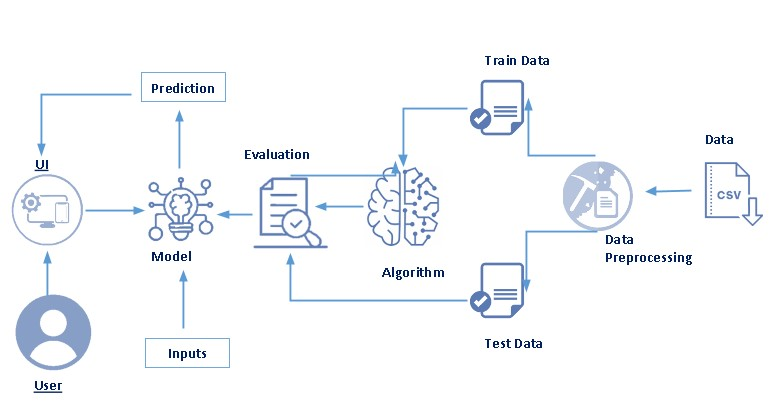
Walmart Store Sales Forecasting

# Project Description

Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends. Here, the company is Walmart. Walmart is a renowned retail corporation that operates a chain of hypermarkets. Walmart has provided a data by combining the data of 45 stores including store information and monthly sales. Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. The data is provided on weekly basis. We have to find the impact of holidays on the sales of the store. The holidays included are Christmas, Thanksgiving, Super Bowl and Labour Day. We will be using algorithms such as ARIMA, Random Forest, and XgBoost. We will train and test the data with these algorithms. Flask integration and IBM deployment will also be done.

**Architecture**



**Pre-requisites**

To complete this project, knowledge of the following software, concepts and packages is required.

* **Anaconda navigator** 
  + Refer the link below to download anaconda navigator
  + Link: <https://www.youtube.com/watch?v=5mDYijMfSzs>
* **Python packages**
  + Open anaconda prompt as administrator
  + Type “pip install numpy” and click enter.
  + Type “pip install pandas” and click enter.
  + Type “pip install scikit-learn” and click enter.
  + Type “pip install matplotlib” and click enter.
  + Type “pip install seaborn” and click enter.
  + Type “pip install pmdarima” and click enter.

**Prior Knowledge**

You must have prior knowledge of following topics to complete this project.

* **ML Concepts**
  + Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
  + Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>
  + Regression and classification
  + Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
  + Xgboost: <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>
  + Evaluation metrics: <https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/>

<https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383>

# Project Objectives

By the end of this project, you will:

* Know fundamental concepts and techniques used for machine learning.
* Gain a broad understanding about data.
* Have knowledge on pre-processing the data/transformation techniques on outlier and some visualization concepts.

# Project Flow

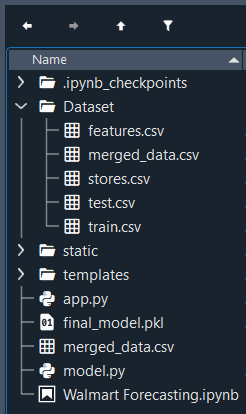
* The data is loaded and cleaning and feature extraction is performed on the data.
* Model is trained using machine learning models.
* Once model is trained, it is tested and evaluation is performed.

To accomplish this, we have to complete all the activities listed below,

* Data collection
  + Collect the dataset or create the dataset
* Visualizing and analyzing data
  + Descriptive analysis
* Data pre-processing
  + Checking for null values
  + Filling null values
  + Splitting data into train and test
* Model building
  + Import the model building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating performance of model
  + Save the model

# Project Structure

Create the Project folder which contains files as shown below



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* Final.model.pkl is our saved model. This model will be used for flask integration.
* Dataset folder contains model csv files – features, train, stores, test and merged\_data .
  + train.csv
    - This is the historical training data, which covers to 2010–02–05 to 2012–11–01. The following fields are present:
    - Store — the store number
    - Dept — the department number
    - Date — the week
    - Weekly\_Sales — sales for the given department in the given store
    - IsHoliday — whether the week is a special holiday week
  + test.csv
    - This file is identical to train.csv, except the weekly sales column is omitted.
    - (**Note:** This dataset will not be used in this project. Rather we will merge all the other datasets and use the merged data file for training and testing both.)
  + features.csv
    - This file contains additional data related to the store, department, and regional activity for the given dates. It contains the following fields:
    - Store — the store number
    - Date — the week
    - Temperature — average temperature in the region
    - Fuel\_Price — cost of fuel in the region
    - MarkDown1–5 — anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
    - CPI — the consumer price index
    - Unemployment — the unemployment rate
    - IsHoliday — whether the week is a special holiday week
    - The holidays considered are:
      * Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13
      * Labor Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13
      * Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13
      * Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

# Data Collection

ML depends heavily on data. It is most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

**Download the dataset**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project, Walmart-dataset.zip is used data which has features.csv, stores.csv, train.csv, and test.csv. This data is downloaded from kaggle.com. Please refer the link given below to download the dataset.

Link: <https://www.kaggle.com/competitions/walmart-recruiting-store-sales-forecasting/data>

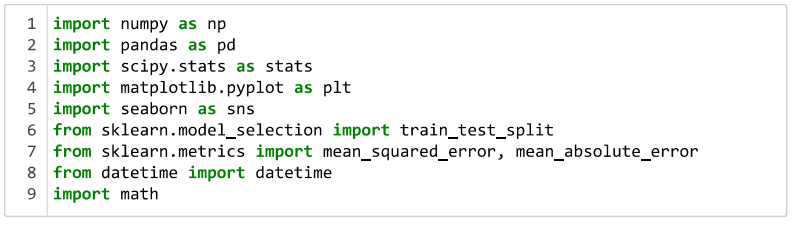
# Visualizing and analysing the data

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

**There are numerous techniques for understanding the data. But here some of them are used. In addition, other and multiple techniques can also be used.**

**Importing the libraries**

**Import the necessary libraries as shown in the image.**



**Read the Dataset**

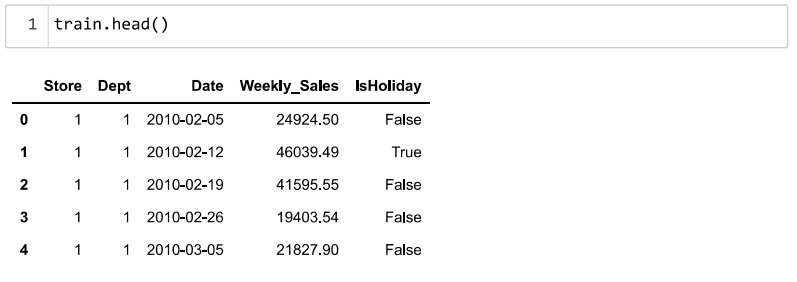
The dataset format might be in .csv, excel files, .txt, .json, etc. So, the dataset can be read with the help of pandas.

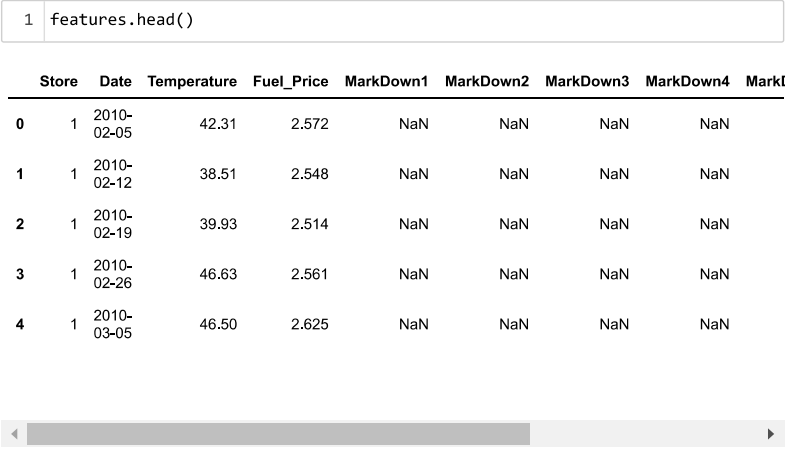
In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of csv file.

