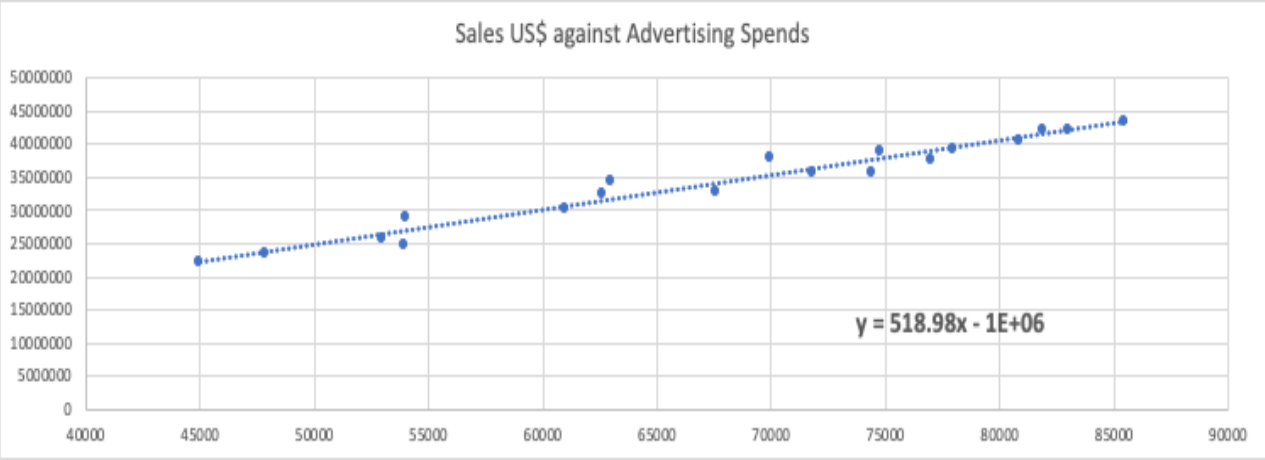
Understanding Linear Regression

Structure of this Module

Understanding Linear Regression through a Prediction Problem (Project)

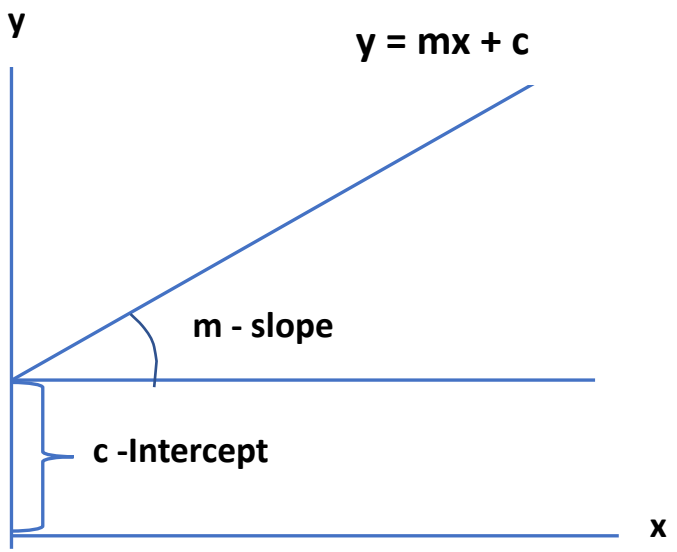
TOPICS
Introduction to Linear Regression
Understanding Cost Functions
Linear Regression Assumptions
Problem Statement
Data Pre-processing
Building our LR Model
Understanding Model Performance
Model Tuning methods

Simple Linear Regression with an Example



Year	Advertising Spend (US$)	Sales US$
2001	45000	22221000
2002	47900	23258000
2003	53950	24625000
2004	54000	28924700
2005	53000	25698000
2006	61000	30158600
2007	62600	32466000
2008	63000	34221000
2009	67600	32812000
2010	71800	35419000
2011	70000	37681000
2012	74400	35682000
2013	74800	38946000
2014	77000	37559000
2015	78000	39215000
2016	81900	41922000
2017	80900	40519000
2018	83000	41958000
2019	85500	43419000
2020	95000	?
2021	102000	?
2022	110000	?

