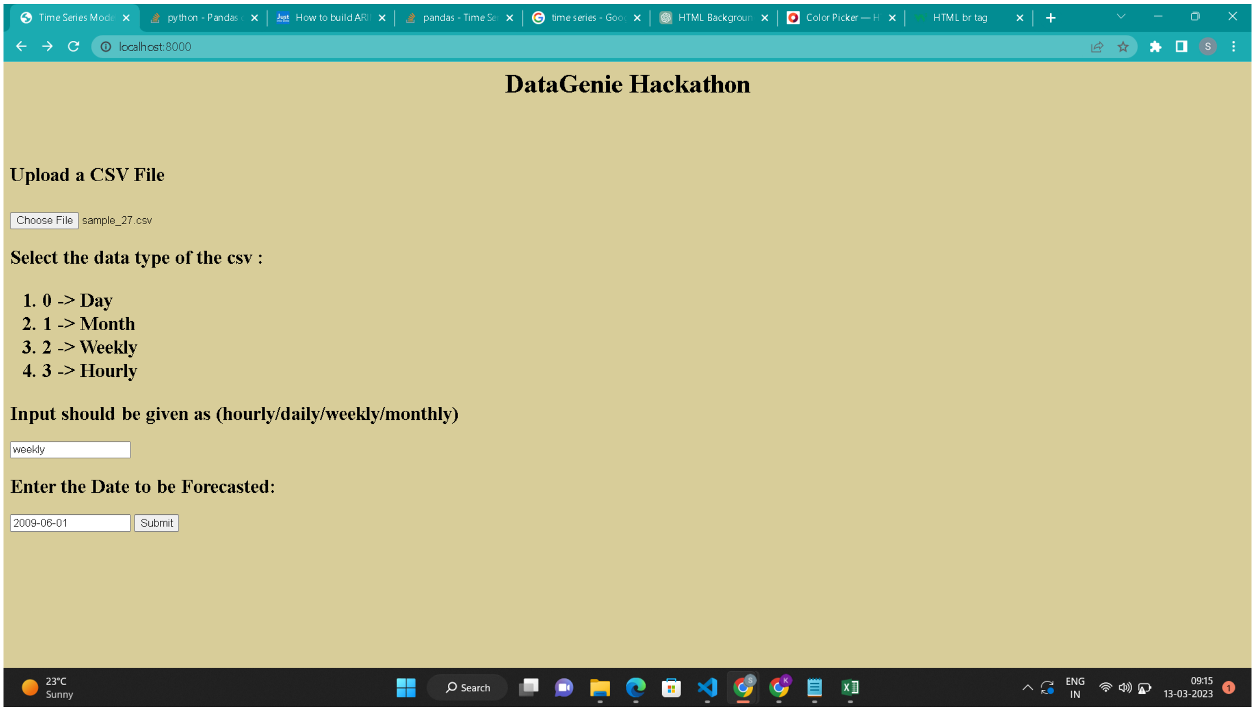
Best Model: ARIMA

Type of File: daily

Prediction for the date 2021-09-22 is approximately equal to [4.00005673]

MAPE Score of Test by Splitting in ratio 90:10 : 0.033448610913123886

Reading a Sample data and input date was given



DataGenie Hackathon

Upload a CSV File

Choose File sample_27.csv

Select the data type of the csv :

