

Why is Python the Best-Suited Programming Language for Machine Learning?

How to Start Learning Machine Learning?

ML | Training Image Classifier using Tensorflow Object Detection API

Top Machine Learning Applications in 2019

How Does Google Use Machine Learning?

How Does NASA Use Machine Learning?

10 Basic Machine Learning Interview Questions

Top Machine Learning Trends in 2019

Running Python script on GPU.

Relationship between Data Mining and Machine Learning

Overview of Data Science

Age of AI-based recruitment... What to expect?

ML | Frequent Pattern Growth Algorithm

ML | Spectral Clustering

Python | Convert image to text and then to speech

Recurrent Neural Networks Explanation

ML | Fuzzy Clustering

ML | Active Learning

ML | Multiple Linear Regression (Backward Elimination Technique)

ML | Raw and Central Moments

Pattern Recognition | Basics and Design Principles

Long Short Term Memory Networks Explanation

Inductive Learning Algorithm

Chi-Square Test for Feature Selection - Mathematical Explanation

ML | Extra Tree Classifier for Feature Selection

Text Generation using Recurrent Long Short Term Memory Network

Artificial Intelligence: Cause Of Unemployment

CNN | Introduction to Padding

ML | OPTICS Clustering Explanation

ML | Mathematical explanation of RMSE and R-squared error

Python | Linear Regression using sklearn

Prerequisite: Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

This article is going to demonstrate how to use the various Python libraries to implement linear regression on a given dataset. We will demonstrate a binary linear model as this will be easier to visualize.

In this demonstration, the model will use Gradient Descent to learn. You can learn about it here.

Step 1: Importing all the required libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Step 2: Reading the dataset

You can download the dataset here.

```
cd C:\Users\Dev\Desktop\Kaggle\Salinity
# Changing the file read location to the location of the dataset
df = pd.read_csv('bottle.csv')
df_binary = df[['Salnty', 'T_degC']]

# Taking only the selected two attributes from the dataset
df_binary.columns = ['Sal', 'Temp']

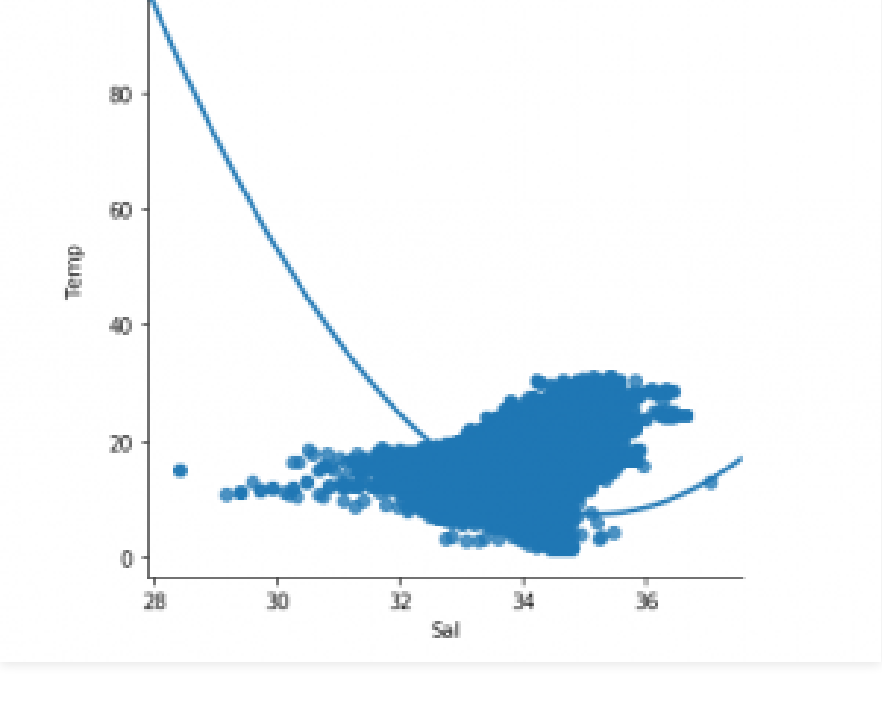
# Renaming the columns for easier writing of the code
df_binary.head()

# Displaying only the 1st rows along with the column names
```