Simple Linear Regression with an Example



- Predictors
- Beta Coefficients
- Independent and Target Variables
- Making Predictions
- Multiple Linear Regression
- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots\ldots + \beta_n x_n$

Linear Regression

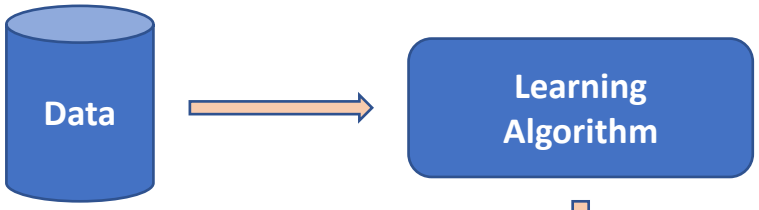
A Linear Regression model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the *bias term* (also called the *intercept term*).

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$ **<<< Linear Regression Equation**

- $\hat{y}$ is the predicted value.
- n is the number of features.
- x_i is the i^{th} feature value.
- $\boldsymbol{\theta}_j$ is the j^{th} model parameter (including the bias term $\boldsymbol{\theta}_0$ and the feature weights $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_n$)

Example: $y = 3.X_1 + 4.X_2 + 2.X_3 + 8.X_4 + 6.X_5 + 9.7$

Learning



$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$y = f(x)$$

Learning Algorithm **learns** from the Data to find optimum values of parameters

$$y = 3.X_1 + 4.X_2 + 2.X_3 + 8.X_4 + 6.X_5 + 9.7$$

Model

Learning and Cost Function

"What does the Machine Learning Algorithm actually learn?"

In simple terms, a Machine Learning Algorithm learns a function ***f*** such that ***f*(*X*)** maps to *y*.
In other words, the Algorithm learns how to **model a relationship** between ***X*** (i.e., features, or, more traditionally, independent variable(s)) and ***y*** (the target, response or more traditionally the dependent variable).

In the case of Linear Regression, the Linear Regression model learns to find out the optimum values of θ_0 to θ_n in order for the correct prediction of *y* given *X*. The relationship that the Algorithm learns is the '**model**', and θ_0 to θ_n are called the '**model parameters'**.

There are several ways to learn the parameters of a LR model, **minimising a cost function** is a commonly used one and is used in our case of Linear Regression Model.

A ***cost function or loss function*** is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y.

This is typically expressed as a difference or distance between the **predicted value** and the **actual value (ground truth)**.

The objective of an ML model, is to find parameters, weights or a structure that **minimises** the cost function.

The technique that our Linear Regression Algorithm uses is called Gradient Descent.

Cost Functions

The Cost Function calculation for Linear Regression Models can be varied. However, they all are calculated by determining the differences between the Actual Value and Predicted Value of the Target Variable for all data points within the Test Dataset. Let's say the actual values are denoted by '**y**' and the predicted values are denoted by '**ŷ**' (y-hat).

SSR or Sum of Squared Residuals: $\sum_{i=1}^{n}(y - \hat{y})^2$
For all pairs of y and ŷ, SSR is the sum of the squares of their differences.

RMSE or Root Mean Squared Error: $\sqrt{\sum_{i=1}^{n}(y - \hat{y})^2}$
For all pairs of y and ŷ, RMSE is the square root of SSR.

MSE or Mean Squared Error: $\sum_{i=1}^{n}(y - \hat{y})^2$ / n
For all pairs of y and ŷ, MSE is the mean of SSR.

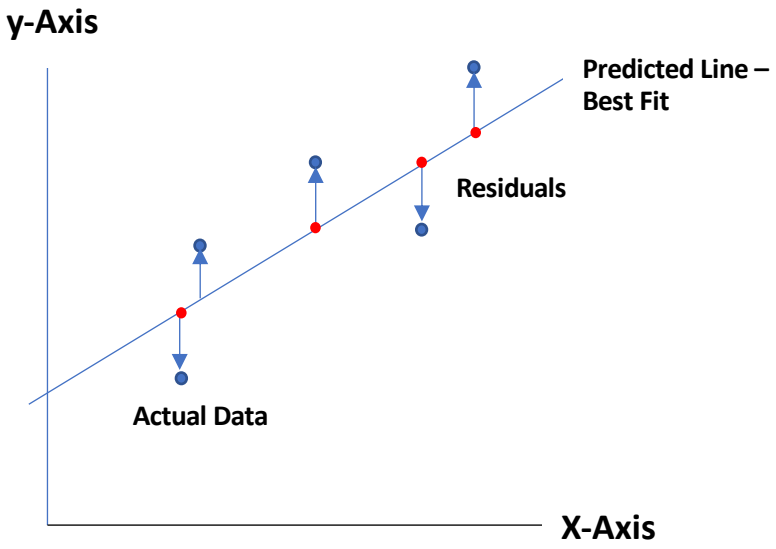
Co-efficient of Determination or R^2 is another measure that is used frequently to measure predictability of LR models. It is a value between 0 and 1, with 1 indicating the perfect predictability, hence, higher the value of R^2 is better. The value of R^2 is determined as the follows:

