Python Program

Chapter 8: Regression Analysis

Chapter Objectives

In this chapter, we will:

- Introduce Linear Regression
- Compare two algorithms
  – Scikit-learn
  – Statsmodel

Chapter Concepts

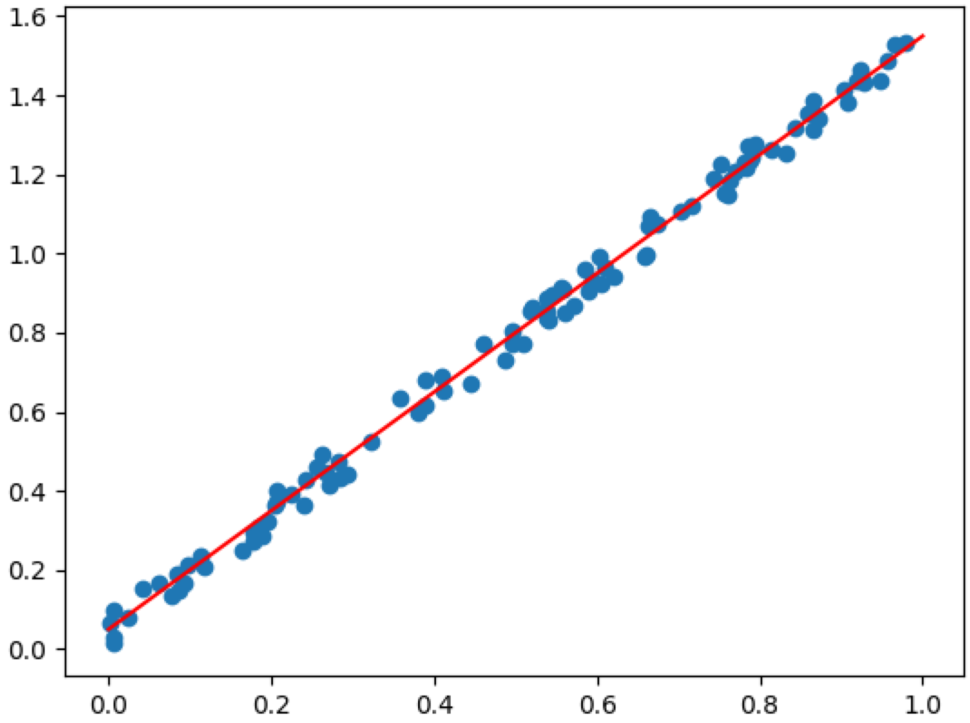
Regression Analysis

Algorithms

Chapter Summary

Linear Regression

- Given a collection of X, Y points, you could easily see there is a pattern
- If you remember enough algebra, you could describe the pattern of dots as roughly following the red line, which could be described with the formula y = 1.5x + .01



Linear Regression (continued)

- The idea is that the line that best describes the pattern of dots is the one that has the least distances of the dots from the line
- The formula that describes the line could then be used to predict a value that we have not observed
  – The better the line and formula are at describing that pattern of dots, the more accurate that prediction should be
- Extrapolate this idea onto more than just two axes and instead try to find a line that goes through many different dimensions and you have the idea of multiple linear regression
  – $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$
- Has many use cases
  – Predicting a stock or commodity price
  – Predicting election results
  – Predicting crime rate

Linear Regression (continued)

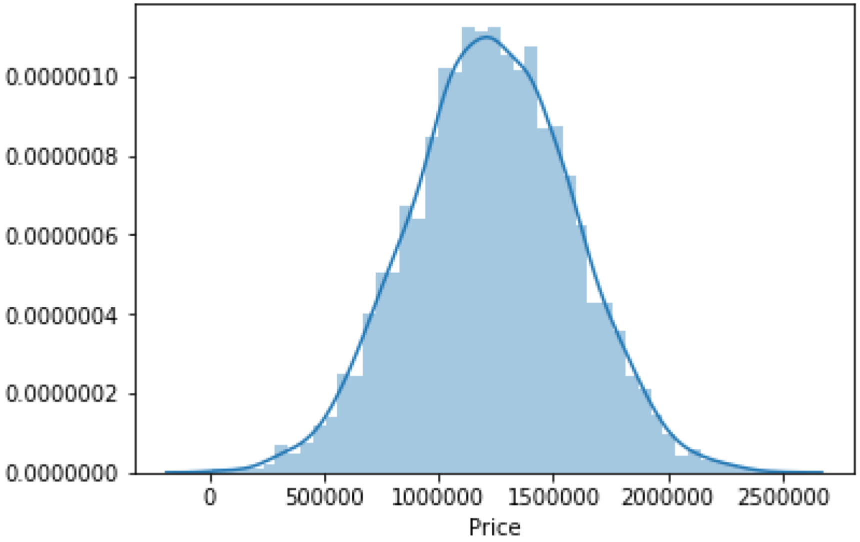
- Is a supervised model that requires training from a known set of data and testing to see how good it is at predicting before using it for real predictions
- Only works with numeric values
  - Categorical data needs to be dummy encoded
- Does not deal well with missing data, so must be fixed by removing or replacing with central tendency
- There are many algorithms to do this, each with its own pros and cons

