**Time Series Analysis and Sales Forecasting for Automotive Using IBM Cloud Services**

**A PROJECT REPORT**

**Submitted by**

**Team No. 1**

Shiv Taneja

(19BAI10039)

Hitesh Agrawal

(19BAI10030)

Krishan Kumar Gupta

(19BAI10114)

Shrutika Nikhar

(19BCY10107)

**1. INTRODUCTION**

**a. OVERVIEW**

Time-series Sales forecasting has proved to be a crucial topic in every business, helping to process data taken over a long period of time. Stock exchange, logistics, retail are classic industries where the ability to build predictive models becomes a crucial differentiator in a highly-competitive business environment. With our project we have tried to apply time series analysis and sales forecasting method for Automotive using IBM cloud services: finding hidden patterns, detecting trends in sales over the years, and predicting sales in the future.

**b. PURPOSE**

Unknown Demand for vehicles is a significant problem that decreases the productivity of the production environment. Thus, forecasting plays a vital role in planning the production and inventory of an industry. Time series forecasting is a type of forecasting in which last period data is used to determine the industry's trends and sales in the future.

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 analyse 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.

**2 LITERATURE SURVEY**

**2.1. EXISTING PROBLEM**

1. Seher Arslankya in his research paper implemented time series analysis and artificial neural network methods to estimate sales for future months for the leading company in the automotive industry in Turkey. He examined the monthly data between January 2011 and July 2016 using multiple regression, moving average and artificial neural network model. Furthermore, he compared MAPE values for all models, which resulted in the ANN model giving the best result.
2. Arnis Kirshners did a comparative analysis of short time series processing methods intending to scrutinize these methods' ability to be used when analysing short time series. The author has analysed the moving average Method, exponential smoothening, and exponential smoothening with development trends resulting from moving average having the smallest squared error value but with large forecast smoothening for initial data.
3. Shamsul Masum elucidates on time series forecasting and its classification and approaches and strategies of time series forecasting. Furthermore, the author has demonstrated how an inappropriate point to point rolling forecast strategy leads to unrealistic outcomes and supports his argument with a comparative analysis of two strategies using the ARIMA model. The author has concluded with the result of rolling single point outcome being deceptive for Euro Dollar Exchange Rates case considered.
4. Tamal Datta Chaudhuri proposed six different forecasting methods for predicting the time series index of the healthcare sector. The author has demonstrated a decomposition approach of time series for data from January 2010 to December 2016 and illustrated how the decomposition results provide us with useful insights into the behaviour and properties exhibited by time series. The author observed that results from the ARIMA model with a horizon of 12 months came out to be the best model with the lowest RSME value, while the Holt Winters method with a horizon of 12 months has the highest RSME value.
5. Jaydip Sen, in his research paper, uses the Time series - decomposition based Method to analyse the past of the Indian realty sector and predict its future. He uses time series forecasting methods in R programming language to determine future results. He uses time series index value data of the Indian realty sector for six years from 2010-2016 month wise. The methods used for accurate predictions are: Holt Winters exponential smoothening and Autoregressive Integrated Moving Average. He analyses the results from the above-mentioned broad concepts and observes, which is the best one. With the result obtained, he argues that these can be immensely useful for portfolio managers and stock traders to buy or sell stocks at the correct time.
6. Samita sood elaborates how time series forecasting can be deployed in determining the future development of the Indian tourism industry from past secondary data. The author uses the data for the number of tourists in India from 1980 to 2020 and uses models such as ARIMA and Holt Winter to forecast foreign tourist travel and compare the two models' accuracy. On the basis of MAPE and RMSE, the authors conclude that Holt Winter is a more accurate model than ARIMA in this particular situation.
7. William R Huss implements univariate estimation techniques such as Holt winter exponential smoothing, Multiple regression, Linear regression to study the load on 49 largest electric utilities in the United States and forecast the load for future planning. The author uses electric utility data from 1972 to 1982, and the results indicate that for shorter periods, Holt Winters Exponential smoothing method is highly accurate, and for more extended periods, extrapolation of Linear regression horizons proves to be efficient.
8. Deepa, in this paper, reviews, and forecasts the Indian Motorcycle market using Time series forecasting. The author uses SARIMA (Seasonal autoregressive Integrated Moving Method) and Holt Winters Method for prediction. The author compares several years' data and uses MAE and MAPE method to determine which model is more accurate and concludes that both the models are pretty significant but Holt Winters method is numerically more precise than the SARIMA model. According to the author the studies can be further enhanced by using more such models.