$$R^2 = 1 - \frac{RSS}{TSS}$$

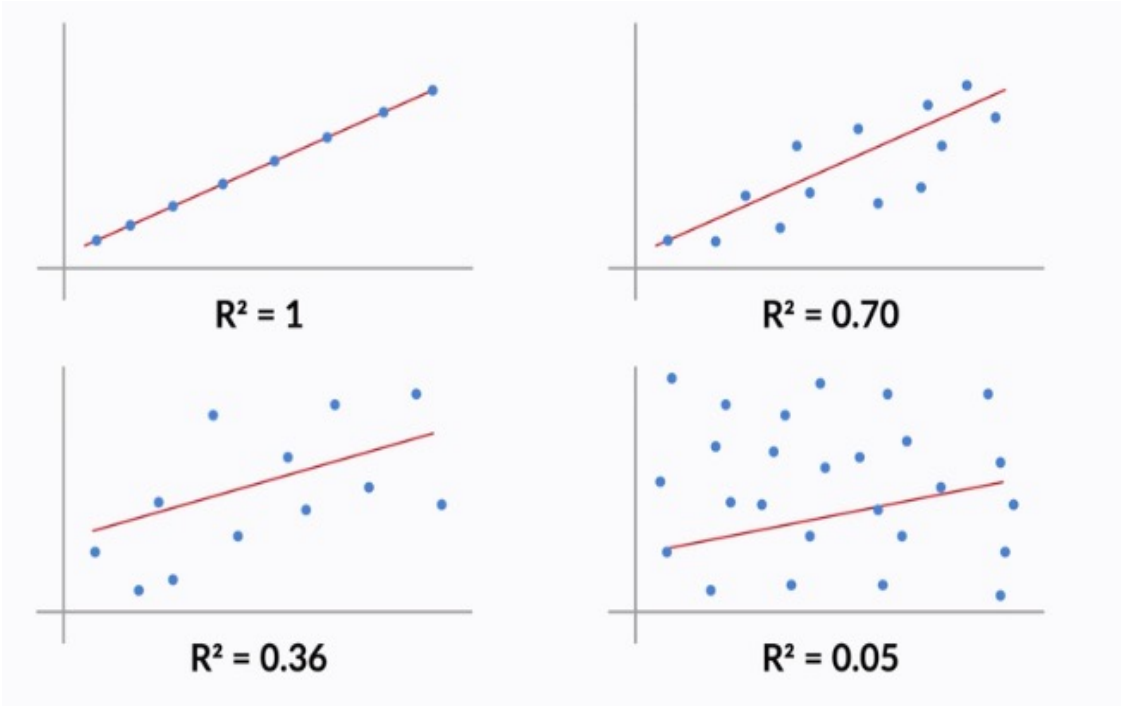
[RSS = SSR and TSS or the Total Sum of Squares is the sum of squares of the differences between the predicted values and the mean of actual values].