Dataset

- For our examples, let's use a public data set of housing data

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
USAhousing = pd.read_csv('USA_Housing.csv')
print(USAhousing.columns)
print(USAhousing.head())
sns.distplot(USAhousing['Price'])
```

- The data has no categorical columns but does have an address we will ignore
- Plotting the distribution of Prices shows that they are normally distributed



Chapter Concepts

Create a Scikit Model

- Prep the data and fit the model on the training set

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
x = USAhousing[['Avg. Area Income', 'Avg. Area House Age',
'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms',
'Area Population']]
y = USAhousing['Price']
trainX, testX, trainY, testY = train_test_split(x, y, test_size
= 0.4, random_state = 101)

from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(trainX, trainY)
```

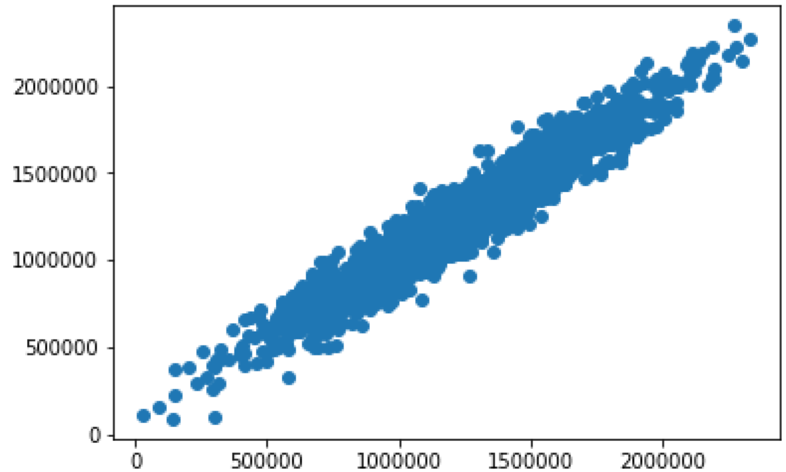
View the Model

- Run predictions on the test set and compare them to the real values to see how well the model did at predicting

```
predictions = lm.predict(testX)
plt.scatter(testY, predictions)
print("Mean squared error: %.2f" % mean_squared_error(testY,
predictions))
print('Variance score: %.2f' % r2_score(testY, predictions))
```

- The variance score, also called R-Squared, indicates how well in general the model fits
  - Ranges from 0 – 1, the closer to 1 the better
  - .92 means this model fit reasonably well

```
Mean squared error: 10460958907.21
Variance score: 0.92
```



Create a Stats Model

- The stats library offers a different version of the algorithm
- Provides a little more information about the accuracy of the model

```
import statsmodels.api as sm
model = sm.OLS(trainY, trainX).fit()
print (model.summary())
predictions = model.predict(testX)
plt.scatter(testY, predictions)
plt.show()
```

Interpret the Model

- You automatically get the R-squared from the summary function
  - Additionally, the Adjusted R-squared is helpful because it helps to compare models with different numbers of predictor variable
- The P-value also identify which features are most significant in influencing the value you are trying to predict, closer to one is better.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.965
Model:                            OLS   Adj. R-squared:                  0.965
Method:                 Least Squares   F-statistic:                 1.633e+04
Date:                Fri, 17 May 2019   Prob (F-statistic):               0.00
Time:                        23:08:27   Log-Likelihood:                -41426.
No. Observations:                3000   AIC:                         8.286e+04
Df Residuals:                    2995   BIC:                         8.289e+04
Df Model:                           5
Covariance Type:            nonrobust
================================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Avg. Area Income              10.1001      0.346     29.176      0.000       9.421      10.779
Avg. Area House Age         4.972e+04   3870.040     12.846      0.000    4.21e+04    5.73e+04
Avg. Area Number of Rooms  -9135.0559   4226.074     -2.162      0.031   -1.74e+04    -848.754
Avg. Area Number of Bedrooms 4272.2896  4029.066      1.060      0.289   -3627.728    1.22e+04
Area Population                8.4544      0.419     20.171      0.000       7.633       9.276
==============================================================================
Omnibus:                        0.002   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.999   Jarque-Bera (JB):                0.000
Skew:                          -0.000   Prob(JB):                         1.00
Kurtosis:                       2.998   Cond. No.                     9.34e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.34e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
Mean squared error: 59842619587.25
Variance score: 0.53
```

Chapter Concepts

Regression Analysis

Algorithms

Chapter Summary

Next Steps

- Regression has a lot more complexity to it once you master the basics
- Some subjects to explore in this area:
  - Under- and over-fitting a model
  - Correlation between the independent variables
  - Non-linear regression

Chapter Summary

In this chapter, we have:

- Introduced Linear Regression
- Compared two algorithms
  - Scikit-learn
  - Statsmodel