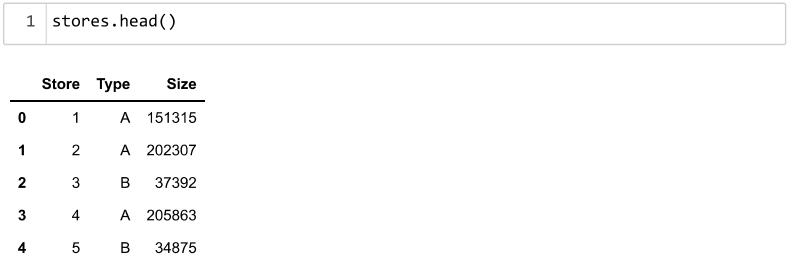
All the datasets are read in the same way.



After reading the datasets, we will be viewing them by using the .head() function which will by default display first 5 rows of the dataset.

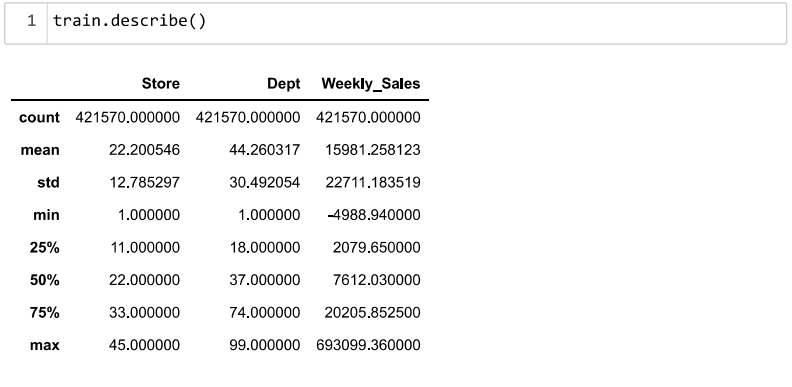


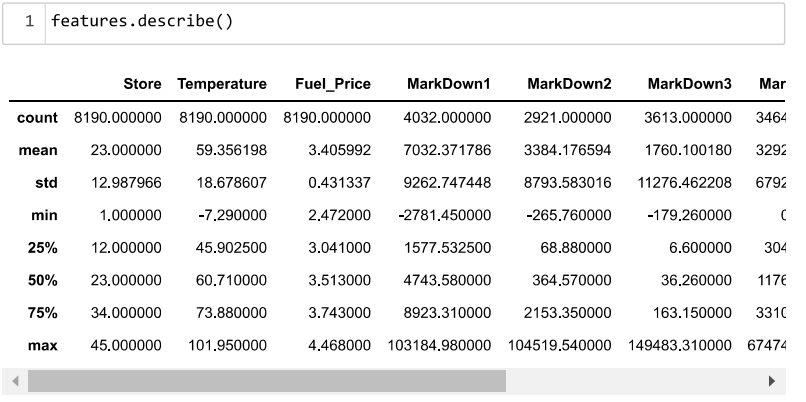


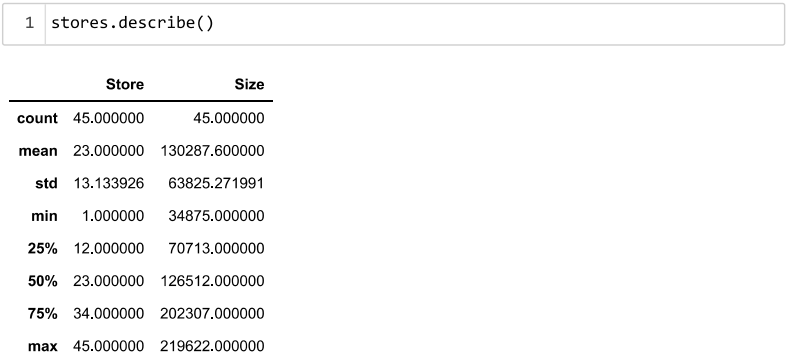


**Descriptive analysis**

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe(). With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.







# Data Pre-processing

As we seen and understood the description of the data, lets pre-process the collected data.

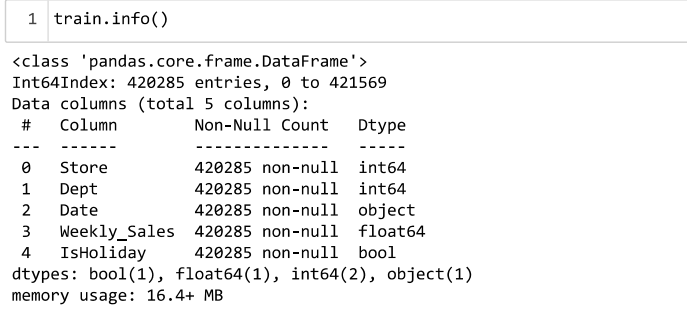
The download data set is not suitable for training the machine learning model as it might have so much of randomness so, the dataset has to be cleaned properly in order to fetch good results. This activity includes the following steps.

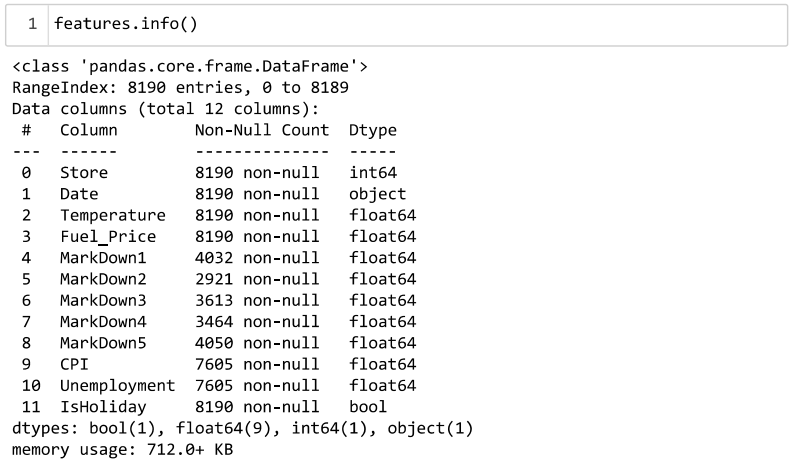
* Handling missing values
* Handling categorical data
* Handling outliers
* Scaling Techniques
* Splitting dataset into training and test set

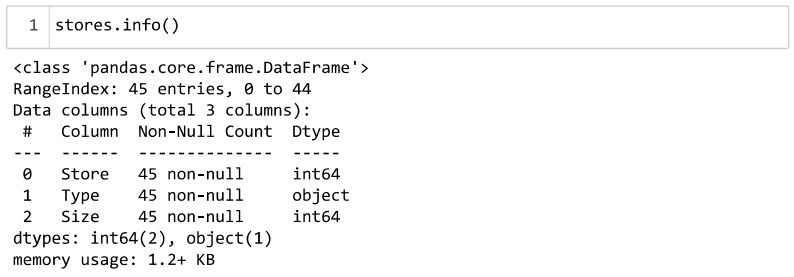
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

**Checking for null values**

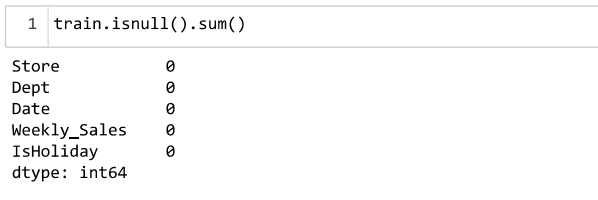
* To find the data type, .info() function is used.