Cost Functions

Residuals Calculation



Physical Significance of R^2

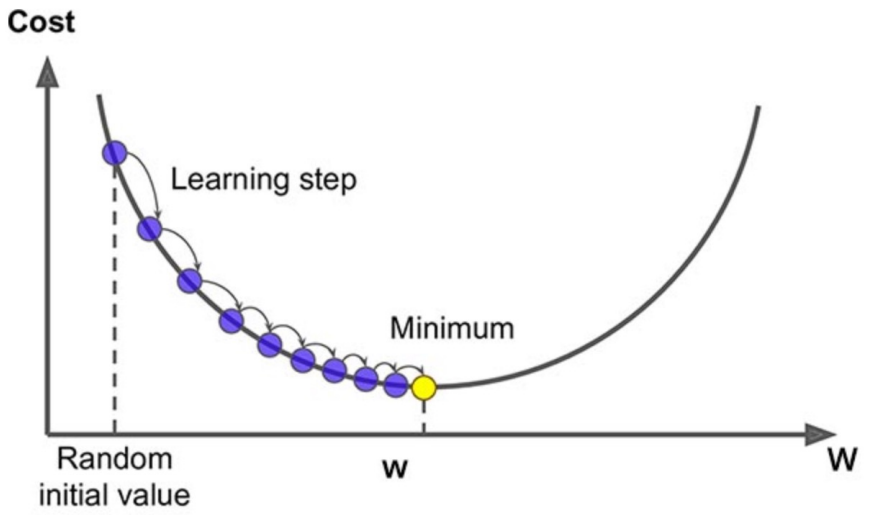


Gradient Descent

The Gradient Descent Method is an optimization technique that is used by Linear Regression Algorithms that aims at finding the best possible Linear Equation between the predictor and target variables by **minimizing the cost function**. Finding the best possible equation means finding the set of optimum β-coefficients ($\beta_1, \beta_2, \beta_3$) and intercept ($\beta_0$).

The way Gradient Descent Algorithms works is that it **assigns an initial set of values to the β-coefficients to the equation** and determines the predicted target (y) values for the input dataset.

Thereafter the **derivative of the equation is calculated**. The derivative refers to the slope of the equation at a given point. The slope will indicate which direction the values of the coefficients have to be moved in order to get a lower value of the cost function.

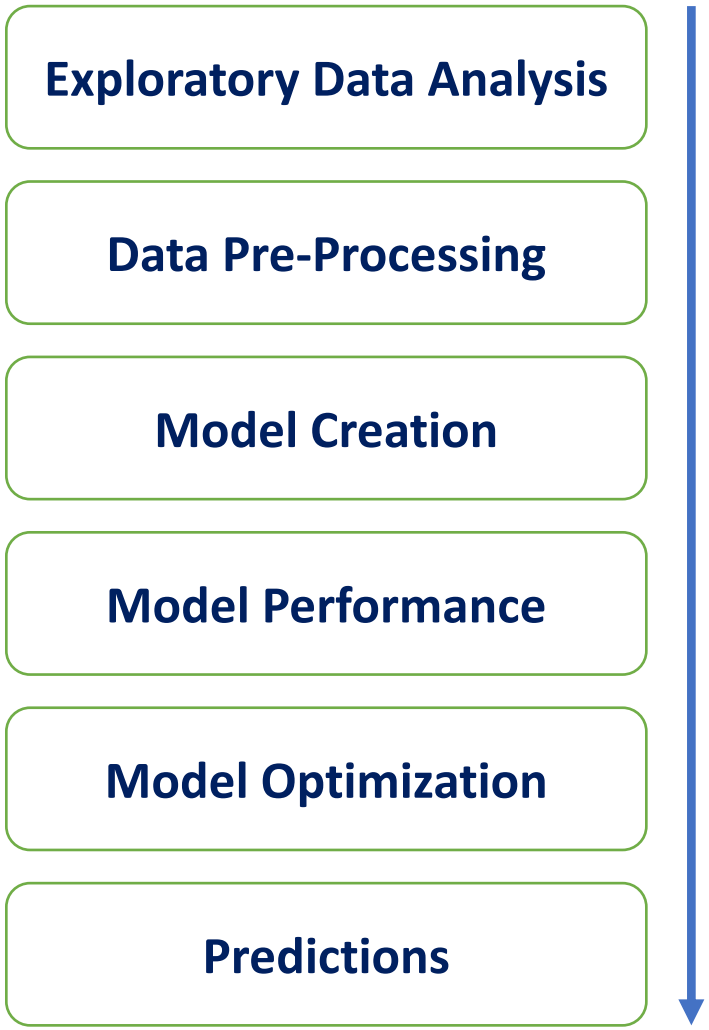


As the next step, Gradient Descent will **update the values of the β-coefficients towards the direction indicated by the derivative**. The update of the values is by a quantum that is known as the "**Learning Rate**" or the Alpha value. This value can be specified as part of the Linear Regression as a Hyper Parameter to optimize the speed of the model building process. A higher Learning rate will make the model building faster but will have the risk of skipping the most optimum combination of coefficients. On the other hand, a lower Learning Rate will make the model building slower but will have the ability to attain a best set values for the final model.