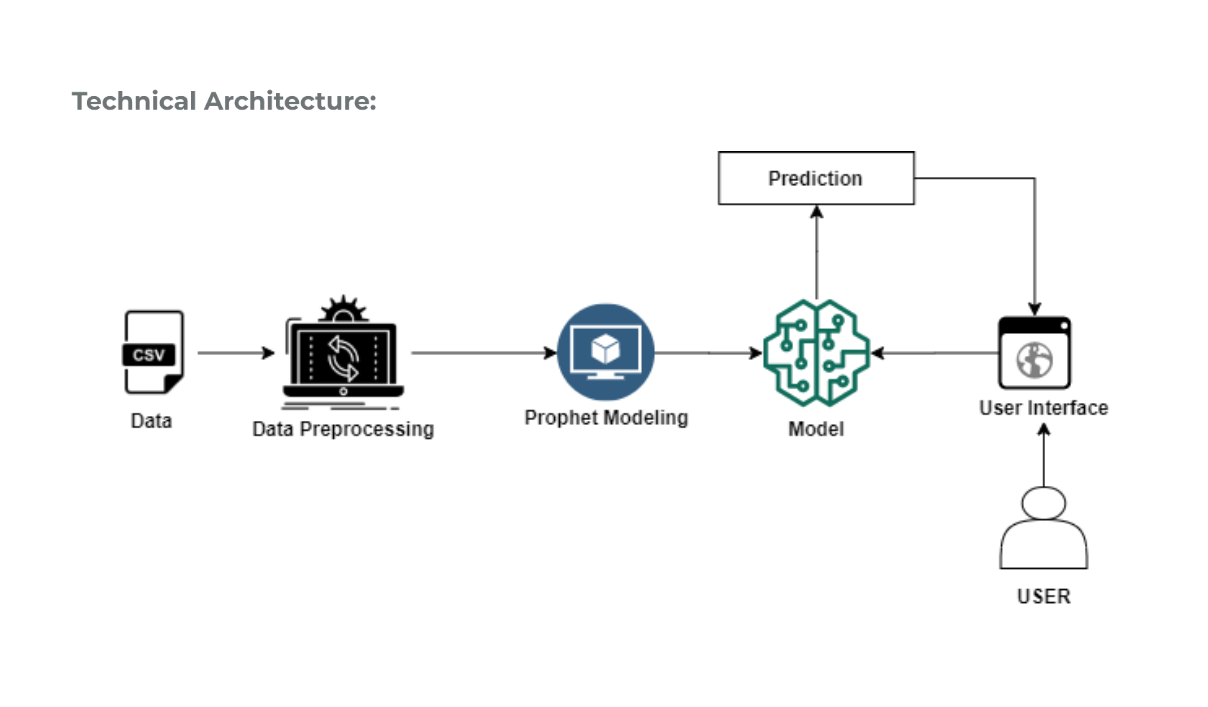
**2.2. PROPOSED SOLUTION**

The automotive sector is one of the developing sectors, where the most massive investments are made. It has an extensive business volume. Indian automobile production increased at 2.63% CAGR between FY16-FY20, with 26.36 million vehicles being manufactured in the country in FY20. Overall, domestic automobile sales increased at 1.29% CAGR between FY16-FY20, with 21.55 million vehicles being sold in FY20. It contributes significantly to the economic development of countries in the world. Today India’s automotive industry is expected to reach Rs 16 to 18 trillion by 2026. Therefore, it is crucial for companies in this sector to correctly manage their resources. To do that, companies should predict the future in the best possible way and anticipate possible issues. Automotive sectors are the critical sector of the Indian economy. There is a dire need for demand forecasting for production planning and predicting future trends in the industry. Demand forecasting plays a significant role in decision making for the upliftment of business. For the past several years, there has been immense development in techniques used for demand forecasting, and new and improved methods are being developed for more accurate results.

**3. THEORITICAL ANALYSIS**

**3.1. BLOCK DIAGRAM**

Forecasting or predicting the sale value helps the investors to invest in such a time where profits can be maximum. This project provides guidance to individuals who are willing to invest or buy a car and help them in knowing the price of a day using the prophet library. Time series analysis is made on the data for accurate predictions.



**3.2. HARDWARE / SOFTWARE DESIGNING**

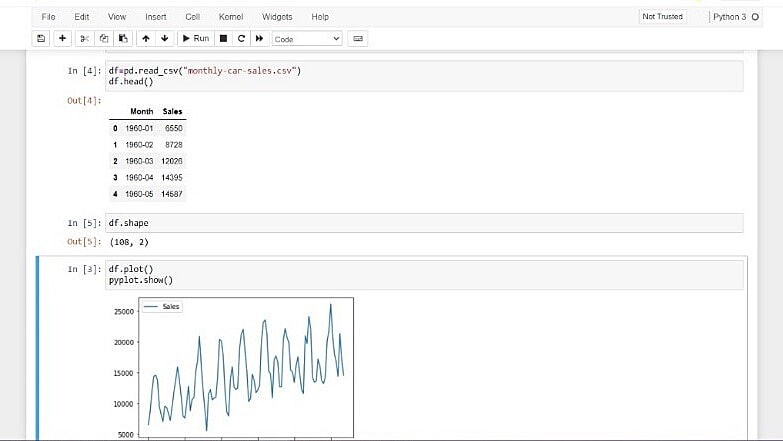
**Software requirements**

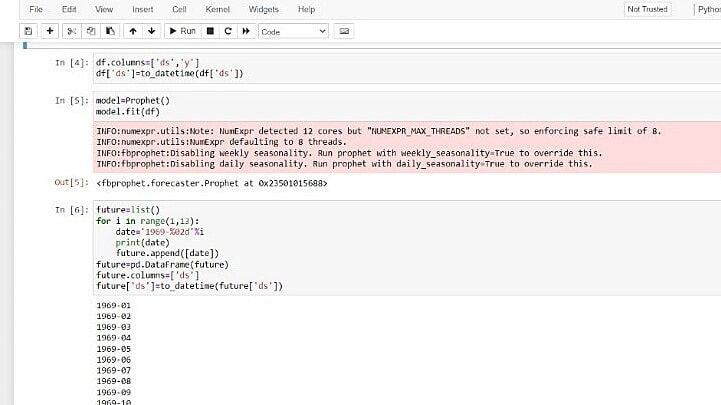
* Anaconda – Python, Jupyter notebook, Spyder
* Windows/ Linux/ Mac os
* Tensorflow
* Opencv
* Numpy
* FbProphet
* IBM cloud

