A Mini Project Report

on

Sales forecasting through Machine Learning

submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Konda Rahul(17211A05D5)

Kova HimaBindu(17211A05E3)

Manchikanti Sreeja(17211A05G3)

Munchidala Akhilesh(17211A05J0)

Under the guidance of

P. Jhansi Devi, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

B.V.RAJU INSTITUTE OF TECHNOLOGY

(UGC Autonomous, Accredited by NBA & NAAC)

Vishnupur, Narspur, Medak(Dist.), Telangana State,India-502313

2017 - 20201

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC)
Vishnupur, Narspur, Medak (Dist.),
Telangana State, India – 502313

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project entitled **"Sales forecasting through Machine Learning"**, being submitted by

Konda Rahul(17211A05D5)
Kova HimaBindu(17211A05E3)
Manchikanti Sreeja(17211A05G3)
Munchidala Akhilesh(17211A05J0)

In partial fulfillment of the requirements for the award of degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING to B.V.RAJU INSTITUTE OF TECHNOLOGY is a record of bonafide work carried out during a period from May 2019 to July 2020 by them under the guidance of **P.Jhansi Devi**, Assistant Professor, CSE Department.

This is to certify that the above statement made by the students is/are correct to the best of my knowledge.

P.Jhansi Devi
Associate Professor

The Project Viva-Voce Examination of this team has been held on

___________________.

Mr. Karthik Kovuri	**Dr. Ch. Madhu Babu**	
Project Coordinator	Professor & HoD-CSE	EXTERNAL EXAMINER

B. V. Raju Institute of Technology
(UGC Autonomous, Accredited By NBA & NAAC)
Vishnupur, Narspur, Medak (Dist.),
Telangana State, India – 502313

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Sales forecasting through Machine Learning"** in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology and submitted in the Department of Computer Science and Engineering, B. V. Raju Institute of Technology, Narsapur is an authentic record of my own work carried out during a period from May 2019 to July 2020 under the guidance of **P.Jhansi Devi**, Associate Professor. The work presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Konda Rahul(17211A05D5)
Kova HimaBindu(17211A05E3)
Manchikanti Sreeja(17211A05G3)
Munchidala Akhilesh(17211A05J0)

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completion. Whatever we have done is due to such guidance and assistance. We would not forget to thank them.

We thank **P.Jhansi Devi** for guiding us and providing all the support in completing this project. We are thankful to **Mr. V. Pradeep Kumar**, our section project coordinator for supporting us in doing this project. We are thankful to **Mr. Karthik Kovuri**, project coordinator for helping us in completing the project in time. We thank the person who has our utmost gratitude is **Dr. Ch. Madhu Babu**, Head of CSE Department.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the staff members of CSE Department.

Konda Rahul(17211A05D5)
Kova HimaBindu(17211A05E3)
Manchikanti Sreeja(17211A05G3)
Munchidala Akhilesh(17211A05J0)

Sales forecasting through Machine Learning

ABSTRACT

The ability to predict the data accurately is extremely valuable in a vast array of domain such as stocks, sales or even sports. Presented here is the study and implementation of several ensemble classification algorithms employed on sales data, consisting of weekly retail sales numbers from different departments in Wal-Mart retail outlets all over the USA .The models implemented for prediction are random forests, gradient boosting and extremely randomized trees classifiers.

The hyperparameters of each model were varied to obtain the best mean absolute error (MAE) value and r2 score. The no of estimators hyperparameter , which specifies the no of decision trees used in the model, plays a particularly important role in the evaluation of the MAE value and r2 score and is dealt with in an attentive manner comparative analysis of the three algorithms is performed to indicate the best algorithm and the hyperparameter.

KEYWORDS:

machine learning,weekly sales,data sets,variables,random forest,models

CONTENTS

Chapter 1

INTRODUCTION

1.1 Motivation:

In today's world where competition is cut-throat and making business decisions is increasingly difficult , the propensity to accurately make predictions is of extreme relevance .The basis is of sales prediction which is a more established yet still profoundly captivating application of forecasting . When organizations spread their capital and customers possess a deluge of options, even the slightest upper hand will have a significant impact on fortunes of organization . Sales forecasting uses trends identified from historical data to predict future sales.

1.2 Problem Definition:

The decision makers of stores should be able to analyse the effects of various factors affecting the sales of their products in their stores .The various factors include weather conditions i.e., temperature,store size, fuel prices ,markdown in prices,unemployment and CPI to determine the sales.

1.3 Objective of Project:

The objective of the project are as followed:

- ❏ To analyze the sales across different departments of the stores type and create weekly and monthly dashboards
- ❏ To analyze the effects of various factors influencing the sales
- ❏ To identify the most significant factors
- ❏ To build a model able to predict the sales
- ❏ And check the efficiency of the model constructed.

1.4 Limitations of the project:

The sales data which belonged to the sales of particular regions and it cannot be assured that the similar results will be obtained from the study conducted on the sales data belonged to the other region as the sales may vary in other regions.

Due to the unavailability of the stores's information like customer details, certain campaigns and discounts. They haven't been included in data which would benefit in obtaining better forecasts.

Chapter 2

LITERATURE SURVEY

2.1 Introduction:

In today's world making business decisions is increasingly difficult, the propensity to accurately make predictions is of extreme relevance. For example, it would be exceptionally beneficial to be able to predict the ups and downs of a country's economy or the fluctuations of its stock market prices. Forecasting has been done across a wide array of domains and spheres including environmental fields such as weather or even in sports performance due to the advantageous nature of prediction. The basis of this idea of sales prediction which is a more established yet still profoundly captivating application of forecasting. When organizations spread their capital and customers possess a deluge of options, even the slightest upper hand will have a significant impact on the fortunes of the organization. Sales forecasting uses trends identified from historical data to predict future sales, enabling educated decisions including assigning or redirecting current inventory, or effectively managing future production.

2.2 Existing System:

Traditional statistical Forecasting

Which is good practise for stable markets ,ill-disposed to changes

Traditional statistical methods (TSM) have been here for ages and remain a staple of forecasting processes. The only difference if compared with the previous century is that all calculations are performed automatically, by modern software. For example, you can create forecasts for sales and trends in Excel. To predict the future, statistics utilizes data from the past. That's why statistical forecasting is often called *historical.* The common recommendation is collecting data on sales for at least two years. Traditional forecasting demands planning solutions based on statistical techniques seamlessly integrate with Excel and existing Enterprise Resource Planning (ERP) systems . The most advanced systems can consider seasonality and market trends as well as apply numerous methods to finetune results.

2.3 Disadvantages of Existing system:

An important prerequisite of statistical forecasting accuracy is stability. We assume that history repeats itself: Situations that occurred two or three years ago will reoccur. Which is far from being true. Flawless in an ideal world, statistical methods often fail to foresee illogical alterations in customer preferences or predict when market saturation will occur.

- Data growth issues
- Confusion in tool selection
- Lack of data professional
- Security of data
- Integration of variety of data

2.4 Proposed System:

Using Machine Learning for sales Forecasting.Machine learning applies complex mathematical algorithms to automatically recognize patterns, capture demand signals and spot complicated relationships in large datasets. Apart from analyzing huge volumes of information, smart systems continuously retrain models, adapting them to changing conditions thus addressing volatility. These capabilities enable ML-based software to produce more accurate and reliable forecasts in complex scenarios.

- aggregating historical and new data from different sources;
- cleansing data;
- determining which forecasting algorithm fits your product best;
- building predictive models to identify likely outcomes and discover relationships between various factors; and
- monitoring models to measure their business results and improve prediction accuracy.

Chapter3

ANALYSIS

3.1 Introduction:

In this chapter we are going to discuss the proposed system analysis in detail about what sales forecasting is made of, its features and requirements likewise the tools and technologies that are used are not left out.

3.2 Software requirements specification:

A software requirements specification (SRS) captures a complete description about how the system is expected to perform. Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

3.2.1 Software requirements:

Operating System : Windows Server 7/8 10(64-bit) Or Linux

Programming Language Translators: Python 3.7 .

Libraries:Pandas,Numpy,Matplotlib,Seaborns,SKlearn.

IDE : Anaconda with Jupiter.

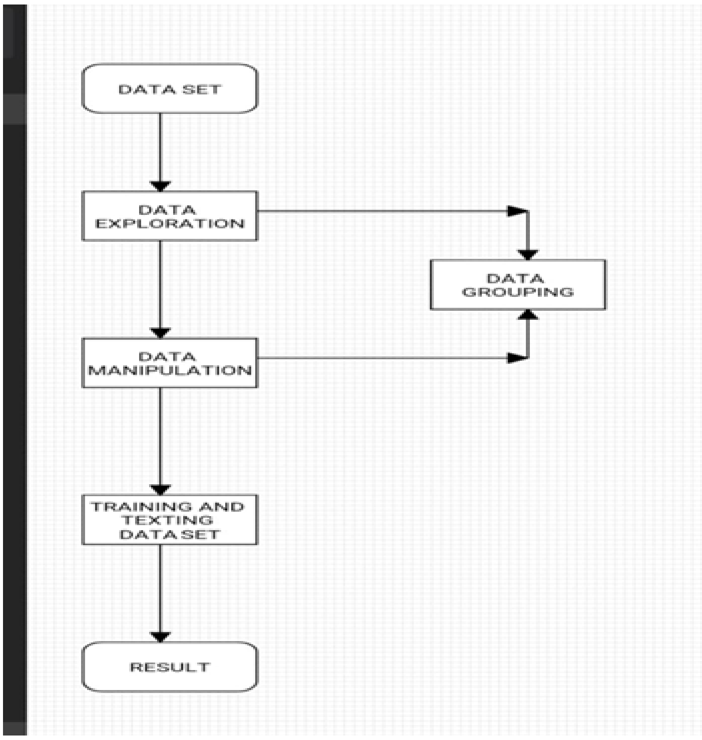
3.2.2 Hardware specifications:

Processor and Speed :64-bit,four-core,2.5 GHz minimum per core

RAM capacity : 4 GB for development and evaluation use

Hard Disk : 10GB for development and evaluation use in total capacity Of 80GB

3.3 Algorithms and Flowcharts:



DATA EXPLORATION: It is the initial step in data analysis,where users explore a large data set in an unstructured way to uncover initial patterns,characteristics and points of interest. It is a combination of manual methods and automated tools.

DATA MANIPULATION: It refers to the process of adjusting data to make it organised and easier to read.Dta manipulation adjusts data by inserting,deleting and modifying data .

DATA GROUPING: Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points ,we can use a clustering algorithm to classify each data point into a specific group.

TRAINING AND TESTING DATASET: Train/Test is a method to measure the accuracy of your model it is called Train/Test because you split the data into two sets :a training set and a testing set.80%for training and 20% for testing .You train the model using the training set.

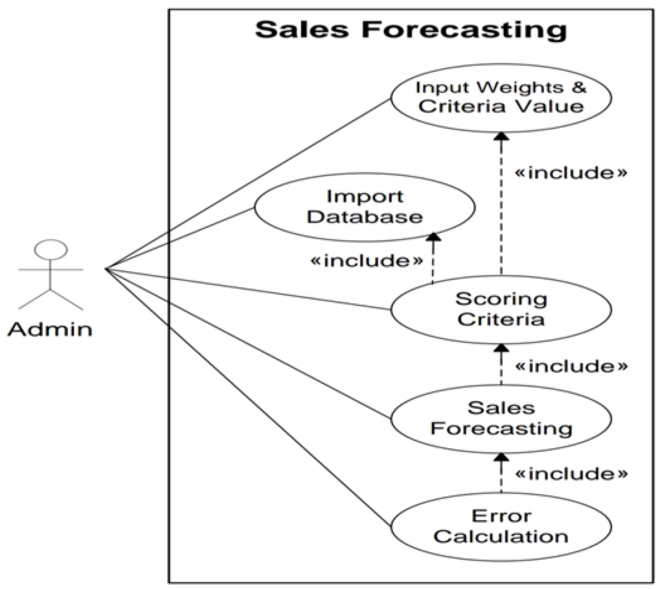
Chapter 4
DESIGN

4.1 Introduction:

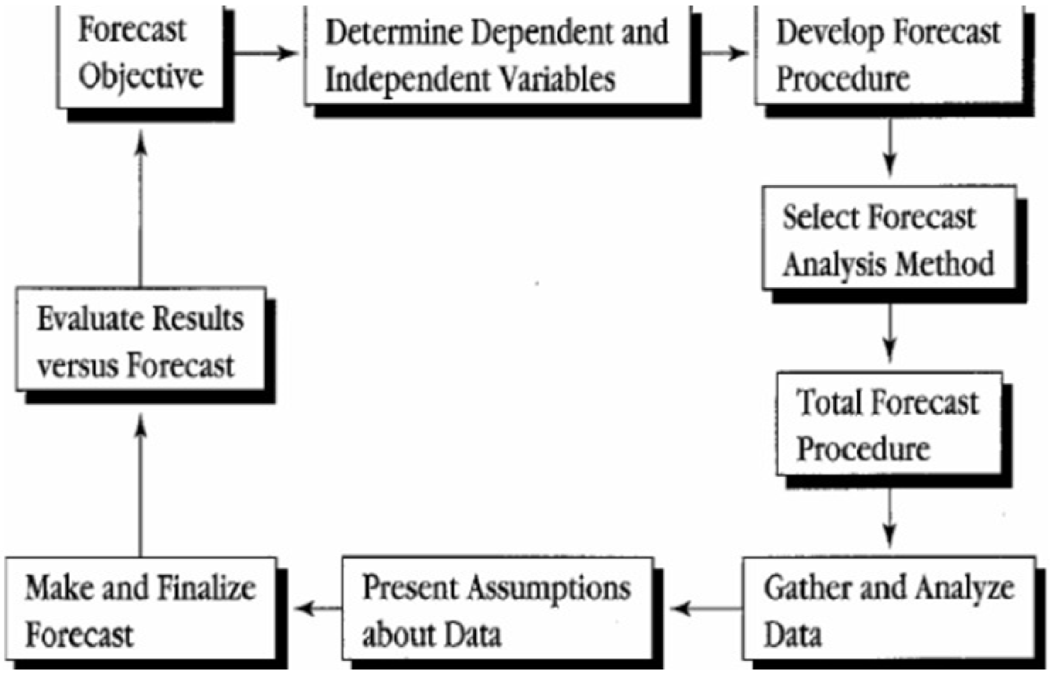
Design is the first step in moving from problem domain to solution domain. The purpose of the design phase is to plan a solution of the problem specified by the requirements document. Starting with what is needed, design takes towards how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software. It has a major impact on the project during later phases, particularly during testing and maintenance. The output of this phase is the design document. This document is similar to a blueprint or plan for the solution and is used later during implementation, testing and maintenance.

4.2 UML Diagrams:

Use Case Diagram:



Architecture Diagram:



Class diagram :

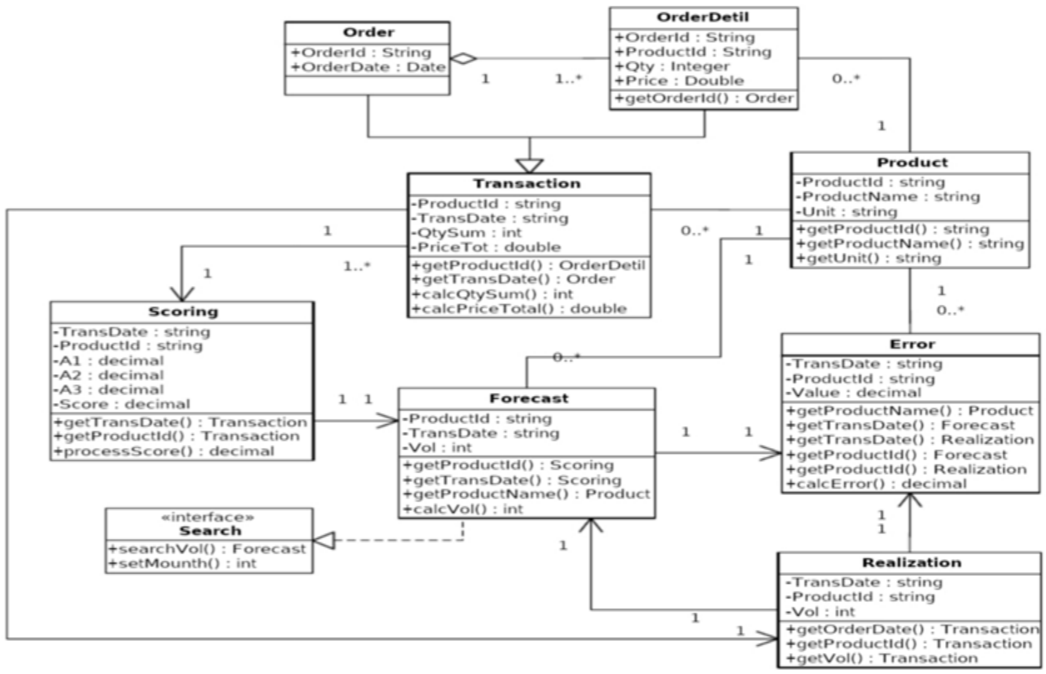


Fig. 2: The class diagram for the sales forecasting application

4.3 Module design and organization:

The main modules of the system are:

- collection of Data
- cleaning the raw data
- analysing the data
- identifying significant variable
- building model
- finding the best fit model

Chapter 5

IMPLEMENTATION

5.1 Introduction:

The system implementation defines the construction, installation, testing and delivery of the proposed system. After thorough analysis and design of the system, the system implementation incorporates all other development phases to produce a functional system

5.2 Explanation of Key Functions:

1.Dataset Overview

This data set is available on the kaggle website. These data sets contained information about
the stores, departments, temperature, unemployment, CPI, isHoliday, and MarkDowns.

Stores :

Store: The store number. Range from 1–45.
Type: Three types of stores 'A', 'B' or 'C'.
Size: Sets the size of a Store would be calculated by the no. of products available in the particular store ranging from 34,000 to 210,000.

Features:
Temperature: Temperature of the region during that week.
Fuel_Price: Fuel Price in that region during that week.

MarkDown1:5 : Represents the Type of markdown and what quantity was available during that week.
CPI: Consumer Price Index during that week.
Unemployment: The unemployment rate during that week in the region of the store.

Sales:
Date: The date of the week where this observation was taken.
Weekly_Sales: The sales recorded during that Week.
Dept: One of 1–99 that shows the department.
IsHoliday: a Boolean value representing a holiday week or not.

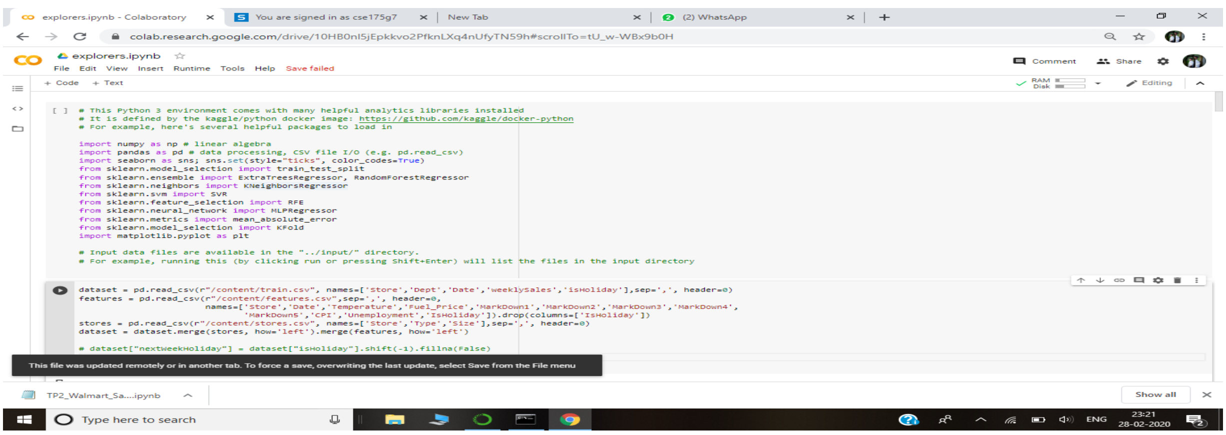
2.Libraries and Data Loading

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from matplotlib.gridspec import GridSpec
import seaborn as sns
from scipy import stats
from scipy.special import boxcox1p

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

import warnings
warnings.filterwarnings("ignore") # ignoring annoying warnings

from pandasql import sqldf
pysqldf = lambda q: sqldf(q, globals())
features = pd.read_csv('../input/walmart-recruiting-store-sales-forecasting/features.csv.zip')
train = pd.read_csv('../input/walmart-recruiting-store-sales-forecasting/train.csv.zip')
stores = pd.read_csv('../input/walmart-recruiting-store-sales-forecasting/stores.csv')
test = pd.read_csv('../input/walmart-recruiting-store-sales-forecasting/test.csv.zip')
sample_submission =
pd.read_csv('../input/walmart-recruiting-store-sales-forecasting/sampleSubmission.csv.zip')
```



	Store	Dept	Date	weeklySales	isHoliday	Type	Size	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment
0	1	1	2010-02-05	24924.50	False	A	151315	42.31	2.572	NaN	NaN	NaN	NaN	NaN	211.096358	8.106
1	1	1	2010-02-12	46039.49	True	A	151315	38.51	2.548	NaN	NaN	NaN	NaN	NaN	211.242170	8.106
2	1	1	2010-02-19	41595.55	False	A	151315	39.93	2.514	NaN	NaN	NaN	NaN	NaN	211.289143	8.106
3	1	1	2010-02-26	19403.54	False	A	151315	46.63	2.561	NaN	NaN	NaN	NaN	NaN	211.319643	8.106
4	1	1	2010-03-05	21827.90	False	A	151315	46.50	2.625	NaN	NaN	NaN	NaN	NaN	211.350143	8.106
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
421565	45	98	2012-09-28	508.37	False	B	118221	64.88	3.997	4556.61	20.64	1.50	1601.01	3288.25	192.013558	8.684
421566	45	98	2012-10-05	628.10	False	B	118221	64.89	3.985	5046.74	NaN	18.82	2253.43	2340.01	192.170412	8.667
421567	45	98	2012-10-12	1061.02	False	B	118221	54.47	4.000	1956.28	NaN	7.89	599.32	3990.54	192.327265	8.667
421568	45	98	2012-10-19	760.01	False	B	118221	56.47	3.969	2004.02	NaN	3.18	437.73	1537.49	192.330854	8.667
421569	45	98	2012-10-26	1076.80	False	B	118221	58.85	3.882	4018.91	58.08	100.00	211.94	858.33	192.308899	8.667

421570 rows × 16 columns

3.Data manipulation

Checking for null values

feat.isnull.sum()

```
Store              0
Date               0
Temperature        0
Fuel_Price         0
MarkDown1       4158
MarkDown2       5269
MarkDown3       4577
MarkDown4       4726
MarkDown5       4140
CPI              585
Unemployment     585
IsHoliday          0
dtype: int64
```

From the output
we have few NaN for CPI and Unemployment, therefore we fill the missing values with their respective column mean.
And as MarkDowns have more missing values we impute zeros in missing places respectively

```
from statistics import mean
feat['CPI'] = feat['CPI'].fillna(mean(feat['CPI']))
feat['Unemployment']=feat['Unemployment'].fillna(mean(feat['Unemployment']))
feat['MarkDown1'] = feat['MarkDown1'].fillna(0)
```

```
feat['MarkDown2'] = feat['MarkDown2'].fillna(0)
feat['MarkDown3'] = feat['MarkDown3'].fillna(0)
feat['MarkDown4'] = feat['MarkDown4'].fillna(0)
feat['MarkDown5'] = feat['MarkDown5'].fillna(0)
```

4.**Exploratory Analysis:**

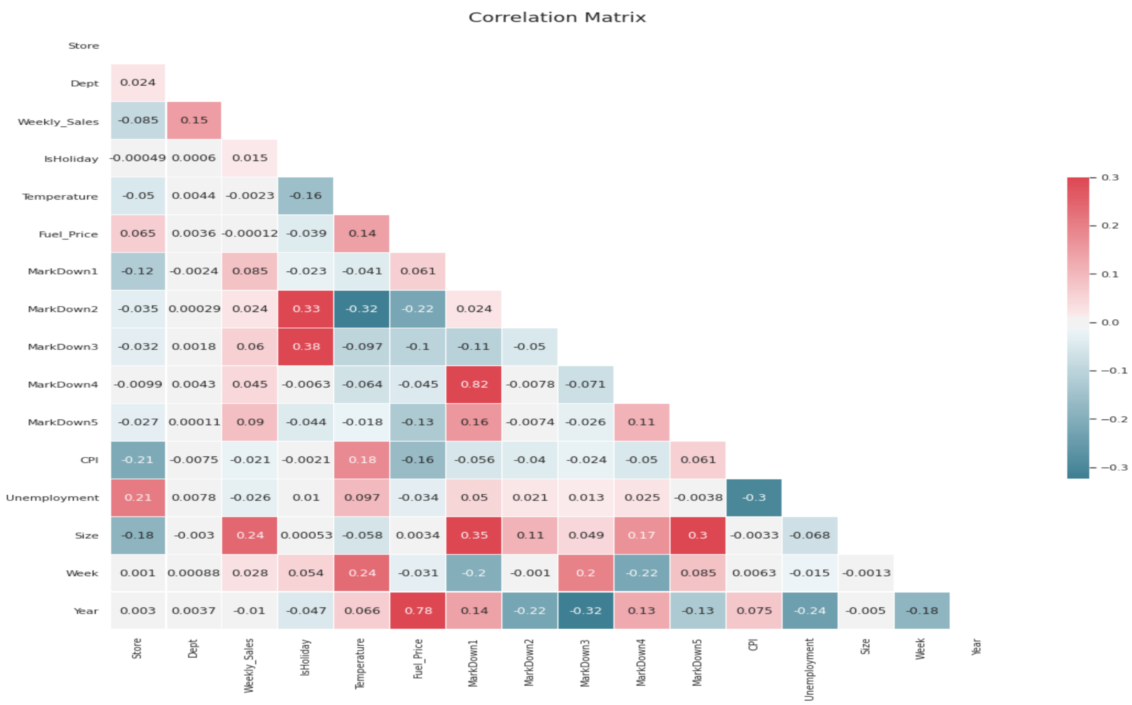
Variables CorrelationLet's see the correlation between variables, using Pearson Correlation.

Correlation Metrics:

- 0: no correlation at all
- 0-0.3: weak correlation
- 0.3-0.7: moderate correlation
- 0.7-1: strong correlation

Positive Correlation indicates that when one variable increases, the other also does. Negative is the opposite.

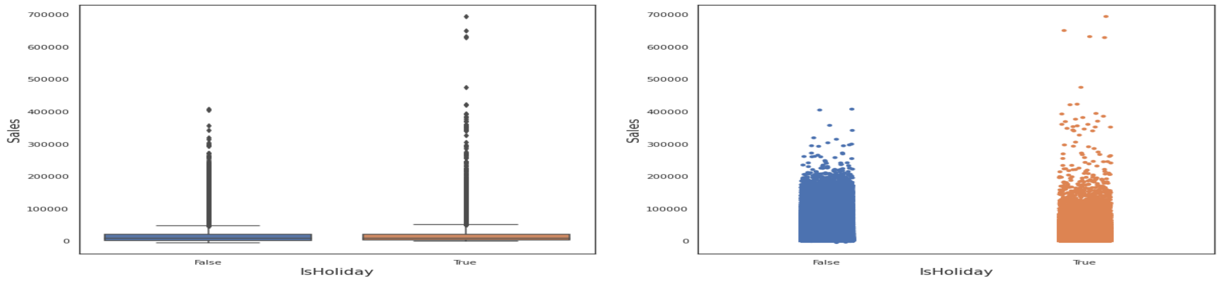
```
sns.set(style="white")
corr = train_detail.corr()
mask = np.triu(np.ones_like(corr, dtype=np.bool))
f, ax = plt.subplots(figsize=(20, 15))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
plt.title('Correlation Matrix', fontsize=18)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3,
center=0,square=True, linewidths=.5, cbar_kws={"shrink": .5},annot=True)
plt.show()
```



Analyzing Variables

Weekly_Sales x IsHoliday

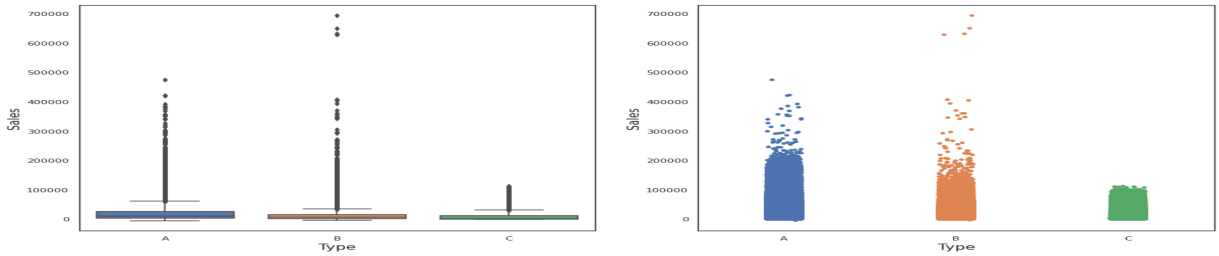
make_discrete_plot('IsHoliday')



This field is going to be important to differentiate Week Holidays. As we can see, Week Holidays have more high sales events than non-Holiday Weeks.

Weekly_Sales x Type

make_discrete_plot('Type')



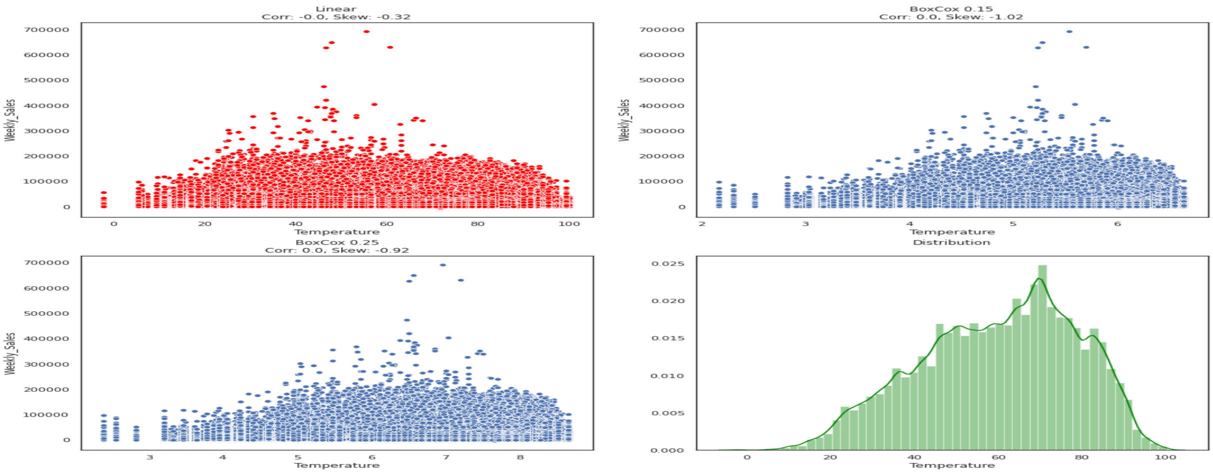
We don't know what 'Type' is, but we can assume that A > B > C in terms of Sales Median. So, let's treat it as an ordinal variable and replace its values.

Ordinal variables are explained in the figure below.

```
train_detail.Type = train_detail.Type.apply(lambda x: 3 if x == 'A' else(2 if x == 'B' else 1))
test_detail.Type = test_detail.Type.apply(lambda x: 3 if x == 'A' else(2 if x == 'B' else 1))
```

Weekly_Sales x Temperature

make_continuous_plot('Temperature')



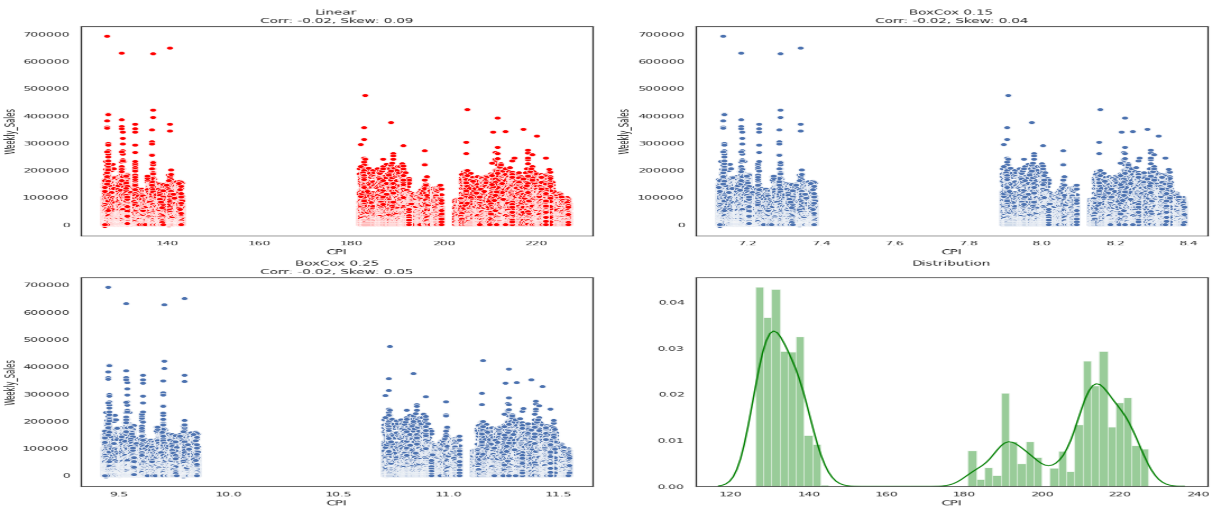
Although skewness changes, correlation doesn't seem to change at all. We can decide to drop it.

train_detail = train_detail.drop(columns=['Temperature'])

test_detail = test_detail.drop(columns=['Temperature'])

Weekly_Sales x CPI

make_continuous_plot('CPI')



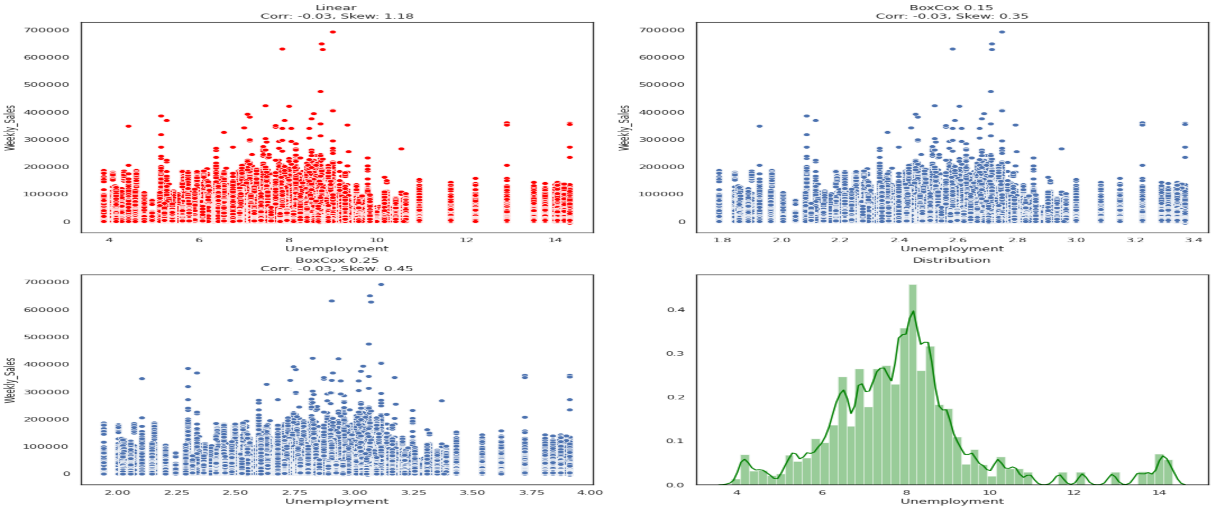
Same for 'CPI'.

train_detail = train_detail.drop(columns=['CPI'])

test_detail = test_detail.drop(columns=['CPI'])

Weekly_Sales x Unemployment

make_continuous_plot('Unemployment')



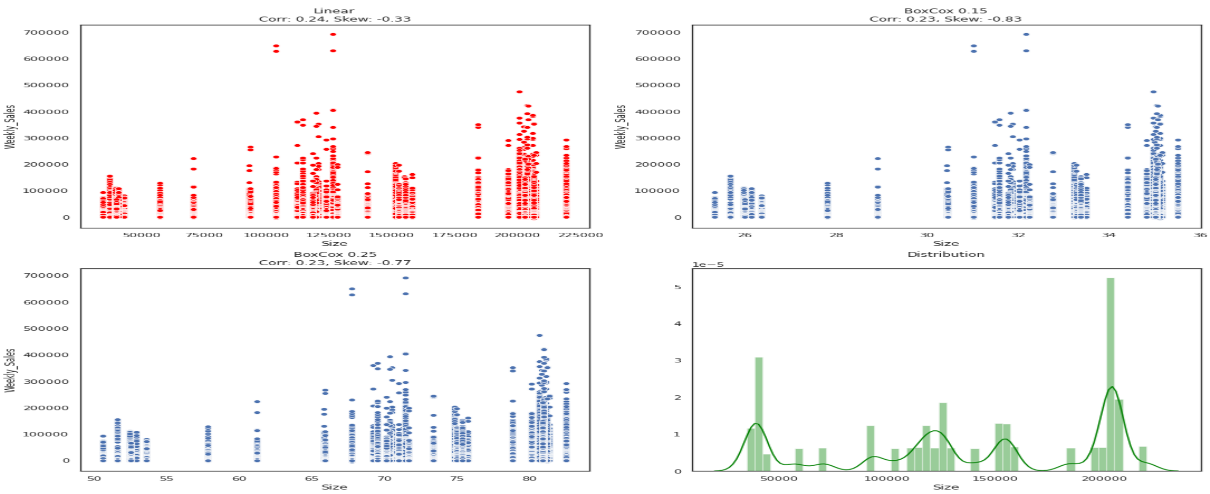
Same for 'Unemployment' rate.

train_detail = train_detail.drop(columns=['Unemployment'])

test_detail = test_detail.drop(columns=['Unemployment'])

Weekly_Sales x Size

make_continuous_plot('Size')



And, finally, we will continue with this variable, since it has moderate correlation with 'WeeklySales'.

5.3 Method of implementation:

5.3.1 Forms and 5.3.2 Output Screens:

Model functions:

As we can see in the figure below, the evaluation is based on Weighted Mean Absolute Error (WMAE), with a weight of 5 for Holiday Weeks and 1 otherwise.

This competition is evaluated on the weighted mean absolute error (WMAE):

$$\text{WMAE} = \frac{1}{\sum w_i} \sum_{i=1}^{n} w_i |y_i - \hat{y}_i|$$

where

- n is the number of rows
- $\hat{y}_i$ is the predicted sales
- y_i is the actual sales
- w_i are weights. w = 5 if the week is a holiday week, 1 otherwise

```
def WMAE(dataset, real, predicted):
    weights = dataset.IsHoliday.apply(lambda x: 5 if x else 1)
    return np.round(np.sum(weights*abs(real-predicted))/(np.sum(weights)), 2)
```

The model chosen for this project is the Random Forest Regressor. It is an ensemble method and uses multiples decision trees ('n_estimators' parameter of the model) to determine final output, which is an average of the outputs of all trees.

Training Model

Preparing Train Set.

```
X_train=train_detail[['Store','Dept','IsHoliday','Size','Week','Type','Year']]
        Y_train = train_detail['Weekly_Sales']
```

Final model:

```
RF = RandomForestRegressor(n_estimators=58, max_depth=27, max_features=6, min_samples_split=3, min_samples_leaf=1)
RF.fit(X_train, Y_train)
```

Predictions

Same fields for Test Data.

```
X_test = test_detail[['Store', 'Dept', 'IsHoliday', 'Size', 'Week', 'Type', 'Year']]
predict = RF.predict(X_test)
```

Christmas Adjustment

Ok, now it's time to make the Christmas Adjustment.
We can remember that Christmas Week has 0 pre-holiday days in 2010, 1 in 2011 and 3 in 2012. So, it's a difference of 3 days from 2012 to 2010 and 2 days from 2012 to 2011. A 2.5 days average, in a week (7 days). So, this is the value that we are going to multiply to Week 51 and add to Week 52 to compensate what the model didn't take into account.
But we are going to use this formula just for 'Stores'+'Departments' that have a big difference between Week 51 and Week 52 Sales. Let's say Week51 > 2 * Week52.
Let's use another dataframe and SQL to solve it quickly.

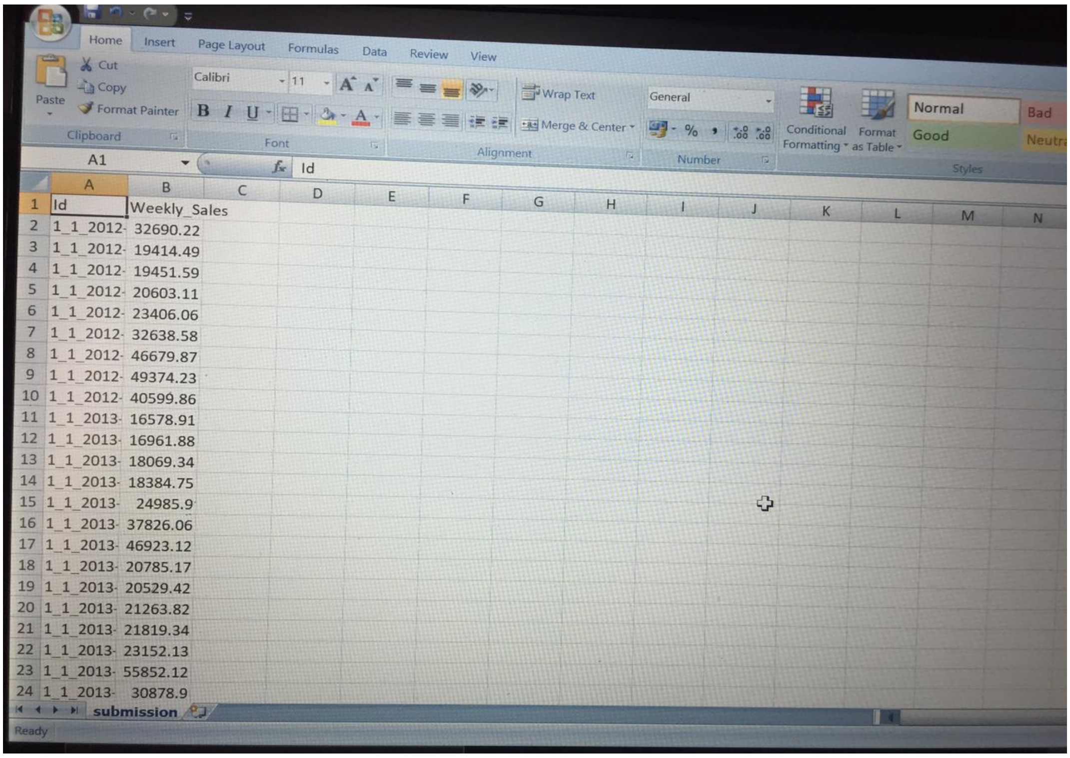
```
Final = X_test[['Store', 'Dept', 'Week']]
Final['Weekly_Sales'] = predict

Final_adj = pysqldf("""
    SELECT
        Store,
        Dept,
        Week,
        Weekly_Sales,
        case
            when Week = 52 and last_sales > 2*Weekly_Sales then
Weekly_Sales+(2.5/7)*last_sales
            else Weekly_Sales
        end as Weekly_Sales_Adjusted
    from(
        SELECT
            Store,
            Dept,
            Week,
            Weekly_Sales,
            case
                when Week = 52 then lag(Weekly_Sales) over(partition by Store,
Dept)
            end as last_sales
        from Final)""")
```

That's it. Let's make the submission.
Last time I checked, the submission file returned 2688.84 (Private) and 2673.97 (Public).

```
sample_submission['Weekly_Sales'] = Final_adj['Weekly_Sales_Adjusted']
sample_submission.to_csv('submission.csv',index=False)
```

5.3.3 Result Analysis:



The main aim of this project is sales forecasting which is important in making business decisions, through this project we are able to predict weekly sales of a store in given condition by exploring various variables influencing sales. This would make dynamic changes in forecasting systems and we can further improve this by future engineering .

The sales data is used to build a model that can forecast by diving in into train and test data this would act on its own in sales forecasting system with the experience of past data.

Chapter 6

TESTING AND VALIDATION

6.1 Introduction:

- Quality assurance is required to make sure that the software system works according to the requirements. Were all the features implemented as agreed? Does the program behave as expected? All the parameters that you test the program against should be stated in the technical specification document.
- Moreover, software testing has the power to point out all the defects and flaws during development. You don't want your clients to encounter bugs after the software is released and come to you waving their fists. Different kinds of testing allow us to catch bugs that are visible only during runtime.

However, in machine learning, a programmer usually inputs the data and the desired behavior, and the logic is elaborated by the machine. This is especially true for deep learning. Therefore, the purpose of machine learning testing is, first of all, to ensure that this learned logic will remain consistent, no matter how many times we call the program.

6.2 Testing:

Usually, software testing includes:

- Unit tests. The program is broken down into blocks, and each element (unit) is tested separately.
- Regression tests. They cover already tested software to see if it doesn't suddenly break.
- Integration tests. This type of testing observes how multiple components of the program work together.

Moreover, there are certain rules that people follow: don't merge the code before it passes all the tests, always test newly introduced blocks of code, when fixing bugs, write a test that captures the bug

6.3 Validation:

Validation is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also referred to as a software quality control. Software validation checks that the software product satisfies or fits the intended use.
Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artifacts or as needs of those who will operate the software only; but, as the needs of all the stakeholders (such as users, operators, administrators, managers, investors, etc.). There are two ways to perform software validation: internal and external. During internal software validation, it is assumed that the goals of the stakeholders were correctly understood and that they were expressed in the requirement artifacts precisely and comprehensively. If the software meets the requirement specification, it has been internally validated. External validation happens when it is performed by asking the stakeholders if the software meets their needs. Different software development methodologies call for different levels of user and stakeholder involvement and feedback; so, external validation can be a discrete or a continuous event. Successful final external validation occurs when all the stakeholders accept the software product and express that it satisfies their needs. Such final external validation requires the use of an acceptance test which is a dynamic test.

However, it is also possible to perform internal static tests to find out if it meets the requirements specification but that falls into the scope of static verification because the software is not running.

```
[ ] splited = pd.concat(splited).reset_index(drop=True)
```

```
[ ] splited
```

	Store	Dept	weeklySales	isHoliday	Size	Temperature	MarkDown1	MarkDown2	MarkDown4	MarkDown5	Type_A	Type_B	Type_C	Month	fold
0	1	1	24924.50	False	151315	42.31	0.00	0.00	0.00	0.00	1	0	0	2	0.0
1	1	1	46039.49	True	151315	38.51	0.00	0.00	0.00	0.00	1	0	0	2	0.0
2	1	1	41595.55	False	151315	39.93	0.00	0.00	0.00	0.00	1	0	0	2	0.0
3	1	1	19403.54	False	151315	46.63	0.00	0.00	0.00	0.00	1	0	0	2	0.0
4	1	1	21827.90	False	151315	46.50	0.00	0.00	0.00	0.00	1	0	0	3	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
421269	45	98	508.37	False	118221	64.88	4556.61	20.64	1601.01	3288.25	0	1	0	9	4.0
421270	45	98	628.10	False	118221	64.89	5046.74	0.00	2253.43	2340.01	0	1	0	10	4.0
421271	45	98	1061.02	False	118221	54.47	1956.28	0.00	599.32	3990.54	0	1	0	10	4.0
421272	45	98	760.01	False	118221	56.47	2004.02	0.00	437.73	1537.49	0	1	0	10	4.0
421273	45	98	1076.80	False	118221	58.85	4018.91	58.08	211.94	858.33	0	1	0	10	4.0

421274 rows × 15 columns

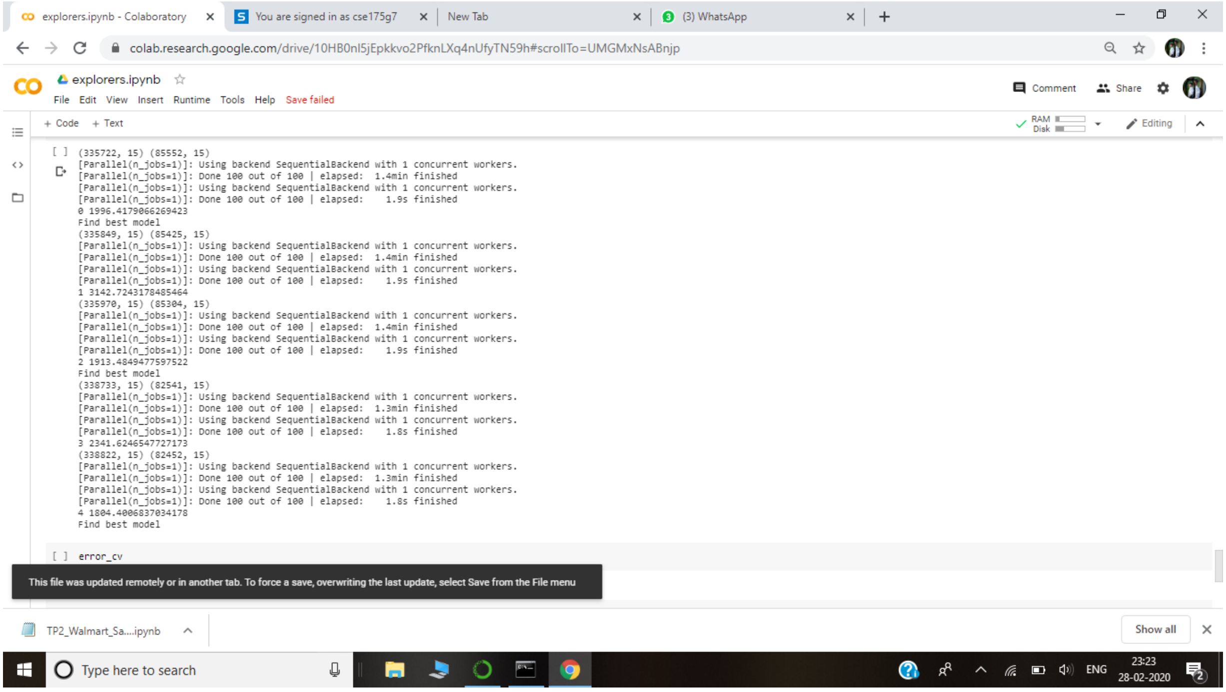
```
[ ] from numpy import *
    best_model = None
    error_cv = 0
    best_error = iinfo(int32).max
    for fold in range(5):
        dataset_train = splited.loc[splited['fold'] != fold]
```

This file was updated remotely or in another tab. To force a save, overwriting the last update, select Save from the File menu