Linear Regression Problem Statement

A Car Sales agency employs inspectors to check any car that come in for being sold to determine the price at which it can be sold. This involves finding all the attributes and features of the car and based on the inspector's past experience, assign a sale price value. The Agency wants to automate this process using the historical transactions (records) it has. It wants us to create a Model that models the relationship between the car features. This model should be able to predict an estimated price for a car given it's features.

Linear Regression Steps



Libraries

- **Pandas**: Series and Dataframe, strong data management and manipulation within the programs
- **Numpy**: Array and array management features
- **Matplotlib**: Graphs and charts
- **Seaborn**: advanced graphs and charts
- **Sklearn**: Scikit-learn (formerly scikits.learn and also known as sklearn) is a Machine Learning Library for Python having various Classification, Regression and Clustering algorithms and is designed to interoperate with the Python numerical and scientific libraries like NumPy and SciPy.
- **Sklearn:Statsmodel**: OLS to create the LinearRegression model. “a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.”
- **Sklearn.model_selection**: Importing test_train_split, cross_validation, tuning_hyperparameters
- **Sklearn.metrics**: Model metrics – R2_score here
- **Sklearn.linear_model** – Machine Learning models, LinearRegression in this case
- **Sklearn.feature_selection**: RFE
- **Sklearn.preprocessing** – Feature Scaling

Conversion of Categorical Features

We have said earlier that we need to convert all textual variables that are potential predictor variables to numeric ones as the model can be built only with numeric variables.

There are two types of categorical variables – **Nominal Variables** and **Ordinal variables**.

Nominal data are the ones that do not have an inherent degree within them and are purely of label type. E.g. 'Hatchback', 'Sedan', etc - these values do not convey any apparent degree of better or worse and act purely as labels. This is an example **Nominal variable**.

In contrast, when we look at fields like Price_Category having values like 'Economy', 'PremiumEconomy', 'Premium', 'Luxury', 'SuperLuxery', etc, which in this sequence convey a degree of importance or value for this field (poor, average, good, excellent). This is an example of an **ordinal variable**. Ordinal variables, though contain textual values, but these texts involve degrees of values.

The treatment of these two types of variables to make them ingestible by the model are different.

For Ordinal variables, it is fairly straightforward – **we will map the increasing degrees with correspondingly increasing discreet numeric values**. The numeric values reflect the degree of the Ordinal values.

Creating Dummy Variables

Dummy Variables are used to replace **Nominal Categorical Variables** in Features converting them into numeric variables.

CARBODY
convertible
convertible
hatchback
sedan
sedan
sedan
sedan
wagon
sedan
hatchback
sedan
sedan
sedan
sedan
sedan
sedan
sedan
sedan
sedan
hatchback
hatchback
sedan
hatchback
hatchback
hatchback
hatchback
sedan
sedan
sedan
wagon

convertible	hatchback	sedan	wagon
1	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	0	1
0	0	1	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	0	0
0	0	1	0
0	1	0	0
0	1	0	0
0	1	0	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	1	0
0	0	0	1

Drop Original

drop_first = True

Feature Scaling

Machine learning algorithms like Linear Regression, Logistic Regression, Neural Network, etc. that use Gradient Descent as an optimization technique require input data to be scaled. This makes Feature Scaling a very important pre-processing step for these Algorithms.

Scaling involves **converting the numeric values to a standard minimized scale**.

There are two popular ways to perform data scaling – **Normalization** and **Standardization**.

Example DataSet

X1	X2	X3	X4	X5	X6	X7	Y
LotFrontage	LotArea	BsmtFinSF1	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	SalePrice
65	8450	706	150	856	856	854	208500
80	9600	978	284	1262	1262	0	181500
68	11250	486	434	920	920	866	223500
60	9550	216	540	756	961	756	140000
84	14260	655	490	1145	1145	1053	250000
85	14115	732	64	796	796	566	143000
75	10084	1369	317	1686	1694	0	307000

Different Scales

Feature Scaling for Gradient Descent: The difference in ranges of features will cause different step sizes for each feature. To ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we scale the data before feeding it to the model.