	Sal	Temp
0	33.440	10.50
1	33.440	10.46
2	33.437	10.46
3	33.420	10.45
4	33.421	10.45

Step 3: Exploring the data scatter

```
sns.lmplot(x="Sal", y="Temp", data = df_binary, order = 2, ci = \n
# Plotting the data scatter
```



Step 4: Data cleaning

```
# Eliminating NaN or missing input numbers
df_binary.fillna(method ='ffill', inplace = True)
```

Step 5: Training our model

```
X = np.array(df_binary['Sal']).reshape(-1, 1)
y = np.array(df_binary['Temp']).reshape(-1, 1)

# Separating the data into independent and dependent variables
# Converting each dataframe into a numpy array
# since each dataframe contains only one column
df_binary.dropna(inplace = True)

# Dropping any rows with Nan values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size

# Splitting the data into training and testing data
regr = LinearRegression()

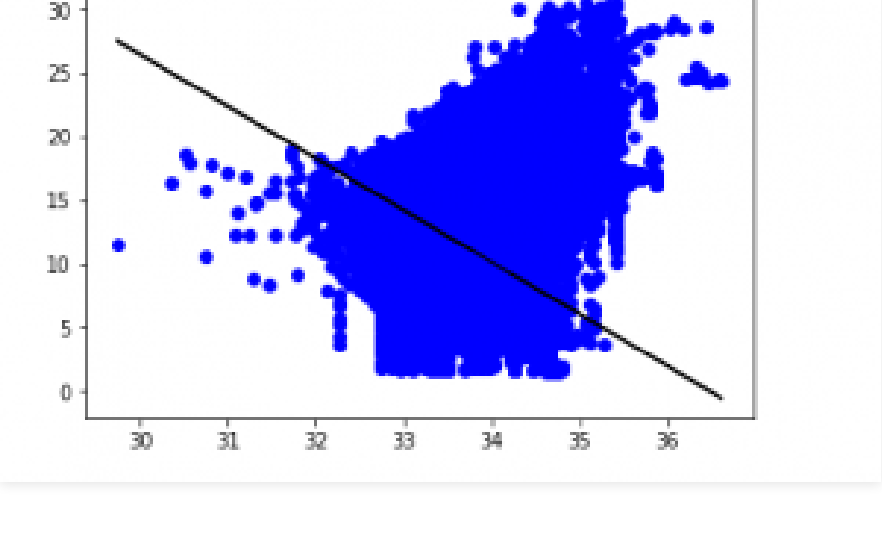
regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```

0.20780376990868232

Step 6: Exploring our results

```
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color ='b')
plt.plot(X_test, y_pred, color ='k')

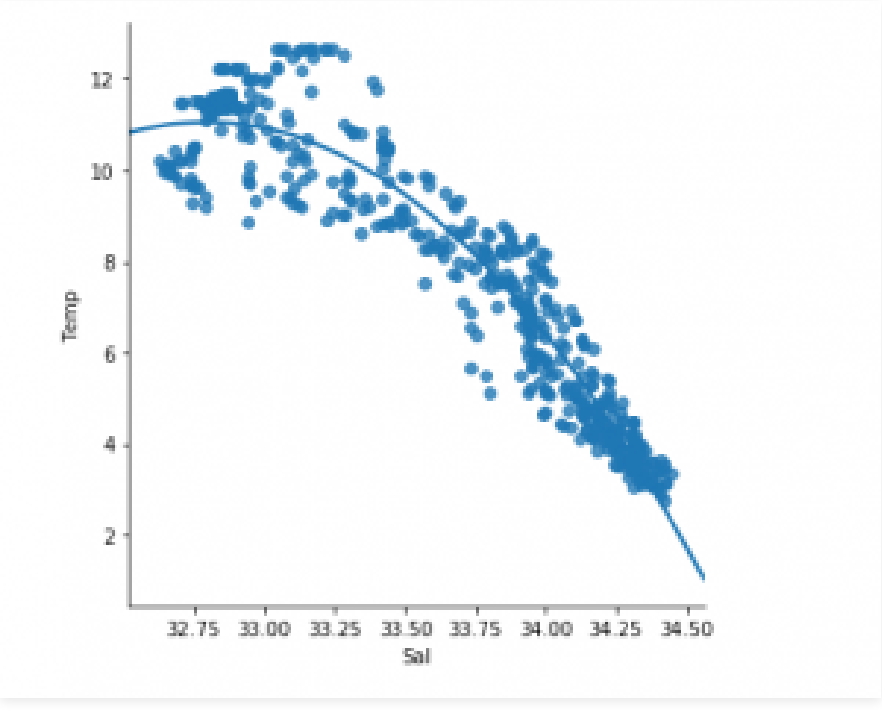
plt.show()
# Data scatter of predicted values
```



The low accuracy score of our model suggests that our regressive model has not fitted very well to the existing data. This suggests that our data is not suitable for linear regression. But sometimes, a dataset may accept a linear regressor if we consider only a part of it. Let us check for that possibility.

Step 7: Working with a smaller dataset