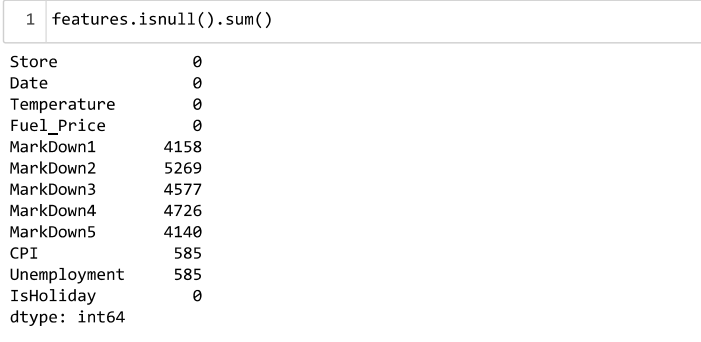


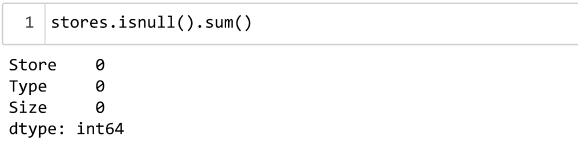




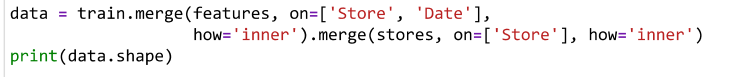
* For checking the null values, .isnull() function is used. To sum those null values .sum() function is used with it.



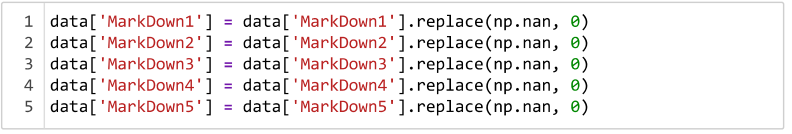




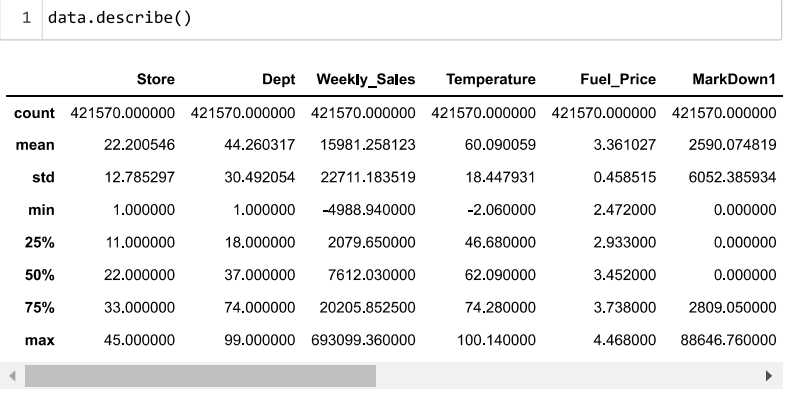
Since the data is divided into separate files, these files will first be merged into a single file and then further analysis will be performed on it.



The missing data is handled by filling them with zeroes by using the .fillna() or the .replace() function.



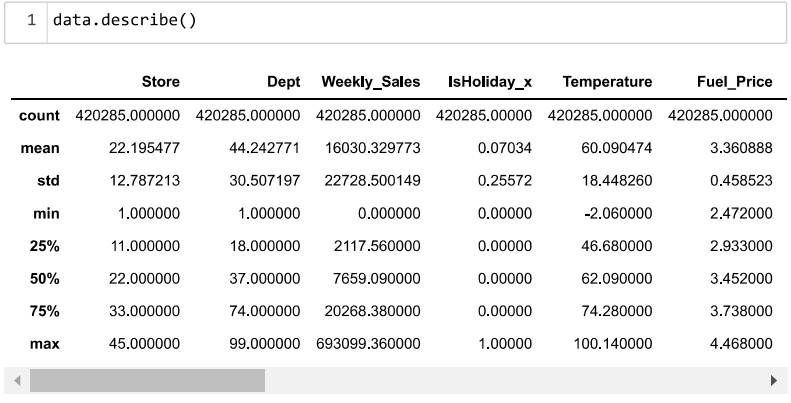
**Handling Negative Values**



As we can see that in the merged dataset, in the Weely\_Sales column, the minimum sales value is negative. Since sales cannot be in negative, they should always be zero or positive. So, only the positive values should be taken, which is done as follows.



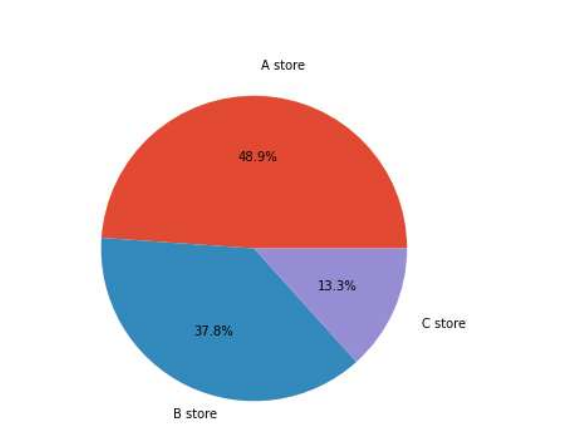
The above statement will only take positive values for the weekly\_sales column.



**Exploratory Data Analysis**

From the description of the stores dataset. We can observe that there are 3 types of stores: Type A, Type B, and Type C. Since there are 45 stores in total, we will plot a pie-chart to see the percentage of each store type.



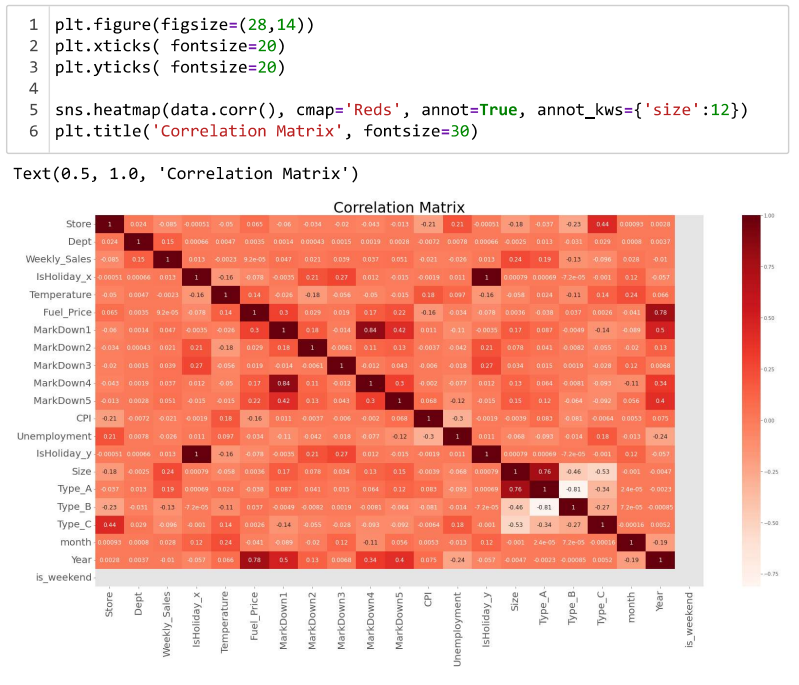


From the pie-chart it can be observed that Type A store have a higher median than the other store types.

We will then check the weekly sales on holidays and non-holidays.



We will also plot a correlation matrix to see the correlation that exists between the various features. Correlation is a bivariate analysis which measures the strength(+1 or –1) of association between two variables and the direction. The direction of the relationship is indicated by a positive or negative sign; a + sign indicates a positive relationship and a – sign indicates a negative relationship.



