**FORECASTING SALES OF STORE USING**

**IBM WATSON STUDIO**

**Mini Project Report**

Submitted By **(BATCH NO: CSE\_015)**

**GOWROJU LAXMIPRIYA 18UK1A0518**

**CHALLAKONDA SRICHANDAN 18UK1A0567**

**JUNNUTHULA AKSHAYA 18UK1A0523**

**MANDA MANASA 18UK1A0531**

**INDEX**

**TITLE PAGE NO**

**CHAPTER 1: INTRODUCTION 4**

* 1. OVERVIEW **4**
  2. PURPOSE **4**

**CHAPTER 2: LITERATURE SURVEY 5**

* 1. EXISTING PROBLEM **5**
  2. PROPOSED SOLUTION **5**

**CHAPTER 3: THEORTICAL ANALYSIS 6**

3.1 BLOCK DIAGRAM **6**

3.2 HARDWARE/SOFTWARE DESIGNING **6**

**CHAPTER 4: PROJECT FLOW 7**

**CHAPTER 5: DATA 8**

**CHAPTER 6: RESULTS 9,10**

**CHAPTER 7: APLLICATIONS 10**

**CHAPTER 8: CONCLUSION 10**

**CHAPTER 9: FUTURE SCOPE 10**

**CHAPTER 10: BIBILOGRAPHY 10**

**CHAPTER 11: APPENDIX 11**

**ABSTRACT**

Sales forecasting is an essential task for the management of a store. Being able to estimate the number of products that a retail store is going to sell in the future will allow the owners of these shops to prepare the inventory that they will need.

In this project, we are building a system that analyses the previous trends of sales which includes sales on various days and predicts future sales. The goal of this project is to forecast the sales of stores by using time series analysis. Here time series analysis algorithms such as RNN (Recurrent Neural Network) & LSTM (Long Term Short Memory) are used to analyze the past trends of sales of stores. Create and deploy flask-based web Application and integrate AI model to it.

The objective of the project is to build a web application where the user gives the last ten days' sales values and gets the prediction for the 11th day which is showcased on UI.

1. **INTRODUCTION**
   1. OVERVIEW

Sales forecasting is the process of using a company’s sales records over the past year’s or past days to predict the short term or long terms sales performance in the future . This is one of the pillars of proper financial planning.

One of the most common yet basic challenges that the management of companies face in making business sales forecasts is that their usual approach is a “top to down “one . This approach leaves very little scope for interaction with the sales manager and the salespersons during the data collection process.

Sales forecasting is an essential task for the management of a store. Being able to estimate the number of products that a retail store is going to sell in the future will allow the owners of these shops to prepare the inventory that they will need.

In this project, we are building a system that analyses the previous trends of sales which includes sales on various days and predicts future sales. The goal of this project is to forecast the sales of stores by using time series analysis. Here time series analysis algorithms such as RNN (Recurrent Neural Network) & LSTM (Long Term Short Memory) are used to analyze the past trends of sales of stores. Create and deploy flask-based web Application and integrate AI model to it.

The objective of the project is to build a web application where the user gives the last ten days' sales values and gets the prediction for the 11th day which is showcased on UI.

* 1. PURPOSE
* Understand the historical demand for the products in the last 11 days.
* Analyse the 11th day sales of the store.
* Use the time series forecasting.
* Create a web based UI to produce forecasts.

1. **LITERATURE SURVEY**
   1. EXISTING PROBLEM

We know that, for every store, sale is one of the most important factors as it is directly proportionate with store profits and losses.

Hence most store owners are interested in forecast of the sales of stores .Previously, it used to be done by manual processes using balance sheets and spread sheets.

Sales forecast team used to study and analyze old statistics and reports and by certain mathematical calculations and through some complex algorithms they try to find out future predictive sale of a store or a company.

This type of manual approach could not able to provide desired exact accuracy and mostly it is fail during stating quarters. To check accuracy , higher management has to wait for last quarter sales and then only they can able to track whatever report generated by sale forecast team is accurate or not . Also it is a quite leanly process.

* 1. PROPOSED SOLUTION

To overcome the drawbacks of manual sales forecast team’s results, we have

Proposed one deep learning based sales forecasting system. This system can able to provide more than 95% accuracy in the sales forecasting of a store or a company.

