Future Sales Prediction

Name:CT.Swetha

Reg.no:912421104049

Phase4 project submission

INTRODUCTION:

One of the most common methods used to predict sales is regression analysis. This method involves using historical sales data to train a model that can predict future sales. The model can take into account factors such as past sales, marketing campaigns, and economic indicators to make its predictions.

DATA SOURCE

One of the most common methods used to predict sales is regression analysis. This method involves using historical sales data to train a model that can predict future sales. The model can take into account factors such as past sales, marketing campaigns, and economic indicators to make its predictions.

Datalink:

https://www.kaggle.com/datasets/chakradharmattapalli/future-sales-prediction

Data preprocessing

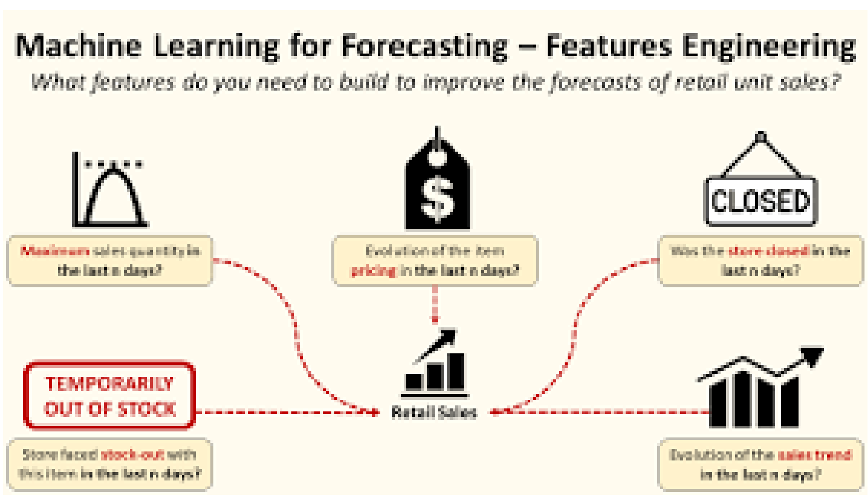
Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data to make it more suitable for the specific data mining task.

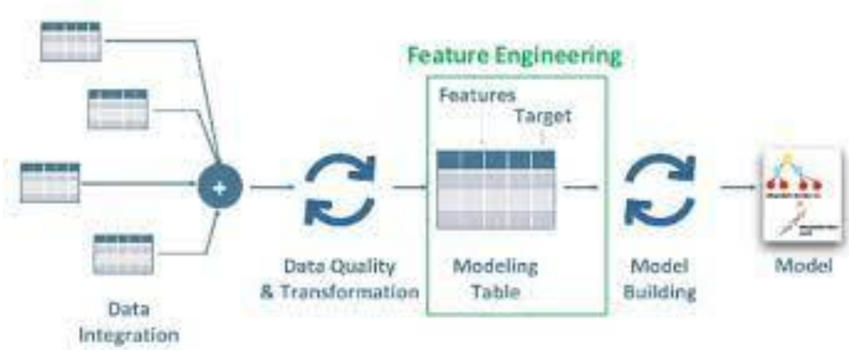
Feature Engineering

Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model. It involves selecting relevant

information from raw data and transforming it into a format that can be easily understood by a mode

Feature engineering can make your models more interpretable. By creating meaningful features, you can gain insights into which factors are driving demand and how they impact your predictions. This knowledge is invaluable for making informed business decisions.





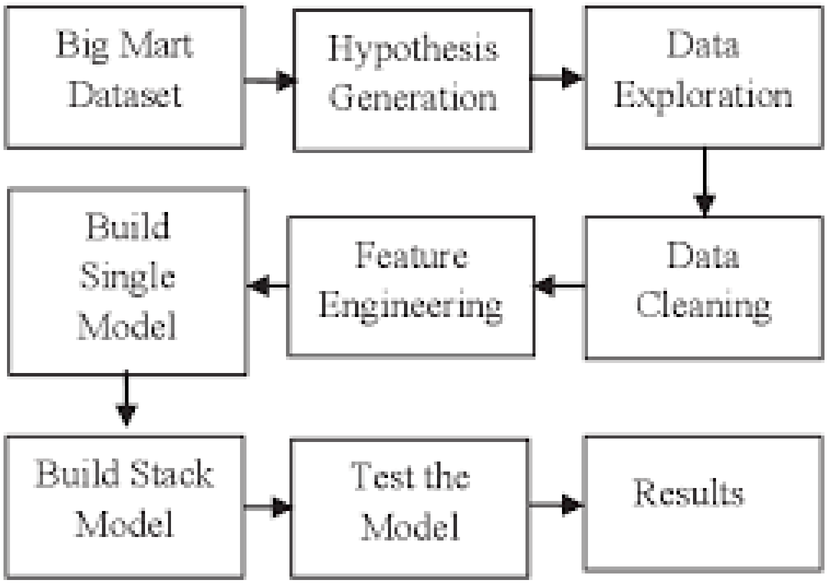
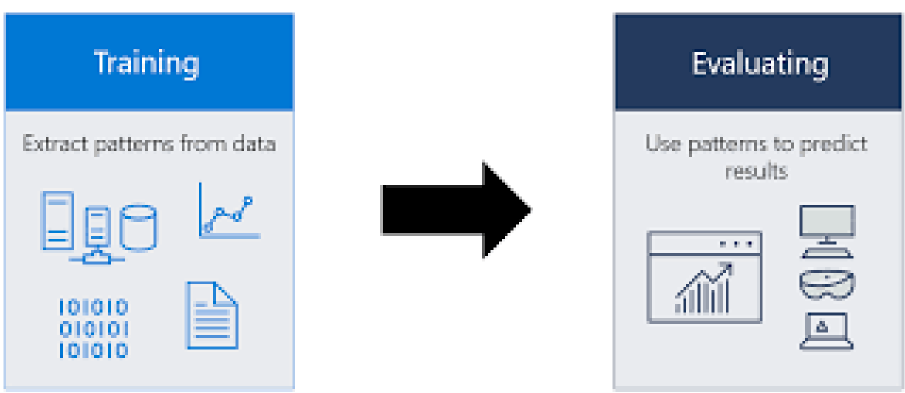


Fig. 1. Flow Diagram of the Proposed System

Model Training

Model training is the phase in the data science development lifecycle where

Practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.



தமிழ் இல்

In English

One of the most common methods used to predict sales is regression analysis. This method involves using historical sales data to train a model that can predict future sales. The model can take into account factors such as past sales, marketing campaigns, and economic indicators to make its predictions.

Predicting Sales

Forecasting the monthly sales with LSTM

Assess historical trendsExamine sales from the previous year. Break the numbers down by price, product, rep, sales period, and other relevant variables. Build those into a “sales run rate,” which is the amount of projected sales per sales period. This forms the basis of your sales forecast.

What is predicting sales of a product?

Use affordable market research techniques

- Ask your sales team. Sales representatives know your market intimately, including what your competitors are doing. ...
- Seek other sources of intelligence. ...
- Consider primary research. ...
- Start with a pilot project. ...
- Monitor your results and adjust.

This series of articles was designed to explain how to use Python in a simplistic way to fuel your company's growth by applying the predictive approach to all your actions. It will be a combination of programming, data analysis, and machine learning.

I will cover all the topics in the following nine articles:

1-Know Your Metrics

2-Customer Segmentation

3-Customer Lifetime Value Prediction

4-Churn Prediction

5- Predicting Next Purchase Day

6- **Predicting Sales**

7- Market Response Models

8- Uplift Modeling

9- A/B Testing Design and Execution

Articles will have their own code snippets to make you easily apply them. If you are super new to programming, you can have a good introduction for Python and Pandas (a famous library that we will use on everything) here. But still without a coding introduction, you can learn the concepts, how to use your data and start generating value out of it.

Example program:

Import pandas aspd # to extract data from dataset(.csv file)

Import csv #used to read and write to csv files

Import numpy asnp #used to convert input into numpy arrays to be fed to the model

Import matplotlib.pyplot asplt
sales forecasting

#to plot/visualize sales data and

.................Column Break.................

Import tensorflow astf
this model is built

acts as the framework upon which

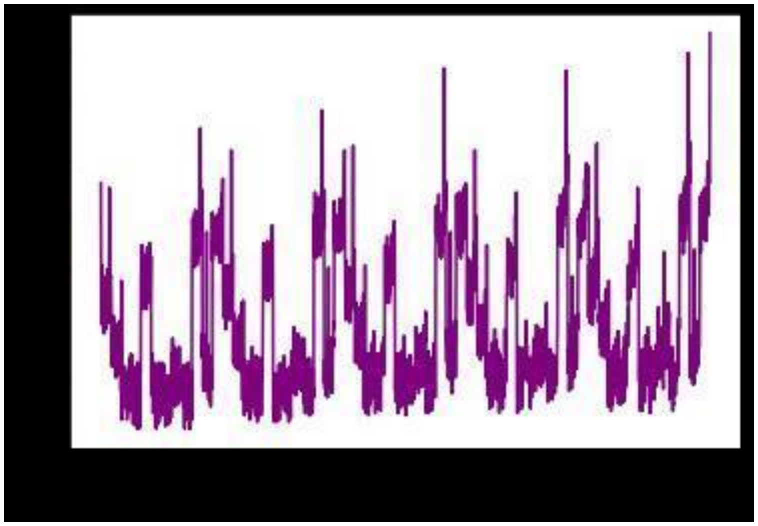
From tensorflow import keras
the model

#defines layers and functions in

List_row,date,traffic =

get_data('/home/abh/Documents/Python/Untitled
Folder/Sales_dataset')
Output:

Original data set for sales data for 5 years:



History = model.fit(

X =

[inp_day,inp_mon,inp_year,inp_week,inp_hol,inp7,inp_prev,inp_sess],

Y = out,

Batch_size=16,

Steps_per_epoch=50,

```
Epochs = 15,

Verbose=1,

Shuffle =False

)

        #all the inputs were fed into the model and the training was
completed
```

Output:

```
def conversion(week,days,months,years,list_row):
```

```
Epoch 1/15
50/50 [==============================] - 6s 15ms/step - loss: 0.0612 - acc: 0.0000e+00
Epoch 2/15
50/50 [==============================] - 1s 18ms/step - loss: 0.0288 - acc: 0.0000e+00
Epoch 3/15
50/50 [==============================] - 1s 20ms/step - loss: 0.0172 - acc: 0.0000e+00
Epoch 4/15
50/50 [==============================] - 1s 15ms/step - loss: 0.0099 - acc: 0.0000e+00
Epoch 5/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0084 - acc: 0.0000e+00
Epoch 6/15
50/50 [==============================] - 1s 18ms/step - loss: 0.0065 - acc: 0.0000e+00
Epoch 7/15
50/50 [==============================] - 1s 16ms/step - loss: 0.0053 - acc: 0.0000e+00
Epoch 8/15
50/50 [==============================] - 1s 18ms/step - loss: 0.0053 - acc: 0.0000e+00
Epoch 9/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0038 - acc: 0.0000e+00
Epoch 10/15
50/50 [==============================] - 1s 15ms/step - loss: 0.0039 - acc: 0.0000e+00
Epoch 11/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0037 - acc: 0.0000e+00
Epoch 12/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0036 - acc: 0.0000e+00
Epoch 13/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0035 - acc: 0.0000e+00
Epoch 14/15
50/50 [==============================] - 1s 17ms/step - loss: 0.0032 - acc: 0.0000e+00
Epoch 15/15
50/50 [==============================] - 1s 18ms/step - loss: 0.0029 - acc: 0.0000e+00
```

```
#lists have been defined to hold different inputs

 Inp_day = []

 Inp_mon = []
```

Inp_year = []

Inp_week=[]

Inp_hol=[]

Out = []

#converts the days of a week(Monday,Sunday,etc.) into one hot vectors and stores themasa dictionary

Week1 = number_to_one_hot(week)

#list_row contains primary inputs

For row in list_row:

#Filter out date from list_row

D = row[0]

#the date was split into three values date, month and year.

D_split=d.split('/')

If d_split[2]==str(year_all[0]):

#prevents use of the first year data to ensure each input contains previous year data as well.

Continue

```
Model: "model_1"
_____________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
=============================================================================================
input_day7 (InputLayer)         [(None, 7, 1)]       0
_____________________________________________________________________________________________
dense_15 (Dense)                (None, 7, 5)         10          input_day7[0][0]
_____________________________________________________________________________________________
input_day (InputLayer)          [(None, 31)]         0
_____________________________________________________________________________________________
input_mon (InputLayer)          [(None, 12)]         0
_____________________________________________________________________________________________
input_year (InputLayer)         [(None, 5)]          0
_____________________________________________________________________________________________
input_week (InputLayer)         [(None, 7)]          0
_____________________________________________________________________________________________
input_hol (InputLayer)          [(None, 1)]          0
_____________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 7, 5)         220         dense_15[0][0]
_____________________________________________________________________________________________
input_day_prev (InputLayer)     [(None, 1)]          0
_____________________________________________________________________________________________
input_day_sess (InputLayer)     [(None, 5)]          0
_____________________________________________________________________________________________
dense_10 (Dense)                (None, 5)            160         input_day[0][0]
_____________________________________________________________________________________________
dense_11 (Dense)                (None, 5)            65          input_mon[0][0]
_____________________________________________________________________________________________
dense_12 (Dense)                (None, 5)            30          input_year[0][0]
```

Input_hol = Input(shape=(inp_hol.shape[1],),name = ‘input_hol’)

Input_day7 = Input(shape=(inp7.shape[1],inp7.shape[2]),name = ‘input_day7’)

Input_day_prev = Input(shape=(inp_prev.shape[1],),name = ‘input_day_prev’)

Input_day_sess = Input(shape=(inp_sess.shape[1],),name = 'input_day_sess')

The model is quite straight-forward, all inputs were inserted into a dense layer with 5 units and 'relu' as activation function

X1 = Dense(5, activation='relu')(input_day)

X2 = Dense(5, activation='relu')(input_mon)

X3 = Dense(5, activation='relu')(input_year)

X4 = Dense(5, activation='relu')(input_week)

#encode the three

Conclusion: The phase2 submission was about future sales prediction with innovative technology.

Output:

From tensorflow.keras.models import Model

From tensorflow.keras.layers import Input, Dense,LSTM,Flatten

From tensorflow.keras.layers import concatenate

#an Input variable is made from every input array

Input_day = Input(shape=(inp_day.shape[1],),name = 'input_day')

Input_mon = Input(shape=(inp_mon.shape[1],),name = 'input_mon')

Input_year = Input(shape=(inp_year.shape[1],),name = 'input_year')

Input_week = Input(shape=(inp_week.shape[1],),name = ‘input_week’)
X5 = Dense(5, activation=’relu’)(input_hol)

X_6 = Dense(5, activation=’relu’)(input_day7)

X__6 = LSTM(5,return_sequences=True)(x_6) # LSTM isused to remember the importance of each day from the seven days data

X6 = Flatten()(x__10) # done to make the shape compatible to other inputs as LSTM outputs a three dimensional tensor

X7 = Dense(5, activation=’relu’)(input_day_prev)

X8 = Dense(5, activation=’relu’)(input_day_sess)

C = concatenate([x1,x2,x3,x4,x5,x6,x7,x8]) # all inputs are concatenated into one

Layer1 = Dense(64,activation='relu')©

Outputs = Dense(1, activation='sigmoid')(layer1) # a single output is produced with value ranging between 0-1.

now the model is initialized and created as well

Model =

Model(inputs=[input_day,input_mon,input_year,input_week,input_hol,input_day7,input_day_prev,input_day_sess], outputs=outputs)

Model.summary()

OUTPUT:

```
dense_13 (Dense)              (None, 5)            40          input_week[0][0]
__________________________________________________________________________________________________
dense_14 (Dense)              (None, 5)            10          input_hol[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)           (None, 35)           0           lstm_1[0][0]
__________________________________________________________________________________________________
dense_16 (Dense)              (None, 5)            10          input_day_prev[0][0]
__________________________________________________________________________________________________
dense_17 (Dense)              (None, 5)            30          input_day_sess[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)   (None, 70)           0           dense_10[0][0]
                                                               dense_11[0][0]
                                                               dense_12[0][0]
                                                               dense_13[0][0]
                                                               dense_14[0][0]
                                                               flatten_1[0][0]
                                                               dense_16[0][0]
                                                               dense_17[0][0]
__________________________________________________________________________________________________
dense_18 (Dense)              (None, 64)           4544        concatenate_1[0][0]
__________________________________________________________________________________________________
dense_19 (Dense)              (None, 1)            65          dense_18[0][0]
==================================================================================================
Total params: 5,184
Trainable params: 5,184
Non-trainable params: 0
```

Evaluation:

The most traditional way of sales performance evaluation is to look at the past sales data and the present sales data and make comparisons. It can quickly be seen how well they meet their targets, how their sales figures have risen or fallen, and whether their sales performance is in line with the company as a whole.

What is the need for evaluating sales performance?

Evaluation is also required for proper sales planning. A periodic evaluation of performance discloses their relative performance which is needed by the management to make suitable changes in their different assignment as to different territory, products or buyers.

Conclusion:

The phase 4 submission about future sales prediction done successfully.