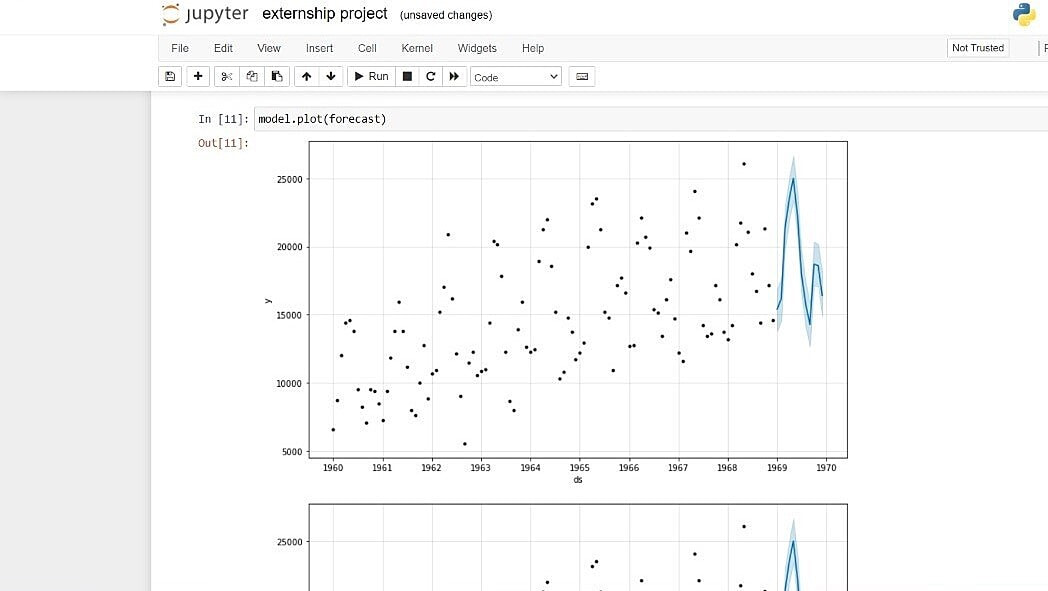
**Hardware requirements**

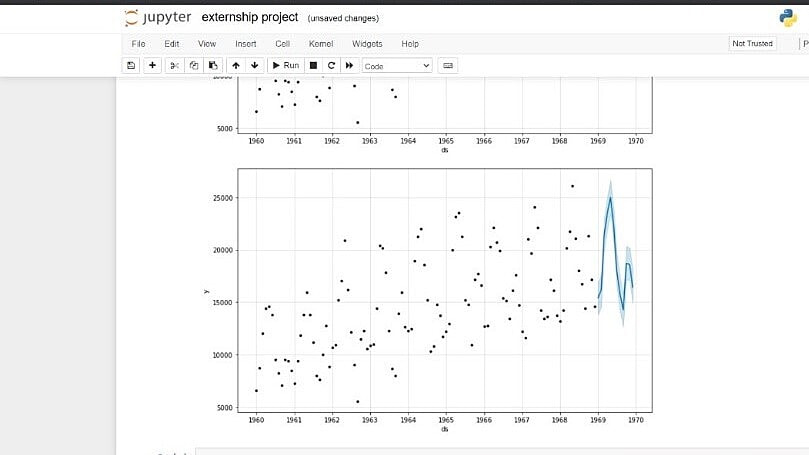
* TPU
* Processor: Minimum 1 GHz
* RAM: Minimum 8 Gb
* Intel core 5 or 7 or amd 5

**4. EXPERIMENTAL INVESTIGATIONS**

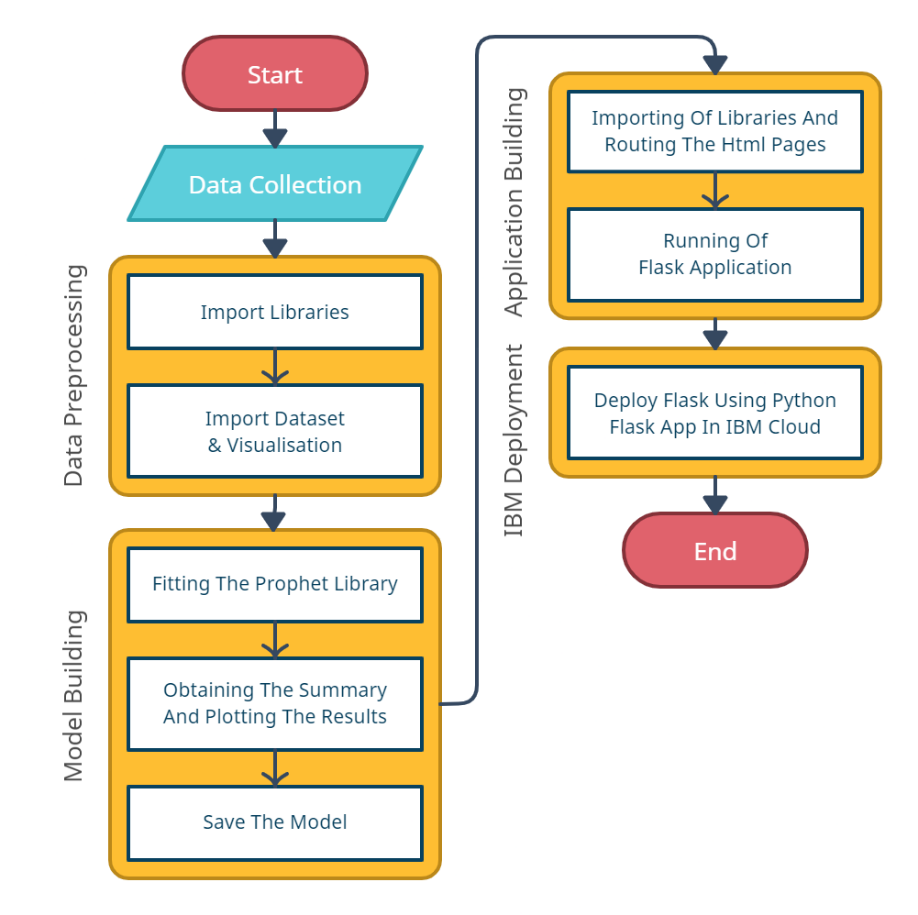






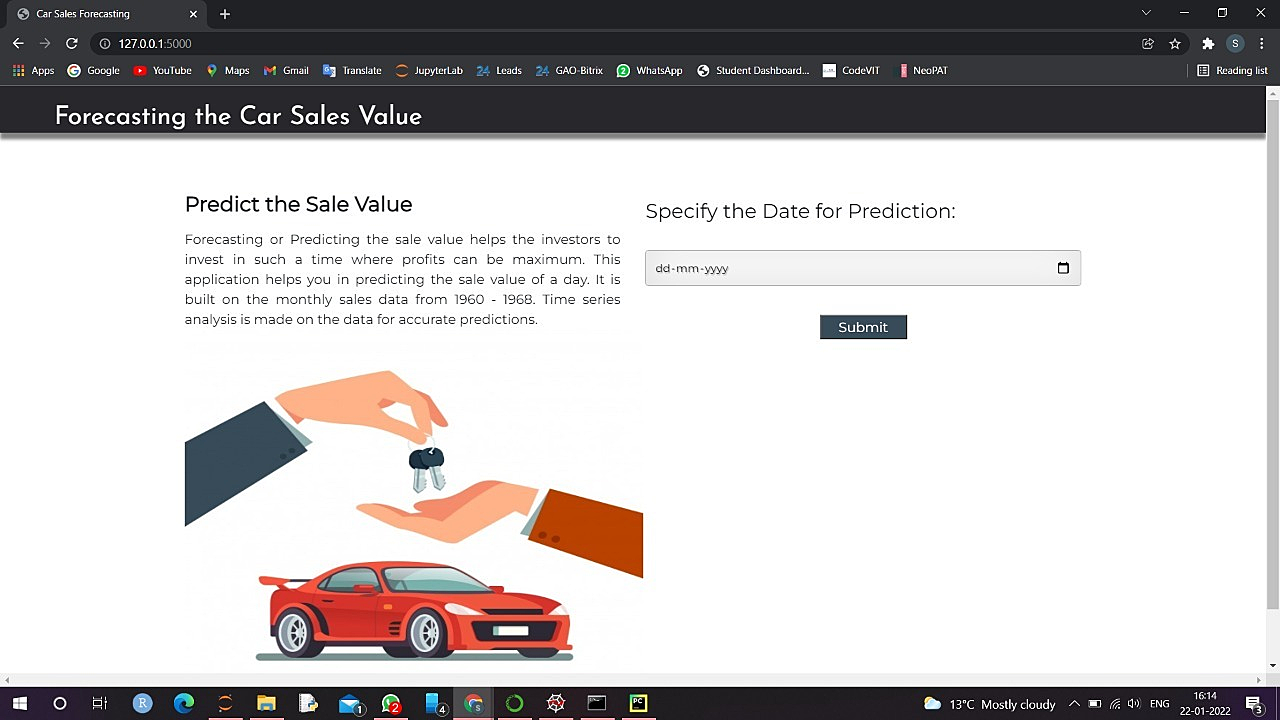


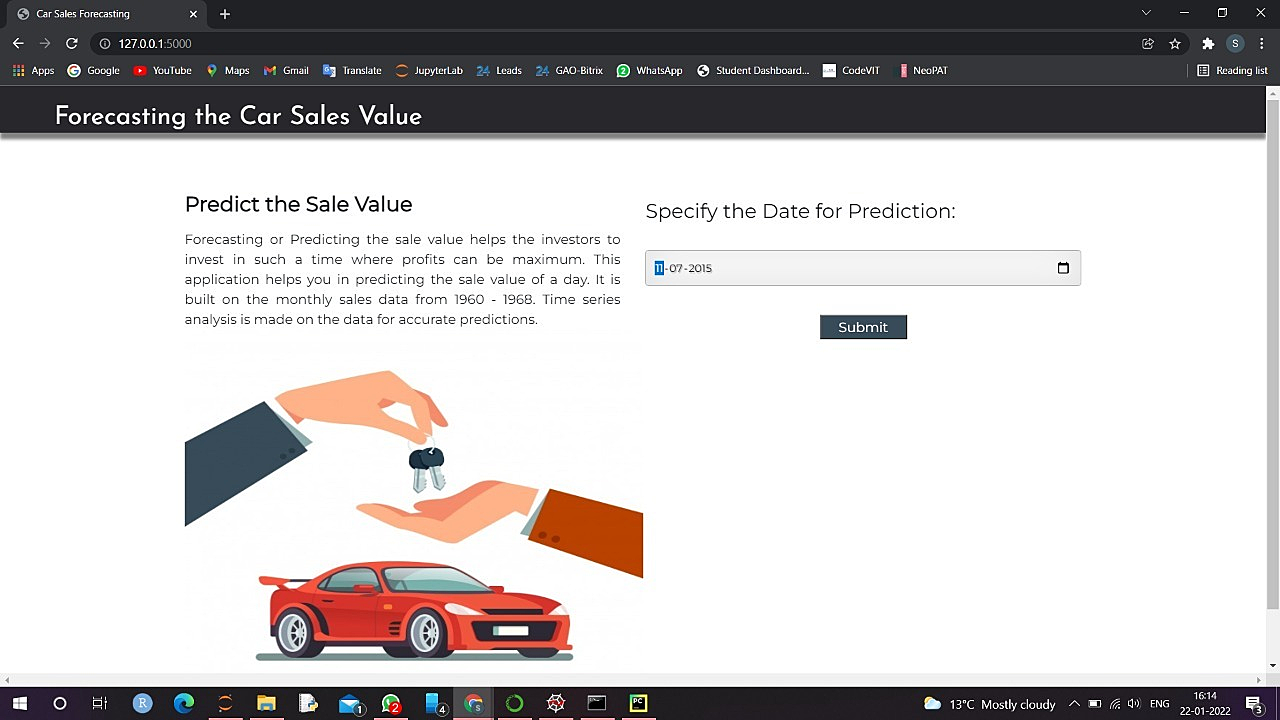
**5. FLOWCHART**

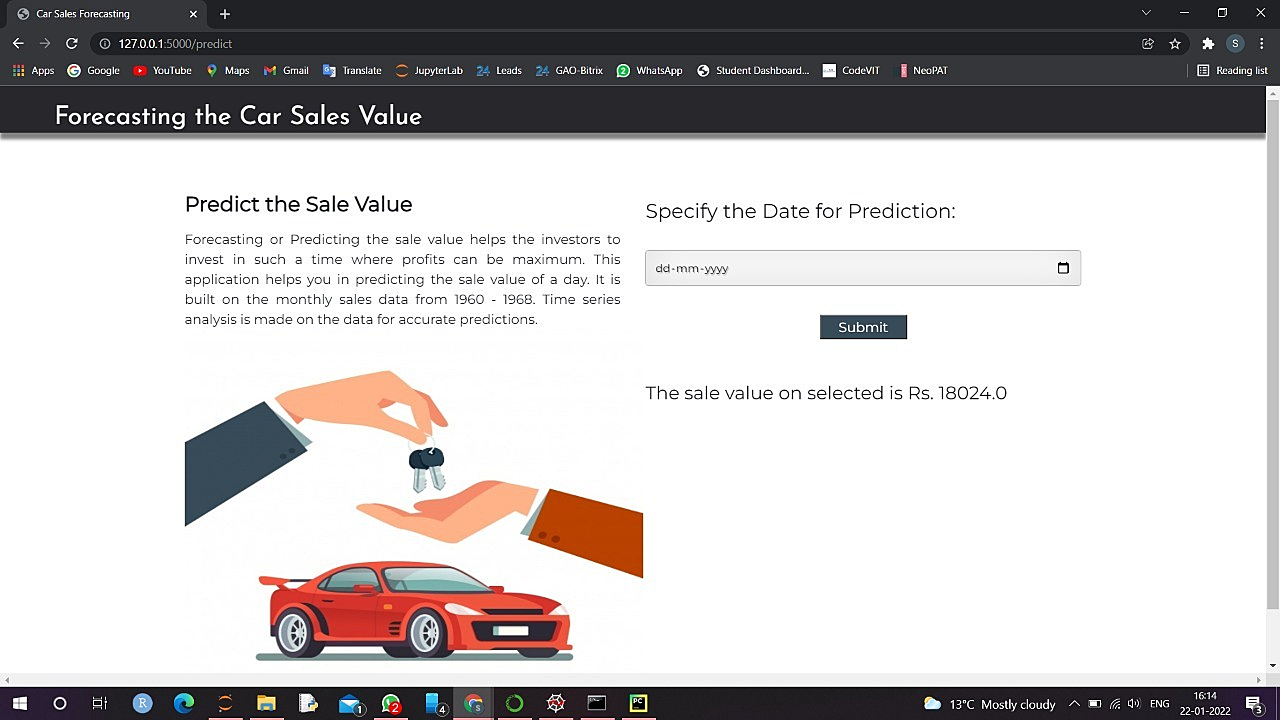


**6. RESULT**

Forecasting or predicting the sale value helps the investors to invest in such a time where profits can be maximum. This project guides individuals who are willing to invest or buy a car and helps them in knowing the price of a day using the prophet library. Time series analysis is made on the data for accurate predictions. The final findings we got here are when the user inputs the date for prediction and submits it, we get the sale value of that particular in Rupees.







**7. ADVANTAGES & DISADVANTAGES**

Our solution for the Time Series Analysis and Sales Forecasting for Automotive have following advantages and disadvantages.

**Advantages:**

1. Time series analysis helps in identifying the patterns and also creates the opportunity to clean your data. It gives the high accuracy and provides simplicity in executing.
2. It is a really handy tool for forecasting purposes. It gives accurate predictions for future values, but it also requires more skill than regression analysis since you need to adapt your model according to the historical data.
3. It may be helpful to see how the commodity, security, or economic component shifts over time. The model is also used to analyze the possible changes in car resale value as many insurance companies depends upon this.

**Disadvantages:**

1. Time series analysis is useful for short-term forecasting, but it could sometimes lead to wrong predictions. This is because it requires historical data in order to construct the models, which means that if some significant changes occurred over time, then those changes will not be included within the forecasted periods.
2. Our model has been built on historical data, so it cannot be used to predict future values or trends that are too far in Time.
3. The model could be over-fitted, which means that due to some random factors, a model can lead to false results.

**8. APPLICATIONS**

By using our solution for the Time Series Analysis and Sales Forecasting for Automotive we can do the following things:

1. A potential buyer of a used car will get a range of car resale value. So that one can buy the used car according to the range.
2. In terms of insurance, the Insured Declared Value (IDV) plays a major role. The term ‘IDV’ refers to the maximum claim, an insurer will pay if the vehicle is damaged beyond repair or is stolen.
3. Manufacturing companies will get a basic resale value range so that they can set the pricing of upcoming cars in such a way that the resale value remains high than the previous year's data of existing cars.

**9. CONCLUSION**

The overall purpose of the study was to prove that it’s possible to efficiently forecast car sales using a simple statistical model. During our research we were able to prove that the Decision Tree Regressor based approach has acceptable outcomes. Such models can be easily implemented with various statistical software and their computational complexity is acceptable. Also, the approach has well-studied statistical properties.

The accuracy of the predictive model for car sales forecast obtained is 87.9%. Hence it has been proved that the percentage error is not greater than 12.1 % for each of the 12 months ahead. Obviously, the accuracy of the model is high enough and the model can be used as a baseline for developing better models. The method is well suited for use in different business domains.

**10. FUTURE SCOPE**

Our only concern about the project is that It can only extract linear relationships within the time series data. Predictions generated may not be suitable for complex nonlinear cases. It does not efficiently extract the full relationship hidden in the data. Another limitation is that the model requires a large amount of data to generate accurate predictions. Hence these above are the two current issues that could be addressed for prospects.

**11. BIBLOGRAPHY**

* Project Description - <https://smartinternz.com/saas-guided-project/1/forecasting-sales-of-store-using-ibm-watson-studio>
* Importance of Time-series [Business Sector] - <http://home.ubalt.edu/ntsbarsh/stat-data/forecast.htm>
* Insight into Time-series Analysis - <http://www.jetir.org/papers/JETIR2107494.pdf>
* ExistingWorks: -<https://www.researchgate.net/publication/353659148_Analysis_of_Time_Series_Forecasting_Techniques_for_Indian_Automotive_Industry>
* Practical Implementation and Insights - <https://mindcraft.ai/concepts/time-series-analysis-and-sales-forecasting-for-automotive/>

**APPENDIX**

* Time-series forecasting using SARIMA-based approach – <https://mindcraft.ai/concepts/time-series-analysis-and-sales-forecasting-for-automotive/>
* IBM academic services - <https://www.youtube.com/watch?v=x6i43M7BAqE>

**SOURCE CODE: <https://github.com/smartinternz02/SI-GuidedProject-7194-1640667159>**

# coding: utf-8

**# # STEP 1: DATA COLLECTION**

**# # STEP 2: DATA PREPROCESSING/ DATA WRANGLING**

**# IMPORTING LIBRARIES**

import numpy as np

import pandas as pd

import os, types

import pandas as pd

from botocore.client import Config

import ibm\_boto3

def \_\_iter\_\_(self): return 0

if os.environ.get('RUNTIME\_ENV\_LOCATION\_TYPE') == 'external':

endpoint\_9d750cc572a946a1aa22eeb94a45c58f = 'https://s3.us.cloud-object-storage.appdomain.cloud'

else:

endpoint\_9d750cc572a946a1aa22eeb94a45c58f = 'https://s3.private.us.cloud-object-storage.appdomain.cloud'

client\_9d750cc572a946a1aa22eeb94a45c58f = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='XKj2TQISLTaxfuxxbVQnH0awhq-oqSsqxZY\_maa57ysq',

ibm\_auth\_endpoint="https://iam.cloud.ibm.com/oidc/token",

config=Config(signature\_version='oauth'),

endpoint\_url=endpoint\_9d750cc572a946a1aa22eeb94a45c58f)

body = client\_9d750cc572a946a1aa22eeb94a45c58f.get\_object(Bucket='carseries-donotdelete-pr-yllvezhwcxyq4x',Key='monthly-car-sales.csv')['Body']

# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object

if not hasattr(body, "\_\_iter\_\_"): body.\_\_iter\_\_ = types.MethodType( \_\_iter\_\_, body )

dataset = pd.read\_csv(body)

dataset.head()

dataset.tail()

dataset.head(10)

dataset['year'] = pd.DatetimeIndex(dataset['Month']).year

dataset['month'] = pd.DatetimeIndex(dataset['Month']).month

#dataset['day'] = pd.DatetimeIndex(dataset['Date']).day

dataset.head()

dataset.corr()

# Dropping Date Column, Because we already splitted the Date into Year, Month and Day

dataset.drop('Month', axis=1, inplace=True)

# Checking for null values

dataset.isnull().any()

dataset.info()

# HANDLING MISSING VALUES

import matplotlib.pyplot as plt

plt.bar(dataset['month'],dataset['Sales'],color='green')

plt.xlabel('Month')

plt.ylabel('y')

plt.title('PRICE OF CAR SALES ON THE BASIS OF MONTHS OF A YEAR')

plt.legend()

import seaborn as sns

sns.lineplot(x='year',y='Sales',data=dataset,color='red')

fig=plt.figure(figsize=(8,4))

plt.scatter(dataset['year'],dataset['Sales'],color='purple')

plt.xlabel('Month')

plt.ylabel('Price')

plt.title('PRICE OF CAR SALES ON THE BASIS OF MONTHS OF A YEAR')

plt.legend()

import seaborn as sns

sns.pairplot(dataset)

plt.show()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

dataset['colum\_name'] = le.fit\_transform(dataset['column\_name']

# Now, Split the dataset into X(independent variable) and Y(dependent variable)

x=dataset.iloc[:,1:3].values #inputs

y=dataset.iloc[:,0:1].values #output price only

x

y

**# SPLIT THE DATA INTO TRAIN AND TEST SETS**

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,

test\_size=0.2,random\_state=0)

x\_train.shape

x\_test.shape

**# # STEP 4: BUILDING AND TESTING THE MODEL**

# MULTIPLE LINEAR REGRESSION

#importing linear regression from scikit learn library

from sklearn.linear\_model import LinearRegression

#mlr is object of LinearRegression

mlr=LinearRegression()

#trainig the model using fit method

mlr.fit(x\_train,y\_train)

y\_pred=mlr.predict(x\_test)

y\_pred

y\_test

from sklearn.metrics import r2\_score

accuracy=r2\_score(y\_test,y\_pred)

accuracy

# DECISION TREE REGRESSOR

#import decision tree regressor

from sklearn.tree import DecisionTreeRegressor

dtr=DecisionTreeRegressor()

#fitting the model or training the model

dtr.fit(x\_train,y\_train)

# PREDICTION

y\_pred=dtr.predict(x\_test)

y\_pred

y\_test

**# ACCURACY EVALUATION**

from sklearn.metrics import r2\_score

dtraccuracy=r2\_score(y\_test,y\_pred)

dtraccuracy

**# RANDOM VALUE PREDICTION**

dataset.head()

y\_p=dtr.predict([[2005,12]])

y\_p

y\_p=dtr.predict([[1997,1]])

y\_p

import pickle

pickle.dump(dtr,open('sales.pkl','wb'))

pwd

**#Deployment**

**Import and Install dependencies**

get\_ipython().system('pip install -U ibm-watson-machine-learning')

from ibm\_watson\_machine\_learning import APIClient

import json

import numpy as np

**#Authenticate and Set space**

wml\_credentials = {

"apikey":"wO8JHOKuRoeaSrIC2KyKeQORrrtlveW1DYS3-eg\_31pv",

"url":"https://us-south.ml.cloud.ibm.com"

}

wml\_client = APIClient(wml\_credentials)

wml\_client.spaces.list()

SPACE\_ID="7f302674-b2a7-49e2-93d4-fed45335a7df"

wml\_client.set.default\_space(SPACE\_ID)

wml\_client.software\_specifications.list()

**#Save and Deploy Model**

import sklearn

sklearn.\_\_version\_\_

MODEL\_NAME = 'car\_model'

DEPLOYMENT\_NAME = 'Sales\_deploy'

CS\_MODEL = dtr

# Set Python Version

software\_spec\_uid = wml\_client.software\_specifications.get\_id\_by\_name('default\_py3.8')

# Setup model meta

model\_props = {

wml\_client.repository.ModelMetaNames.NAME: MODEL\_NAME,

wml\_client.repository.ModelMetaNames.TYPE: 'scikit-learn\_0.23',

wml\_client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID: software\_spec\_uid

}

model\_details = wml\_client.repository.store\_model(

model=CS\_MODEL,

meta\_props=model\_props,

training\_data=x\_train,

training\_target=y\_train

)

#model\_details

model\_uid = wml\_client.repository.get\_model\_uid(model\_details); model\_uid

wml\_client.connections.list\_datasource\_types()

# Set meta

deployment\_props = {

wml\_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT\_NAME,

wml\_client.deployments.ConfigurationMetaNames.ONLINE: {}

}

# Deploy

deployment = wml\_client.deployments.create(

artifact\_uid=model\_uid,

meta\_props=deployment\_props )