```
df_binary500 = df_binary[:500]
# Selecting the 1st 500 rows of the data
sns.lmplot(x="Sal", y="Temp", data = df_binary500,
order = 2, ci = None)
```



We can already see that the first 500 rows follow a linear model. Continuing with the same steps as before.

```
df_binary500.fillna(method ='ffill', inplace = True)
X = np.array(df_binary500['Sal']).reshape(-1, 1)
y = np.array(df_binary500['Temp']).reshape(-1, 1)

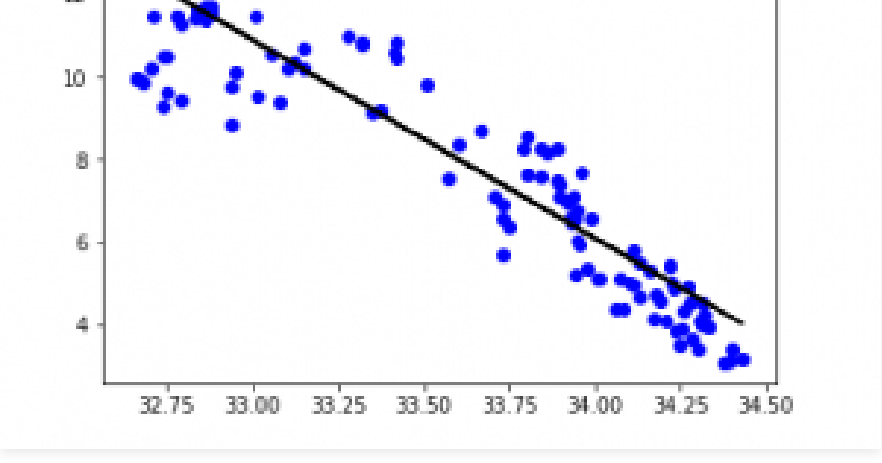
df_binary500.dropna(inplace = True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size

regr = LinearRegression()
regr.fit(X_train, y_train)
print(regr.score(X_test, y_test))
```

0.8475943139663558

```
y_pred = regr.predict(X_test)
plt.scatter(X_test, y_test, color ='b')
plt.plot(X_test, y_pred, color ='k')

plt.show()
```



Recommended Posts:

- Python | Decision Tree Regression using sklearn
- Univariate Linear Regression in Python
- Linear Regression (Python Implementation)
- ML | Multiple Linear Regression using Python
- ML | Linear Regression
- Multiple Linear Regression using R
- Linear Regression using PyTorch
- Simple Linear-Regression using R
- Linear Regression Using Tensorflow
- Gradient Descent in Linear Regression
- ML | Rainfall prediction using Linear regression
- ML | Normal Equation in Linear Regression
- ML | Locally weighted Linear Regression
- Pyspark | Linear regression using Apache MLlib
- A Practical approach to Simple Linear Regression using R



AlindGupta
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : shubham_singh

Article Tags : Machine Learning Python

Practice Tags : Machine Learning



Be the First to upvote.

0

To-do Done

No votes yet.

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Previous

Combining IoT and Machine Learning makes our future smarter

Next

Regularization in Machine Learning

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments