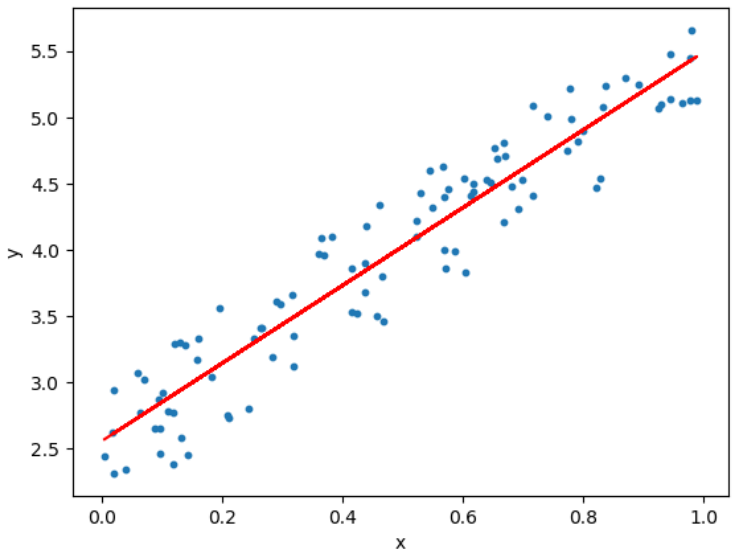
Linear Regression using sklearn

Linear Regression is usually the first machine learning algorithm that every data scientist comes across. It is a simple model but everyone needs to master it as it lays the foundation for other machine learning algorithms.

Where can Linear Regression be used?
It is a very powerful technique and can be used to understand the factors that influence profitability. It can be used to forecast sales in the coming months by analyzing the sales data for previous months. It can also be used to gain various insights about customer behaviour.

From statistics.laed.com;

"Linear regression is the next step up after correlation. It is used when we want to predict the value of a variable based on the value of another variable. The variable we want to predict is called the dependent variable (or sometimes, the outcome variable). The variable we are using to predict the other variable's value is called the independent variable (or sometimes, the predictor variable). For example, you could use linear regression to understand whether exam performance can be predicted based on revision time; whether cigarette consumption can be predicted based on smoking duration; and so forth. If you have two or more independent variables, rather than just one, you need to use multiple regression."



This is the first blog of the machine learning series that I am going to cover. One can get overwhelmed by the number of articles in the web about machine learning algorithms. My purpose of writing this blog is two-fold. It can act as a guide to those who are entering into the field of machine learning and it can act as a reference for me.

Table of Contents

What is Linear Regression

The objective of a linear regression model is to find a relationship between one or more features(independent variables) and a continuous target variable(dependent variable). When there is only feature it is called *Uni-variate* Linear Regression and if there are multiple features, it is called *Multi*ple Linear Regression.

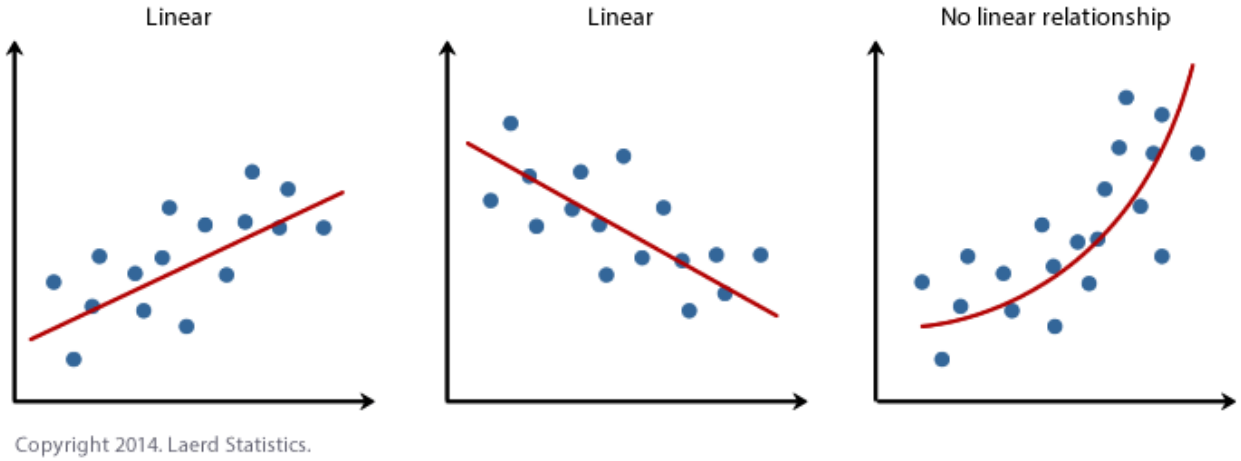
Linear regression is a statistical approach for modeling relationship between a dependent variable with a given set of independent variables. In this lesson, we refer dependent variables as **response** and independent variables as **features** for simplicity.

Assumptions

When you choose to analyse your data using linear regression, part of the process involves checking to make sure that the data you want to analyse can actually be analysed using linear regression. You need to do this because it is only appropriate to use linear regression if your data "passes" six assumptions that are required for linear regression to give you a valid result. In practice, checking for these six assumptions just adds a little bit more time to your analysis.

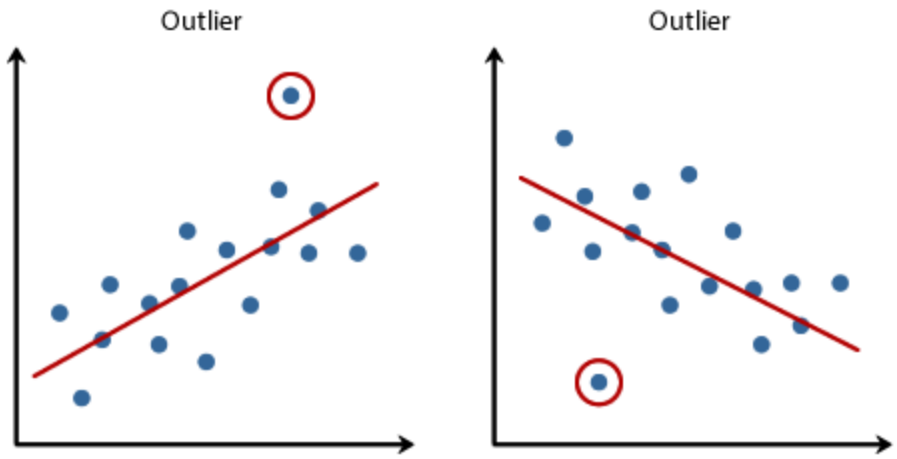
Before we introduce you to these six assumptions, do not be surprised if, when analyzing your own data, one or more of these assumptions is violated (i.e., not met). This is not uncommon when working with real-world data rather than textbook examples, which often only show you how to carry out linear regression when everything goes well! However, don’t worry. Even when your data fails certain assumptions, there is often a solution to overcome this. First, let’s take a look at these six assumptions:

- **Assumption #1:** Your two variables should be measured at the **continuous** level (i.e., they are either **interval** or **ratio** variables). Examples of **continuous variables** include revision time (measured in hours), intelligence (measured using IQ score), exam performance (measured from 0 to 100), weight (measured in kg), and so forth. You can learn more about interval and ratio variables in our article: Types of Variable.
- **Assumption #2:** There needs to be a **linear relationship** between the two variables. Whilst there are a number of ways to check whether a linear relationship exists between your two variables, we suggest creating a scatterplot (In python Scatter matrix). Your scatterplot may look something like one of the following:



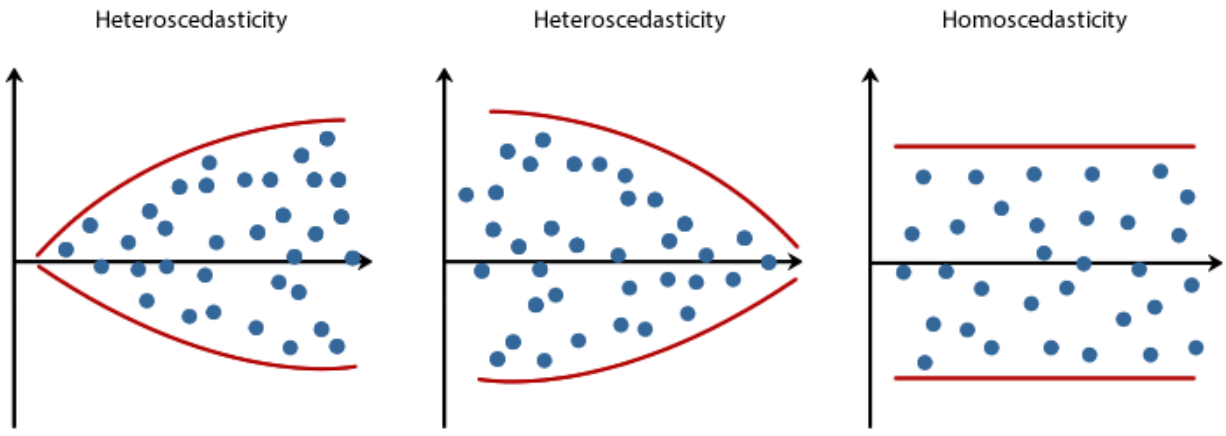
If the relationship displayed in your scatterplot is not linear, you will have to either run a non-linear regression analysis, perform a polynomial regression or "transform" your data.

- **Assumption #3:** There should be **no significant outliers**. An outlier is an observed data point that has a dependent variable value that is very different to the value predicted by the regression equation. As such, an outlier will be a point on a scatterplot that is (vertically) far away from the regression line indicating that it has a large residual, as highlighted below:



The problem with outliers is that they can have a negative effect on the regression analysis (e.g., reduce the fit of the regression equation) that is used to predict the value of the dependent (outcome) variable based on the independent (predictor) variable. **Assumption #4:** You should have **independence of observations**, which you can easily check using the Durbin-Watson statistic.

- **Assumption #5:** Your data needs to show **homoscedasticity**, which is where the variances along the line of best fit remain similar as you move along the line. Whilst we explain more about what this means and how to assess the homoscedasticity of your data in our enhanced linear regression guide, take a look at the three scatterplots below, which provide three simple examples: two of data that fail the assumption (called heteroscedasticity) and one of data that meets this assumption (called homoscedasticity):



Whilst these help to illustrate the differences in data that meets or violates the assumption of homoscedasticity, real-world data can be a lot more messy and illustrate different patterns of heteroscedasticity. Therefore, in our enhanced linear regression guide, we explain: (a) some of the things you will need to consider when interpreting your data; and (b) possible ways to continue with your analysis if your data fails to meet this assumption.

- **Assumption #6:** Finally, you need to check that the **residuals (errors)** of the regression line are **approximately normally distributed**

Applications:

1. Trend lines: A trend line represents the variation in some quantitative data with passage of time (like GDP, oil prices, etc.). These trends usually follow a linear relationship. Hence, linear regression can be applied to predict future values. However, this method suffers from a lack of scientific validity in cases where other potential changes can affect the data.
2. Economics: Linear regression is the predominant empirical tool in economics. For example, it is used to predict consumption spending, fixed investment spending, inventory investment, purchases of a country's exports, spending on imports, the demand to hold liquid assets, labor demand, and labor supply.
3. Finance: Capital price asset model uses linear regression to analyze and quantify the systematic risks of an investment.
4. Biology: Linear regression is used to model causal relationships between parameters in biological systems. Etc

Simple Linear Regression

Simple linear regression is an approach for predicting a **response** using a **single feature**.
It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

Let’s do a simple linear regression

You can download the famous mpg dataset from the UCI Machine Learning Repository, or just google “mpg.csv.” Using pandas, you can quickly read in the CSV into a DataFrame.

Data Set Information:

This dataset is a slightly modified version of the dataset provided in the StatLib library. In line with the use by Ross Quinlan (1993) in predicting the attribute "mpg", 8 of the original instances were removed because they had unknown values for the "mpg" attribute. The original dataset is available in the file "auto-mpg.data-original".

Attribute Information:

1. mpg: continuous
2. cylinders: continuous
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: continuous
8. origin: continuous
9. car name: category

In our data set let’s see the relationship between miles per gallon and horsepower.
A salesperson for a large car brand wants to determine whether there is a relationship the car horsepower and miles per gallon. As such, the car “horsepower (Engine Power)" is the independent variable and the "mpg (Miles per 3 litre)" the car travels is the dependent variable. The salesperson wants to use this information to determine which cars to offer potential customers in new areas where average income is known considering the fuel prices might be a concern to them.

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.linear_model import LinearRegression
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
df = pd.read_csv('mpg.csv')
```

Next up, we will clean the dataset and remove the missing values. Using pandas, we replace question marks with NaNs and remove these rows.

```
df = df.replace('?', np.nan)
df = df.dropna(inplace=True)
```

Dimensions of Dataset

We can get a quick idea of how many instances (rows) and how many attributes (columns) the data contains with the shape property. Also we can check statistical summaries

```
print(df.shape)
print(df.describe())
```

Now we can look at the interactions between the variables.

First, let's look at scatterplots of all pairs of attributes. This can be helpful to spot structured relationships between input variables.

```
from pandas.plotting import scatter_matrix
scatter_matrix(df)
plt.show()
```

Now we can go to our **LinearRegression**, We are going to hold back some data that the algorithms will not get to see and we will use this data to get a second and independent idea of how accurate the best model might actually be.

We will split the loaded dataset into two, 70% of which we will use to train our models and 30% that we will hold back as a validation dataset.

Since this a simple Linear Regression, we reduce our data frame to only one feature (horsepower) and one response (mpg) .

```
df2  = df[['horsepower', 'mpg']]
array = df2.values
X = array[:, 0:1]
Y = array[:,1]
validation_size = 0.30
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
    test_size=validation_size, random_state=seed)
model = LinearRegression()
model.fit(X_train, Y_train)
```

Now that we have our model, we can check how well it performs. In the first instance, we run the model on our test set. Some good evaluation metrics for linear regression are mean squared error and the R^2 score. R^2 gives you the percentage , close to means very accurate. mean squared error mean standard error

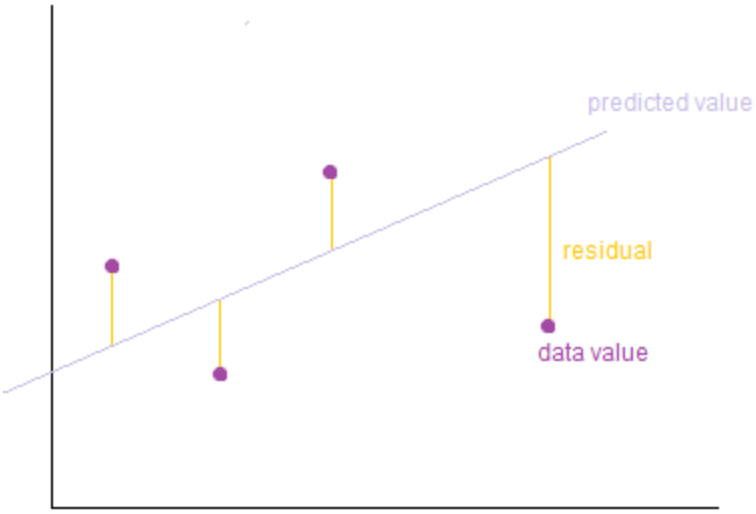
```
y_predicted = model.predict(X_validation)

print("Mean squared error: %.2f" % mean_squared_error(Y_validation, y_predicted))
print('R²: %.2f' % r2_score(Y_validation, y_predicted))
```

Training of the model here means to find the parameters so that the model best fits the data.

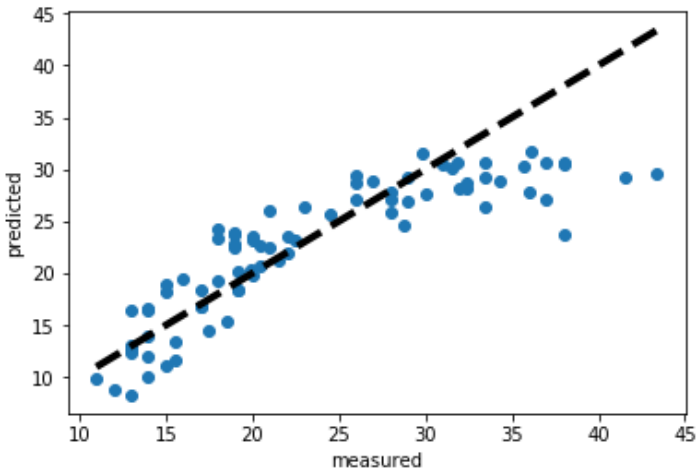
How do we determine the best fit line?

The line for which the the *error* between the predicted values and the observed values is minimum is called the best fit line or the **regression** line. These errors are also called as ***residuals***. The residuals can be visualized by the vertical lines from the observed data value to the regression line.




```
fig, ax = plt.subplots()
ax.scatter(Y_validation, y_predicted)
ax.plot([Y_validation.min(),Y_validation.max()], [Y_validation.min(),Y_validation.max()],
    'k--', lw=4)
ax.set_xlabel('measured')
ax.set_ylabel('predicted')
plt.show()
```

A visual inspection of the performance usually reveals some interesting findings. In this case, we can see that there seems to be some non-linearity in our data.



Read more https://www.investopedia.com/ask/answers/060315/what-difference-between-linear-regression-and-multiple-regression.asp

https://www.statisticshowto.datasciencecentral.com/line-of-best-fit/

https://towardsdatascience.com/car-selection-and-sales-day-prediction-8c4a474f9dca