This system can able to generate whole sales reports with requirement of necessary manpower and other resources. It means that by using this sales forecasting system, one can easily find out that to achieve particular sales target , how much manpower and how much resources are required.

1. **THEORETICAL ANALYSIS**
   1. BLOCK DIAGRAM

Diagram

Description automatically generated

* 1. HARDWARE/SOFTWARE DESIGNING

To complete this project you should have the following software  and packages.

* Anaconda Navigator

Jupyter notebook

Spyder IDE

* Tensor flow
* Keras
* Notepad

1. **PROJECT FLOW**

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

* Data Collection.
  + Collect the dataset or Create the dataset
* Data Preprocessing.
  + Import the Libraries.
  + Importing the dataset.
  + Analyze the data
  + Taking care of Missing Data
  + Feature Scaling
  + Data Visualization
  + Splitting Data into Train and Test.
  + Creating a dataset with a sliding window.
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Adding LSTM Layers
  + Adding Output Layer
  + Configure the Learning Process
  + Training the model
  + Model Evaluation
  + Save the Model
  + Test the Model
* Application Building
  + Create an HTML file
  + Build Python Code

1. **DATA**

Deep Learning depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes neural network training possible. In Deep Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

* Our Sales dataset contains

1. HQ - This attribute represents HeadQuarters of the company.

2. Country - This attribute represents the country of the outlet.

3. State\_of\_outlet - This attribute represents the state of the outlet.

4. City\_of\_outlet - This attribute represents the city of the outlet.

5. Month - This attribute represents the month(ranging from 1 to 12 )

6. Day - This attribute represents a day of the month(ranging from 1 to 31)

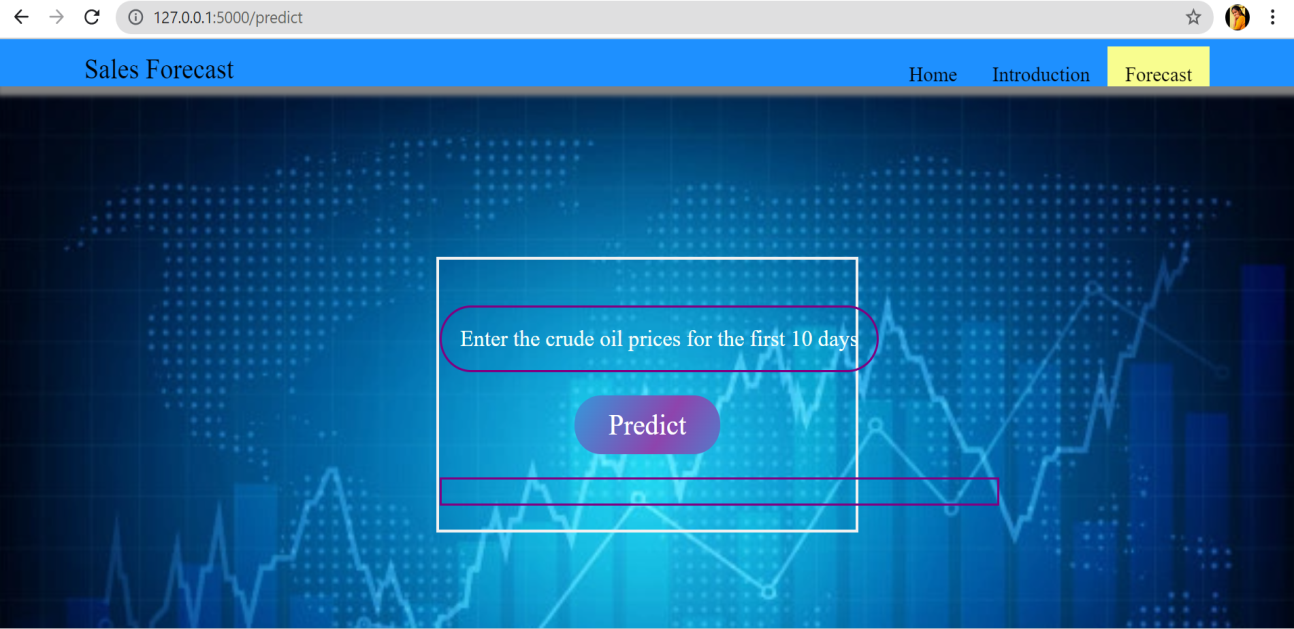
7. Year - This attribute represents the year(ranging from 2005 to 2016)