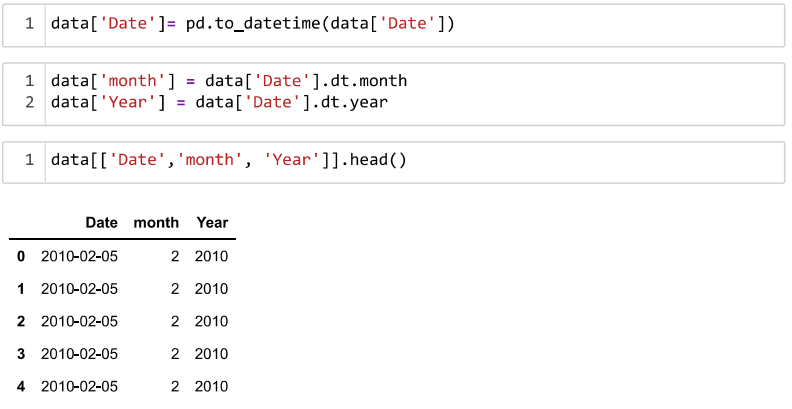
**Handling Categorical Values**

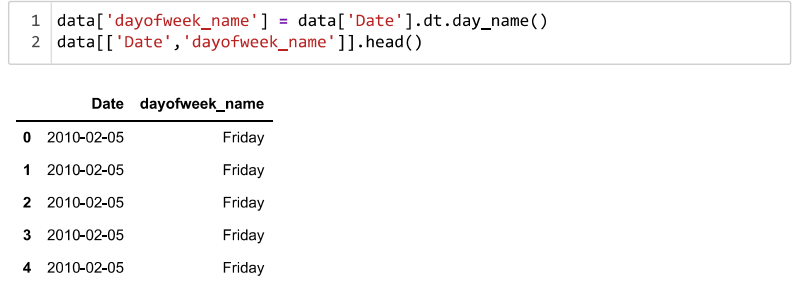
As can be observed, the dataset has categorical data which must be converted from categorical data to integer encoding or binary encoding.

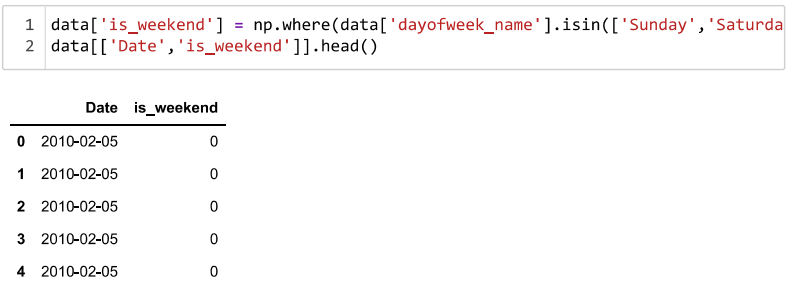
Firstly, the Type column for stores type is converted to indicator variable by using the get\_dummies() function.



Second, the date column is split according to date, month, year and day\_of\_the\_week.



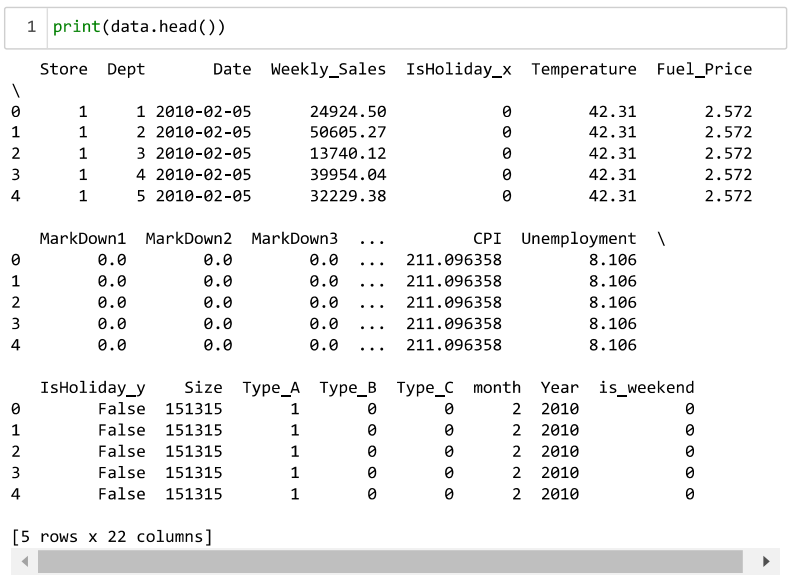




Lastly, the IsHoliday values are changed from Boolean True and False to interger 1 and 0.



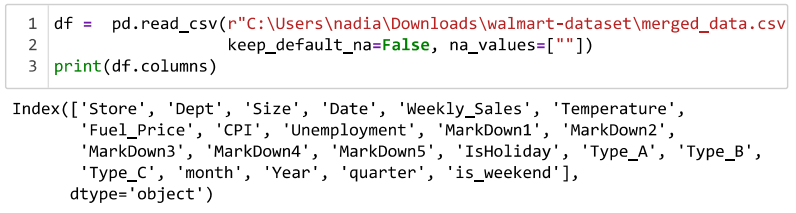
The first five rows of the transformed data is printed and then the changed data is stored into a new csv data format by using the .to\_csv() function.





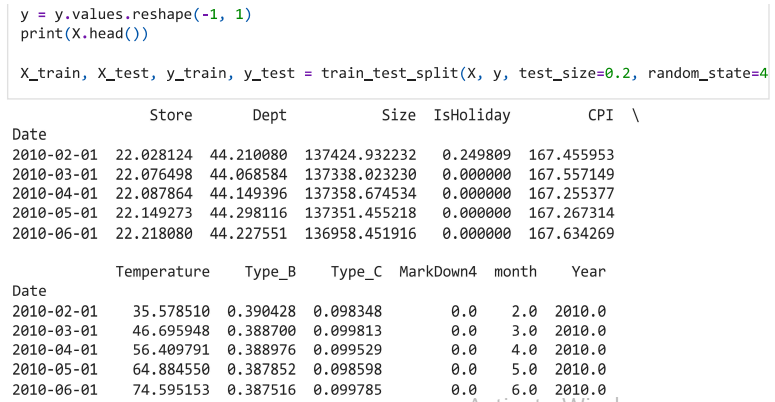
**Splitting data into train and test**

To split the dataset into train and test sets, the merged data file(transformed data file) will be first read.



Then the X and y variables are created. On the input variable, X is passed with dropping the target variable. And on target variable, y variable is passed. For splitting training and testing data, train\_test\_split() function is used from sklearn. As parameters, X, y, *test\_size*, *random\_state*.





We are specifying that from the merged data file, the weekly sales column is the target variable that on which the prediction will be made.

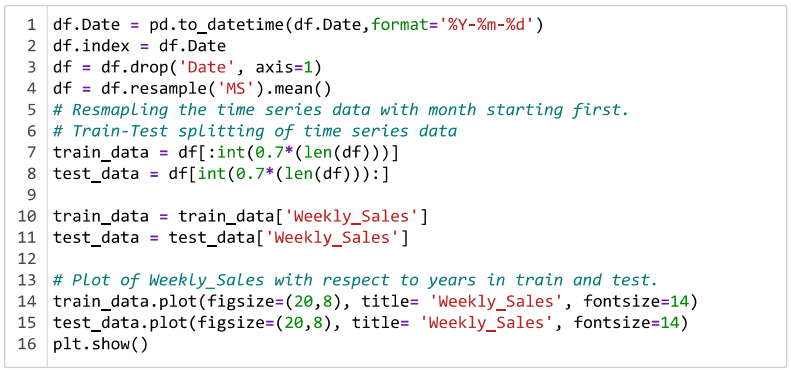
# Model Building

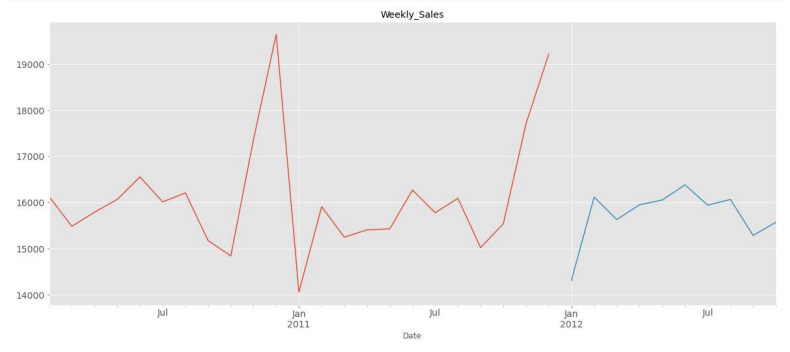
Now that the data is cleaned, it’s time to build the model. We can train our data on different algorithms. For this project we are applying the ARIMA time series forecasting algorithm is used along with Random Forest and XgBoost.

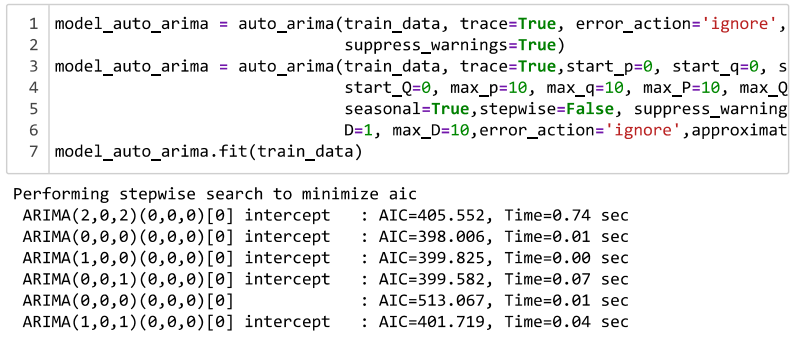
**ARIMA**

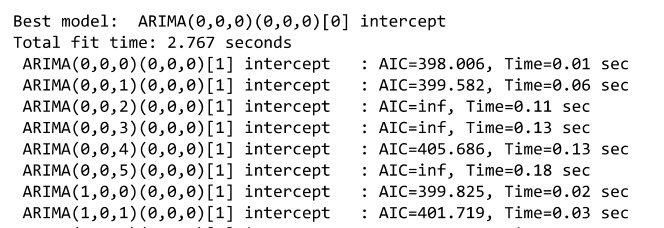
ARIMA is a forecasting technique that projects the future values of a series based entirely on its own inertia. The time series models assume that the given time-series data set in stationary meaning it has got the constant mean and constant variance. In order to apply ARIMA, we are required to calculate the triplet value (p,d,q). The value of p is number of Auto regressors, value of d is number of differences required to make series stationary and q is number of moving averages. In Auto-ARIMA, the triplet (p,q,d) values are automatically selected on the basis of least AIC and BIC scores which best fits the model.

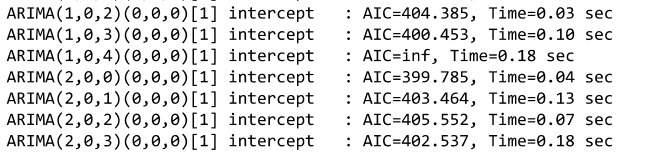


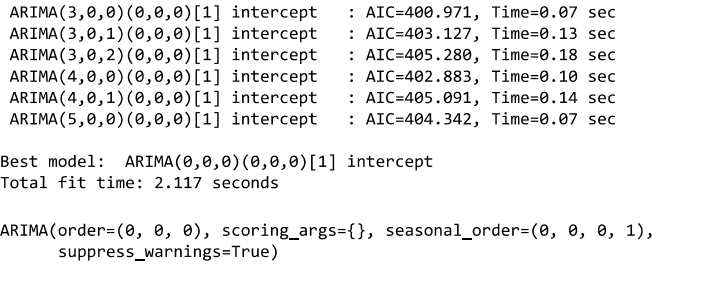


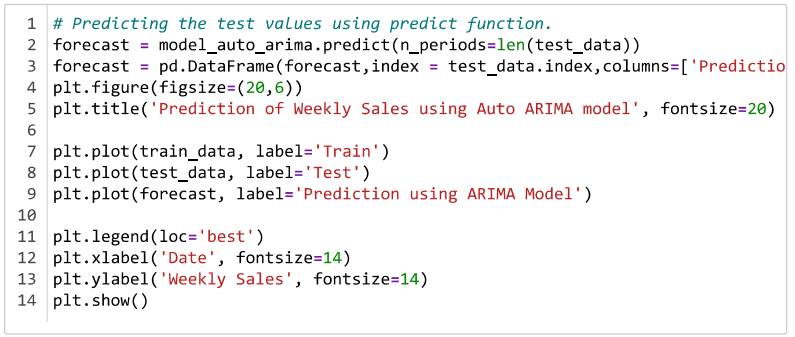


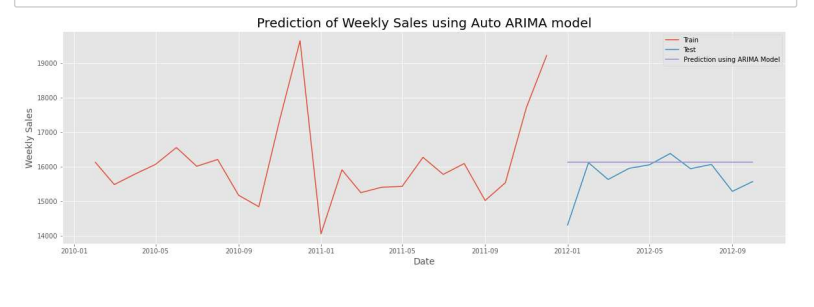


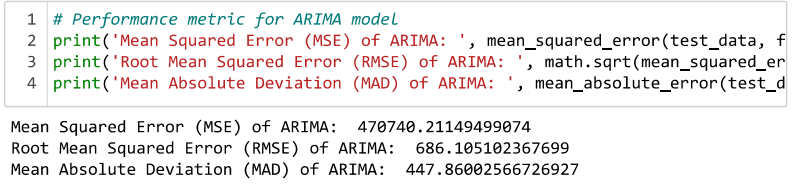






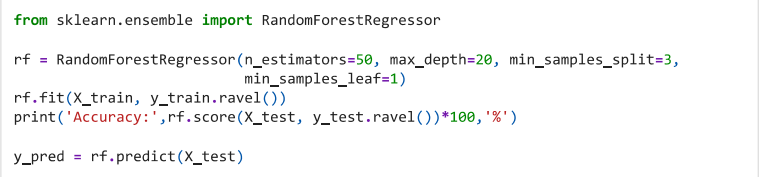


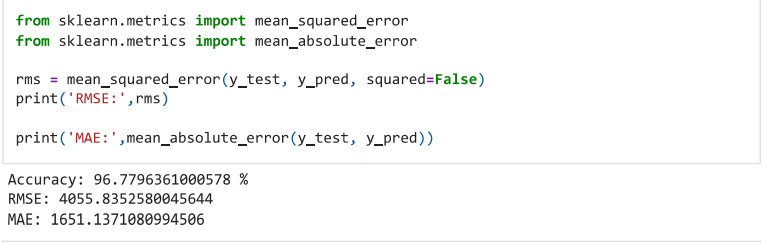


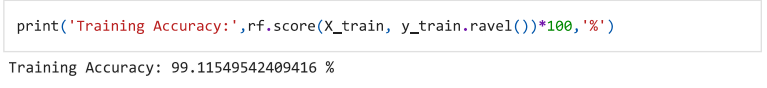


**Random Forest**

RandomForestRegressor algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable.

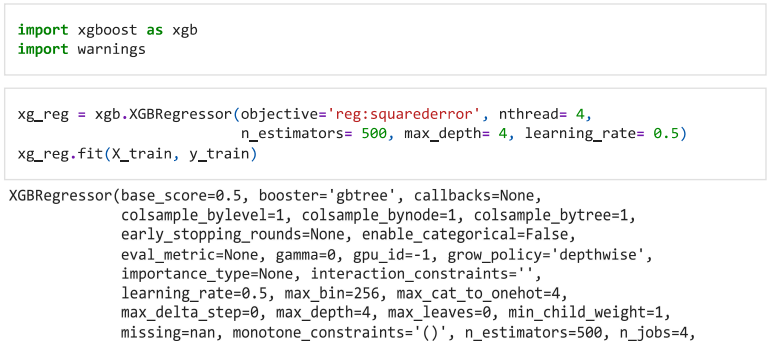


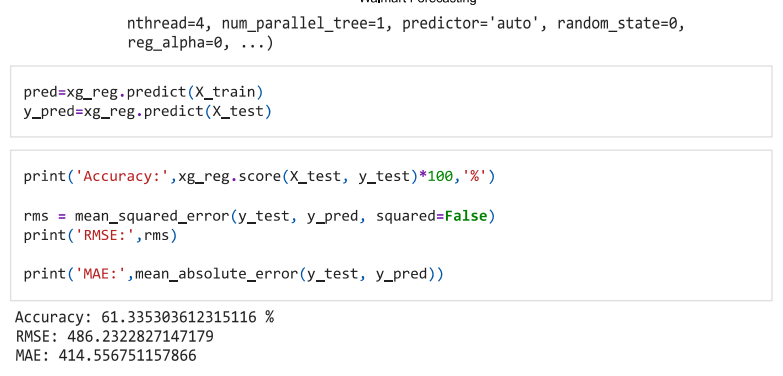


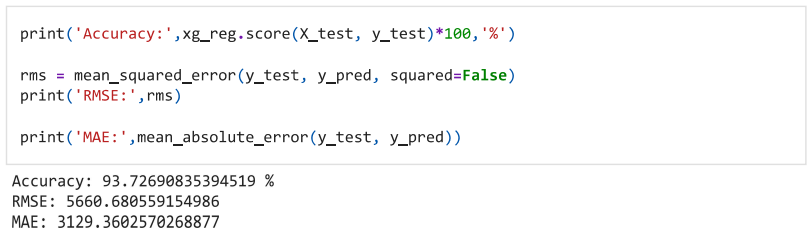


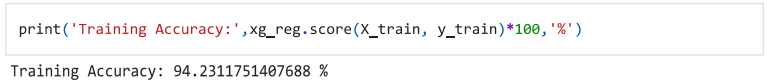
**XgBoost**

XGBRegressor algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable.



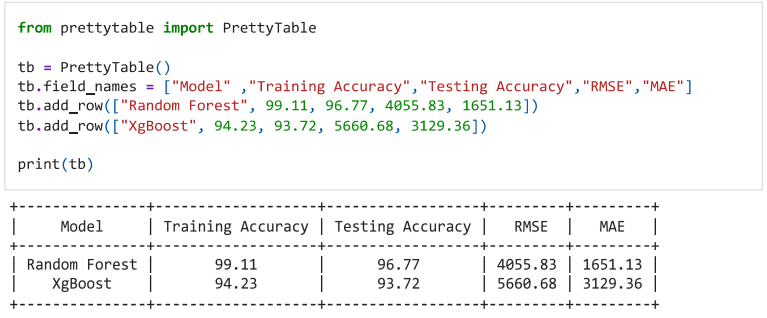




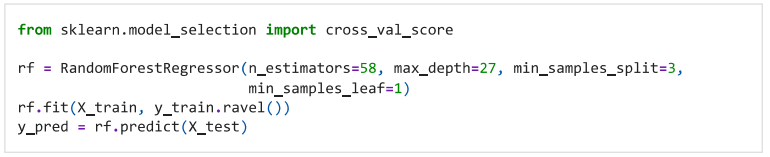


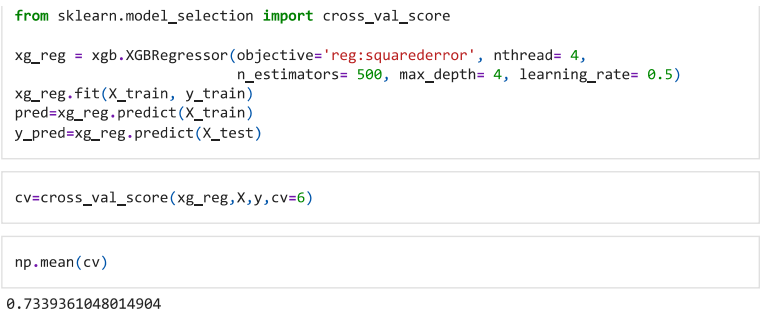
**Comparing the models**

For comparing the models PrettyTable function is defined.



After calling the function, the results of models are displayed as output. From all the models, Random forest is performing well. The XgBoost model is evaluated with cross validation.





The model is saved as



* **Application Building**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where they have to enter the values for predictions. The entered values are given to the saved model and prediction is showcased on the UI.