```
[ ] from numpy import *
    best_model = None
    error_cv = 0
    best_error = iinfo(int32).max
    for fold in range(5):
        dataset_train = splited.loc[splited['fold'] != fold]
        dataset_test = splited.loc[splited['fold'] == fold]
        train_y = dataset_train['weeklySales']
        train_x = dataset_train.drop(columns=['weeklySales', 'fold'])
        test_y = dataset_test['weeklySales']
        test_x = dataset_test.drop(columns=['weeklySales', 'fold'])
        print(dataset_train.shape, dataset_test.shape)
        predicted, model = train_and_predict(train_x, train_y, test_x)
        weights = test_x['isHoliday'].replace(True, 5).replace(False, 1)
        error = calculate_error(test_y, predicted, weights)
        error_cv += error
        print(fold, error)
        if error < best_error:
            print('Find best model')
            best_error = error
            best_model = model
    error_cv /= 5
```

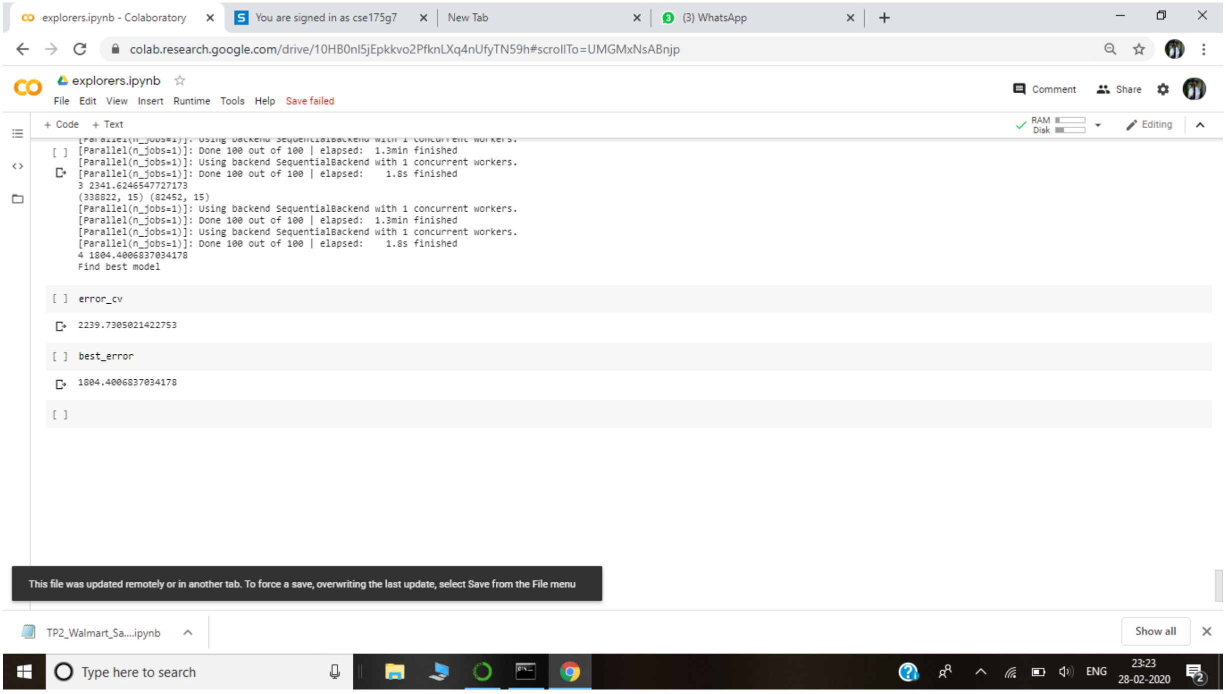
```
(335722, 15) (85552, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.4min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.9s finished
0 1996.4179066269423
Find best model
```

This file was updated remotely or in another tab. To force a save, overwriting the last update, select Save from the File menu



```
(335722, 15) (85552, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.4min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.9s finished
0 1996.4179066269423
Find best model
(335849, 15) (85425, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.4min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.9s finished
1 3142.7243178485464
(335970, 15) (85304, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.4min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.9s finished
2 1913.4849477597522
Find best model
(338733, 15) (82541, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.3min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.8s finished
3 2341.6246547727173
(338822, 15) (82452, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.3min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.8s finished
4 1804.4006837034178
Find best model

[ ] error_cv
```

```
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.3min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.8s finished
3 2341.6246547727173
(338822, 15) (82452, 15)
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:  1.3min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:    1.8s finished
4 1804.4006837034178
Find best model

[ ] error_cv

2239.7305021422753

[ ] best_error

1804.4006837034178

[ ]
```

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion:

Sales forecasting is a pivotal part of the financial planning of business for any organization. It can be said as a self-assessment tool which uses the statistics of the past and the current sales in order to predict future performance.
This project dealt with the implementation of machine learning algorithms on the sales dataset and a comparative analysis was carried out to forecast sales and to determine the best algorithm. Random Trees was confirmed to be a very effective model in forecasting sales data This work shows that there are highly efficient algorithms to forecast sales in big, medium or small organizations, and their use would be beneficial in providing valuable insight, thus leading to better decision-making.

7.2 Future Work:

- Modifying date feature into days, month, weeks.
- The dataset includes special occasions i.e , black Friday, Labour day, etc. On these days people tend to shop more than usual days. So adding these as a feature to data will also improve accuracy to a great extent.
- it would include the Extra Trees model being developed to consider sparse promotional markdown data and moving holidays.
- It would also involve the fine-tuning of the hyperparameters of the models to improve the accuracy of prediction.
- Future work could also entail combining the models to produce an ensemble training model that could represent even the tiniest details present in the data.
- With the development of deep learning techniques, the results of this could be further improved in the near future through the use of more complex and multilayer ANNs.

8.Refferences

[1] Kulkarni, Vrushali Y., and Pradeep K. Sinha. "Random forest classifiers: a survey and future research directions." Int J Adv Comput 36.1 (2013): 1144-53.

[2] Friedman, Jerome H. "Stochastic gradient boosting." Computational Statistics & Data Analysis 38.4 (2002): 367-378.

[3] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." Machine learning 63.1 (2006): 3-42.

[4] Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. "Forecasting with artificial neural networks: The state of the art." International journal of forecasting 14.1 (1998): 35-62.

[5] Allende, Héctor, Claudio Moraga, and Rodrigo Salas. "Artificial neural networks in time series forecasting: A comparative analysis." Kybernetika 38.6 (2002): 685-707.

[6] Adebiyi, Ayodele Ariyo, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. "Comparison of ARIMA and artificial neural networks models for stock price prediction." Journal of Applied Mathematics (2014), Article ID 614342, 7 pages, 2014. doi:10.1155/2014/614342.

[7] James J. Pao, Danielle S. Sullivan, “Time Series Sales Forecasting”, Final Year Project, 2017. Accessed at http://cs229.stanford.edu/proj2017/finalreports/5244336.pdf

[8] Sun, Zhan-Li, et al. "Sales forecasting using extreme learning machine with applications in fashion retailing." Decision Support Systems 46.1 (2008): 411-419.

[9] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Boca Raton, FL: Chapman & Hall, 1984.

[10] Kaggle. “Walmart Sales Forecasting Data”. https://www.kaggle.com/c/walmart-salesforecasting/data

[11] Kaggle. “Walmart Sales Forecasting Leaderboard”. https://www.kaggle.com/c/walmart-salesforecasting/leaderboard

[12] Nikhil Elias, Sales Forecasting using ML algorithms (2018), GitHub repository. https://github.com/NikhilElias/SalesForecasting-using-MLalgorithms/blob/master/Code.py