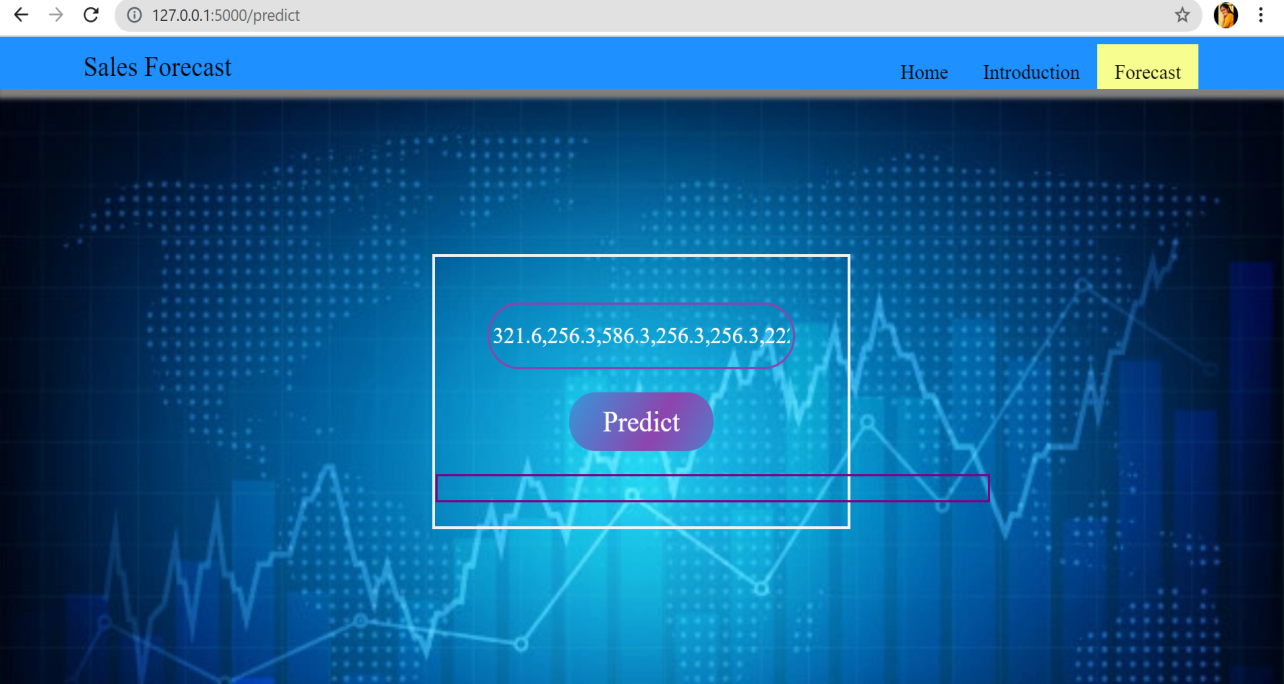
8. Total\_Sales - Target variable (total sales on a particular day). It contains total sales from 19886 to 2018.

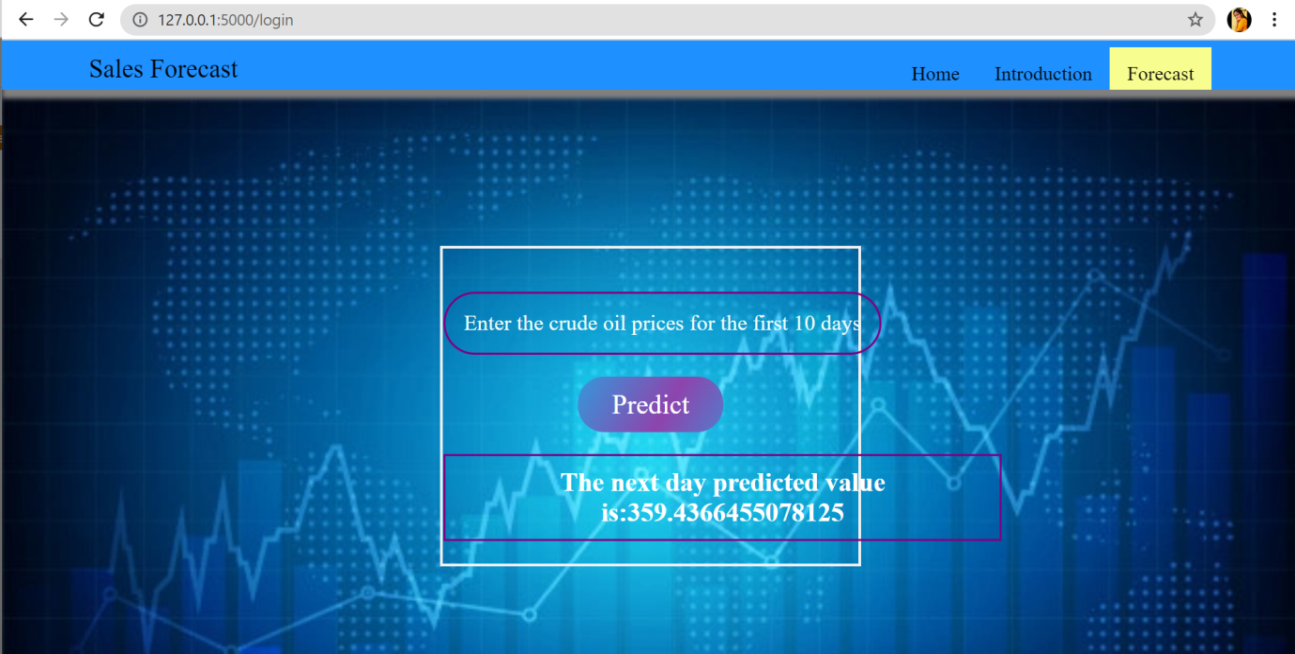
A close-up of a document

Description automatically generated with low confidence

1. **RESULTS**

****

****



1. **APPLICATIONS**

* Forecasting has applications in a wide range of fields where estimates of future conditions are useful. Not everything can be forecast reliably, if the factors that relate to what is being forecast are known and well understood and there is a significant amount of data that can be used very reliable forecasts can often be obtained. If this is not the case or if the actual outcome is affected by the forecasts, the reliability of the forecasts can be significantly lower.
* Forecasting has also been used to predict the development of conflict situations.

1. **CONCLUSION**

* In this project we have done the prediction of 11th day sales of a store after entering the before 10 days sales data
* Prediction is done using LSTM and RNN algorithms

1. **FUTURE SCOPE**

Sales forecasting is an essential task for the management of a store. Being able to estimate the number of products that a retail store is going to sell in the future will allow the owners of these shops to prepare the inventory that they will need.

1. **BIBLIOGRAPHY**

* <https://www.geeksforgeeks.org/>
* <https://www.w3schools.com/>
* <https://stackoverflow.com/>
* <https://www.lucidchart.com/pages/>
* [https://smartinternz.com/Student/guided\_project\_info/4652#](https://smartinternz.com/Student/guided_project_info/4652)

1. **APPENDIX**

**intro.html**

<html>

<style>

.header { position: relative;

top:0;

margin:0px;

z-index: 1;

left: 0px;

right: 0px;

position: fixed;

background-color: #1E90FF ;

color: white;

box-shadow: 0px 8px 4px #FAFAFA;

overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: center;

}

.topnav {

overflow: hidden;

background-color: #ECFAC0;

}

.topnav-right a {

float: left;

color: black;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 18px;

}

.topnav-right a:hover {

background-color: #ECFAC0;

color: black;

}

.topnav-right a.active {

background-color: #ECFAC0

color: white;

}

.topnav-right {

float: right;

padding-right:100px;

}

body {

background-color: ;

background-repeat: no-repeat;

background: #310404 url(https://i.pinimg.com/originals/2e/e6/99/2ee6998e34c3e2eff7b894c66cfc5267.jpg) no-repeat centercenter fixed;

background-size:cover;

background-position: 0px 0px;

}

.button {

background-color: #091425;

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 12px;

border-radius: 16px;

}

.button:hover {

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);

}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

display: inline-block;

margin-bottom:18px;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #091425;

color: white;

padding: 14px 20px;

margin-bottom:10px;

border: none;

cursor: pointer;

width: 17%;

border-radius:4px;

font-family:Montserrat;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}

img.avatar {

width: 30%;

border-radius: 50%;

}

.container {

padding: 16px;

}

span.psw {

float: right;

padding-top: 16px;