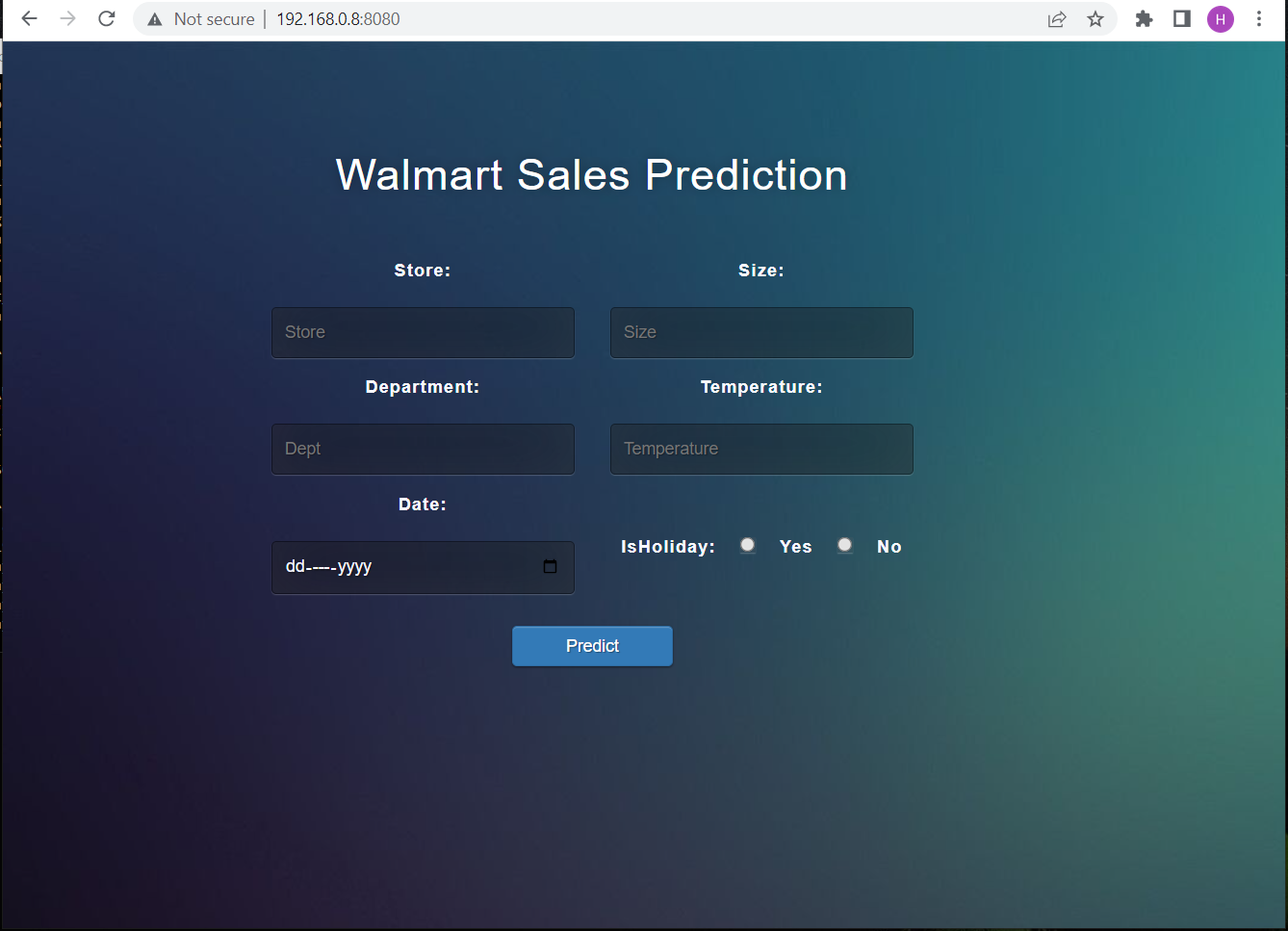
This section has the following tasks

* Building HTML Pages
* Building serverside script

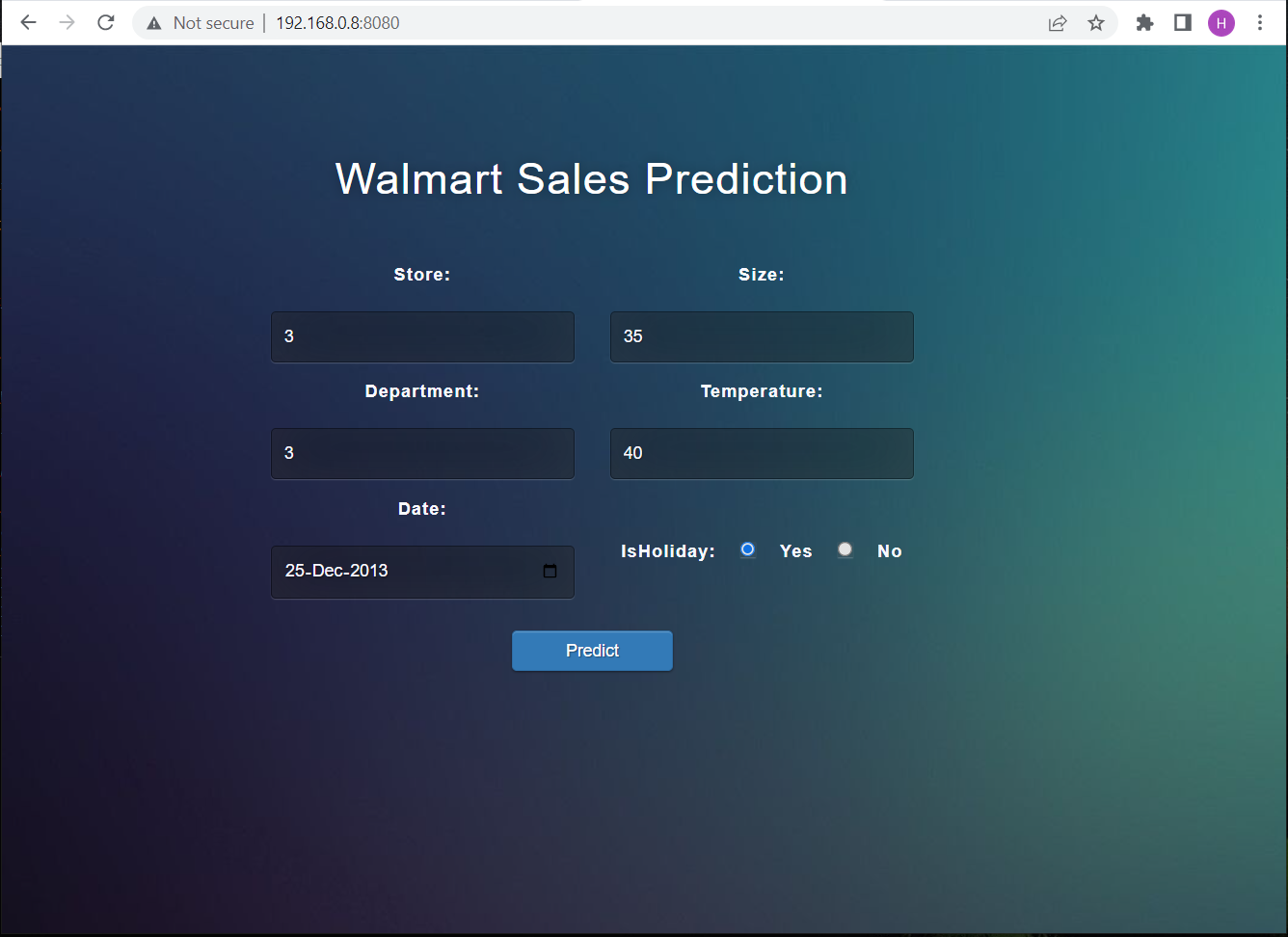
**Building HTML Pages:**

For this project, an HTML file called ‘index.html’ is created and saved in templates folder.

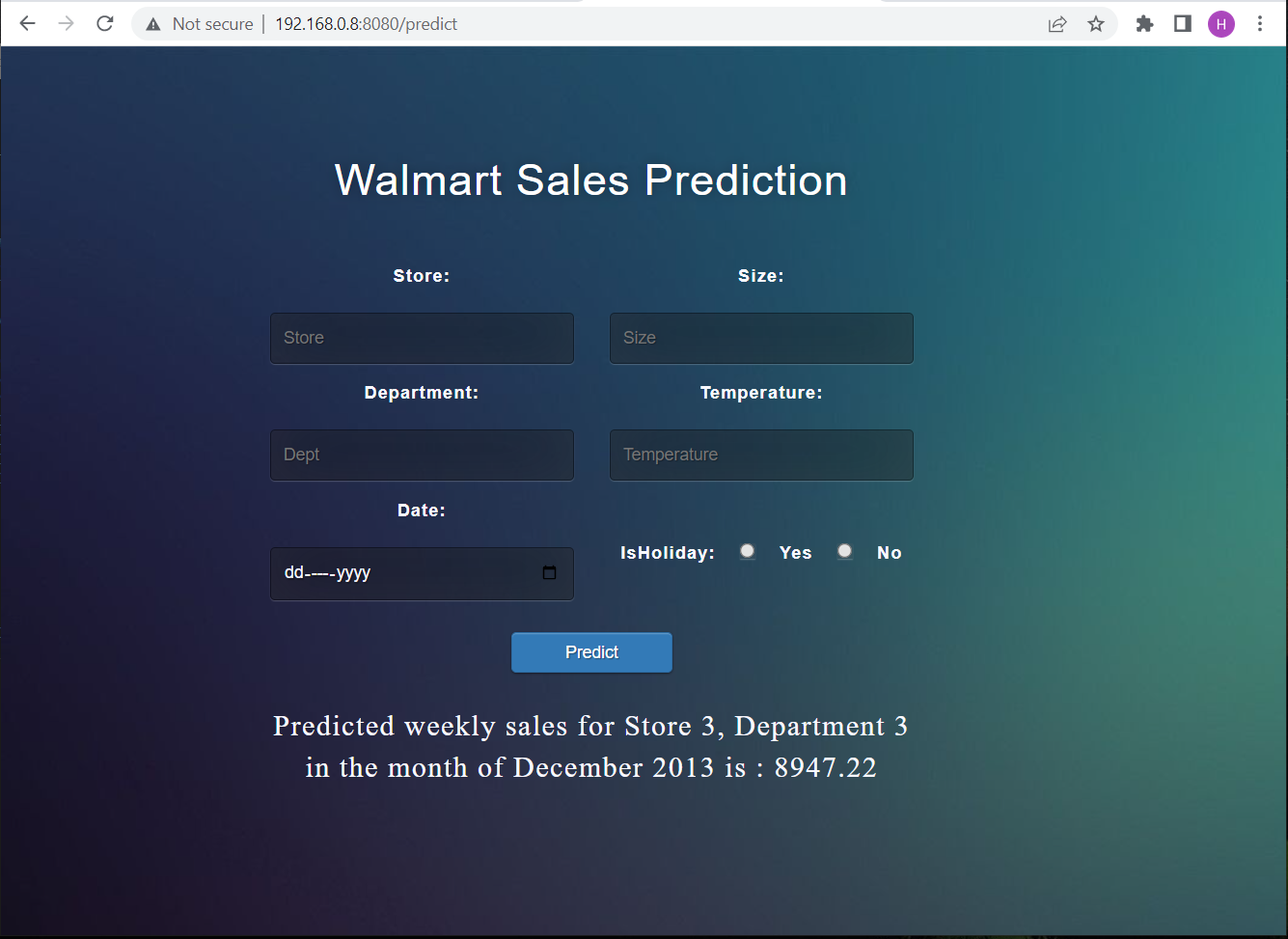
The index.html page looks like:



The index page or the main page is displayed and it has fields for entering certain values to predict the sale of a department and store on a particular and whether that day is a holiday or not.



After entering the details, we can perform the prediction by clicking on the predict button.



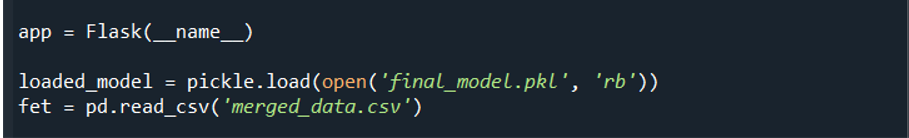
After the prediction is done, the predicted sales are displayed.

**Building the python code in flask:**

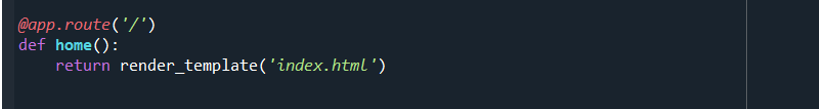
Import the libraries



Then load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument.



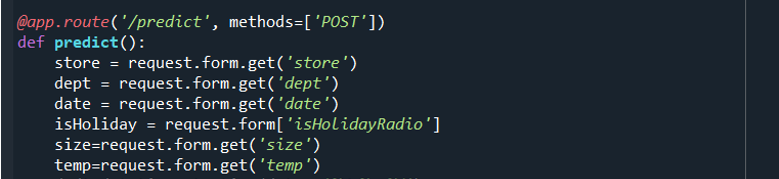
Render the HTML page:

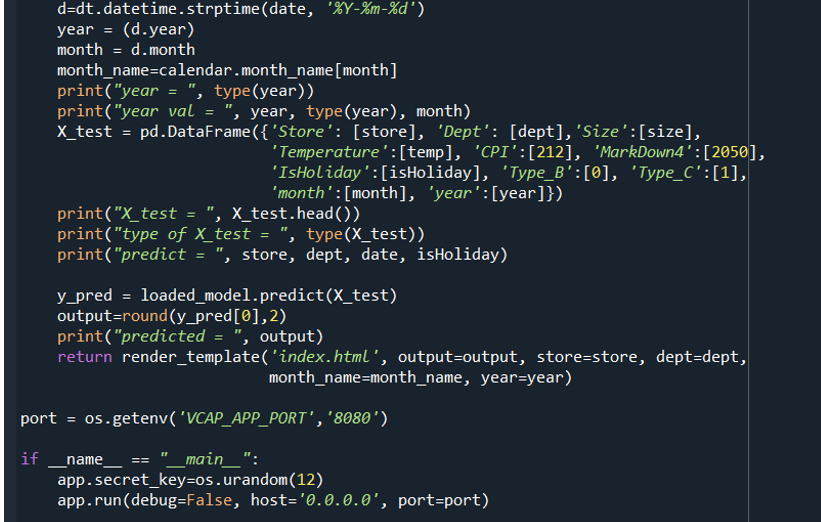


Here we will be using declared constructor to route to the HTML page which we have created earlier and stores in the templates folder.

In the above example, ‘/’ URL is bound with index.html function. Hence, when the home page of the web server is opened in browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

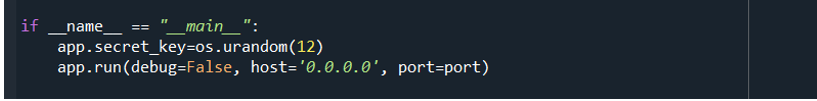
Retrieves the value from UI:





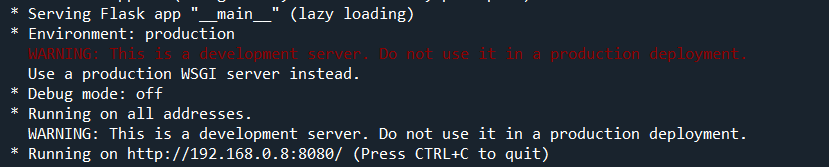
Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the y\_pred() function. This function returns the prediction. And this prediction value will rendered to the text that we have mentioned in the submit.html page earlier.

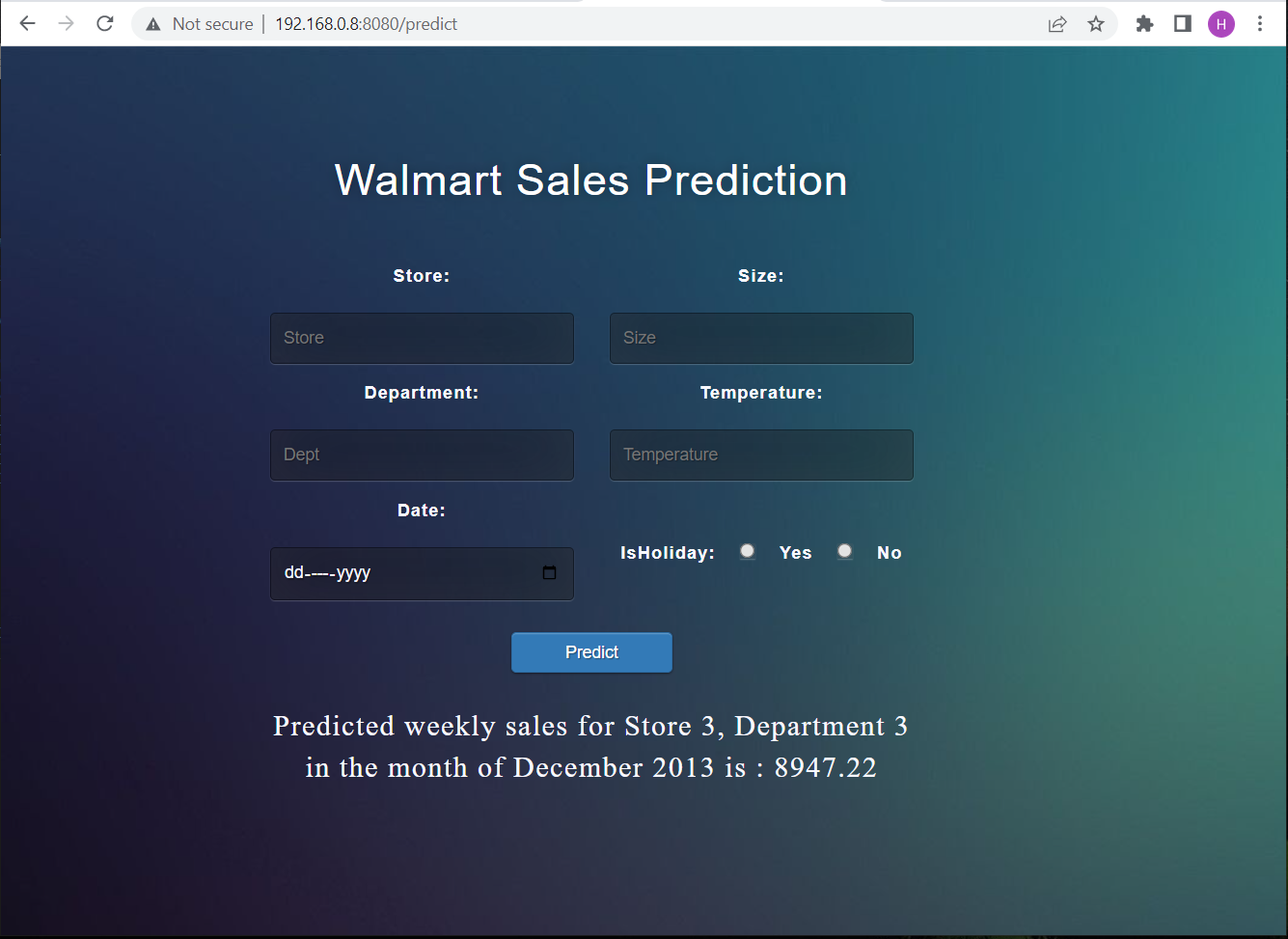
**Main Function:**



**Activity 3: Run the application**

* Open anaconda prompt from the start menu
* Navigate to the folder where your python script is.
* Now type “python app.py” command
* Navigate to the localhost where you can view your web page.
* Click on the predict button from the top right corner, enter the inputs, click on the submit button, and see the result/prediction on the web.





The output can be seen on the HTML page in the browser as well as on the anaconda prompt.