1. 0 -> Day
2. 1 -> Month
3. 2 -> Weekly
4. 3 -> Hourly

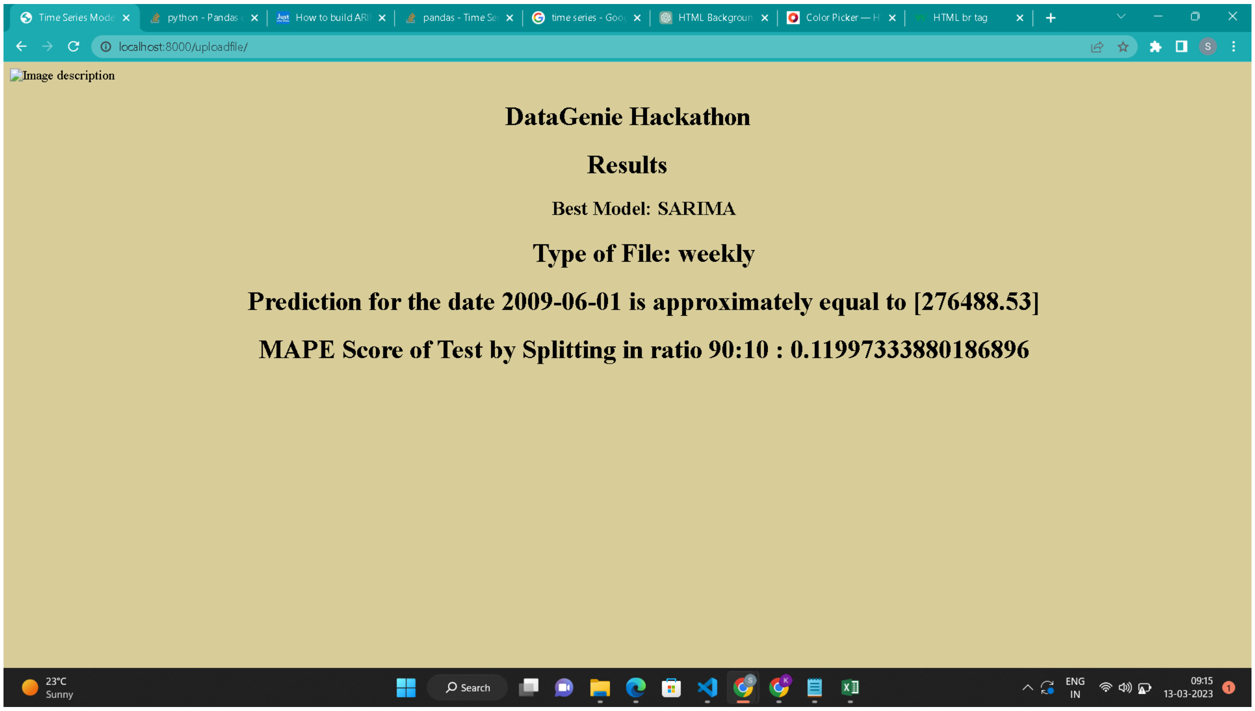
Input should be given as (hourly/daily/weekly/monthly)

weekly

Enter the Date to be Forecasted:

2009-06-01 Submit

Output: Best Model is SARIMA ,its MAPE value and Predicted point value is displayed



DataGenie Hackathon

Results

Best Model: SARIMA

Type of File: weekly

Prediction for the date 2009-06-01 is approximately equal to [276488.53]

MAPE Score of Test by Splitting in ratio 90:10 : 0.11997333880186896

Challenges Faced:

- It is difficult to tune the model since it does not follow any trend , stationary etc and also some of the dataset in sample data is very small so the training is not that efficient .
- I also tried doing prophet and I was able to run that in google colab but I had some installing issue the package in VS code , so I'm not able to add that model
- Since, many of the deployment platforms turned to paid services. I couldn't find any free to use platform to deploy my application.

Checkpoint 1: Completed

Build a classifier model for choosing best model

Checkpoint 2: Completed

Forecasted the new input data by choosing best model

Checkpoint 3 : Completed

Built a Rest Api using Fast API

Checkpoint 4 : Deployment was not done

Basic UI was done using HTML

Future Goals:

1. Create some more features for the classification algorithm to decide the best classifier.
2. Try training my classifier with more data, since now I had only 36 samples to train. It probably led to an overfitted model.
3. Hyper parameter tuning could have been done to further improve the MAPE score and Performance.
4. UI/UX could be developed more interactively.
5. Deployment of the application for easy use of this application by anyone.

Summary:

- The problem statement involves building a classification model to predict which time series model can be a best fit for new input data and forecasting the output for a particular date.
- The sample data provided has 4 categories and 36 datasets with two columns, preprocessed by removing missing values and scaling to have zero mean and unit variance.
- The models used are ARIMA, SARIMA, and XGBoost, and the performance measure used is MAPE.
- Features extracted for classifier model building include stationarity, ACF, PACF, Ymean, YSTD, MAXy, Miny, MedianY, and Percentile. XGBoost is used for multiway classification with 100% accuracy on train data and 75% accuracy on test data.
- The best model is predicted using the classifier model and input data is trained in the chosen model for forecasting. FastAPI and HTML are used for building the user interface