}

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}

.home{

margin:80px;

width: 84%;

height: 500px;

padding-top:10px;

padding-left: 30px;

}

.login{

margin:80px;

box-sizing: content-box;

width: 84%;

height: 420px;

padding: 30px;

border: 10px solid blue;

}

.left,.right{

box-sizing: content-box;

height: 400px;

margin:20px;

border: 10px solid blue;

}

.mySlides {display: none;}

img {vertical-align: middle;}

.slideshow-container {

max-width: 1000px;

position: relative;

margin: auto;

}

.text {

color: #f2f2f2;

font-size: 15px;

padding: 8px 12px;

position: absolute;

bottom: 8px;

width: 100%;

text-align: center;

}

.dot {

height: 15px;

width: 15px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active {

background-color: #F8FD8F;

}

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 1.5s;

animation-name: fade;

animation-duration: 1.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@media only screen and (max-width: 300px) {

.text {font-size: 11px}

}

@importurl(https://fonts.googleapis.com/css?family=Anonymous+Pro);

html{

min-height: 100%;

overflow: hidden;

}

body{

height: calc(100vh - 8em);

padding: 4em;

color: rgba(255,255,255,.75);

font-family: 'Anonymous Pro', monospace;

background-color: black;

}

.line-1{

position: relative;

top: 50%;

width: 200em;

margin: 0 auto;

border-right: 2px solid rgba(255,255,255,.75);

font-size: 180%;

text-align: center;

white-space: nowrap;

overflow: hidden;

transform: translateY(-50%);

}

.anim-typewriter{

animation: typewriter 20s steps(200) 1s 1 normal both,

blinkTextCursor 500ms steps(200) infinite normal;

}

@keyframes typewriter{

from{width: 0;}

to{width: 200em;}

z-index: -1;

}

@keyframesblinkTextCursor{

from{border-right-color: rgba(255,255,255,.75);}

to{border-right-color: transparent;}

}

</style>

<body>

<p class="line-1 anim-typewriter"> Let us build a UI where we get future predictions of Sales</p>

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:black; padding-top:1%;padding-left:5%;">Sales Forecast</div>

<div class="topnav-right"style="padding-top:0.5%;">

**app.py**

import numpy as np

from flask import Flask,render\_template,request

from tensorflow.keras.models import load\_model #we are loading our model from keras

import pickle

app = Flask(\_\_name\_\_) #our flask app

scaler=pickle.load(open("Scaler\_forecast.pkl","rb"))

model = load\_model('sales\_forecast.h5') #loading the model in the flask app

@app.route('/') #rendering html template

def home():

return render\_template("home.html") #rendering html template

@app.route('/intro')

def home1() :

return render\_template("intro.html") #rendering html template

@app.route('/predict')

def home2() :

return render\_template("web.html") #rendering html template

@app.route('/login',methods = ['POST']) #route for our prediction

def login() :

x\_input=str(request.form['year']) #requesting the file

x\_input=x\_input.split(',')

print(x\_input)

for i in range(0, len(x\_input)):

x\_input[i] = float(x\_input[i])

print(x\_input)

x\_input=np.array(x\_input).reshape(1,-1)

temp\_input=list(x\_input)

temp\_input=temp\_input[0].tolist()

lst\_output=[]

n\_steps=10

i=0

while(i<10):

if(len(temp\_input)>=10):

#print("temp\_input",temp\_input)

x\_input=np.array(temp\_input[0:])

print("{} day input {}".format(i,x\_input))

x\_input=np.expand\_dims(x\_input, axis=0)

x\_input=scaler.transform(x\_input)

#x\_input=x\_input.reshape(1,-1)

x\_input = x\_input.reshape((1, n\_steps, 1))

#print("x\_input.....",x\_input)

yhat = model.predict(x\_input, verbose=0)

yhat=scaler.inverse\_transform(yhat)

print("{} day output {}".format(i,yhat))

temp\_input.extend(yhat[0].tolist())

temp\_input=temp\_input[1:]

#print(temp\_input)

lst\_output.extend(yhat.tolist())

i=i+1

else:

print("Please give 10 number of inputs")

return render\_template("web.html",showcase = 'The next day predicted value is:'+str(lst\_output[0][0]))

if \_\_name\_\_ == '\_\_main\_\_' :

app.run(debug = False,port=5000)