For Distance Based Algorithms like KNN, K-means, and SVM: These are the most affected by different range of features. This is because behind the scenes **they use distance measures between data points in the Feature space to determine their similarities.**

Feature Scaling

Normalization (Min-Max Scaling) involves scaling the values in each column separately and proportionately to values between 0 to 1. These scaled values are floating point numbers corresponding to each original value.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Formula for Normalization

- For an observation (row), if the value of X is the minimum value in the column, the numerator will be 0, and hence X' is 0.
- Likewise, when the value of X for an observation is the maximum value in the column, the numerator is equal to the denominator and thus the value of X' will be 1.
- If the value of X is between the minimum and the maximum value, then the value of X' is between 0 and 1.

Standardization is another scaling technique where the values are cantered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation } (x)}$$

Formula for Standardization

Note that in the case of standardization, the scaled values are not restricted to a particular range.

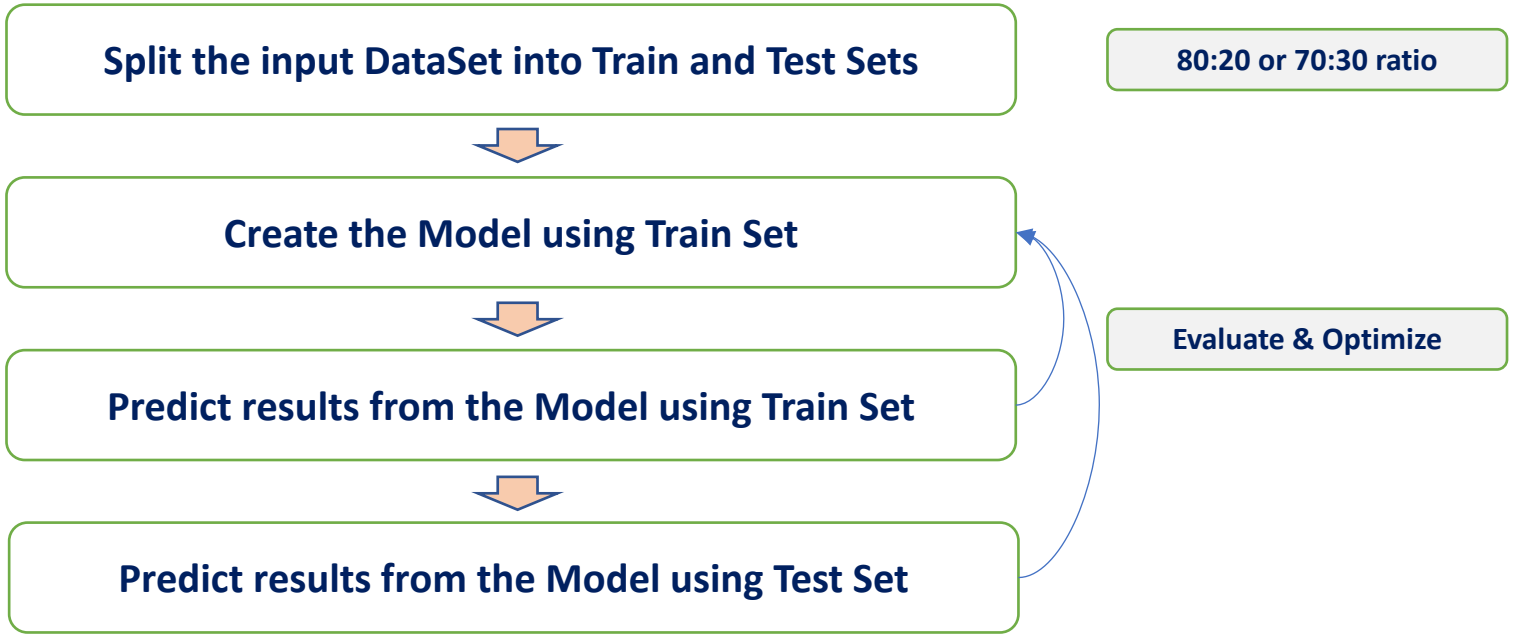
Feature Scaling - Choice

Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest-Neighbours and Neural Networks.

Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

Splitting the Data into Train and Test Sets

The whole process of creating and establishing a Machine Learning model comprise of the following steps as regards to the data:



OLS Assumptions

Assumption of Linearity: OLS assumes that a Linear Relationship exists between the predictors, error term and the target variables. OLS estimates the co-efficients of the predictors (X-variables) with this assumption.

The Linearity is determined by the existence of an equation like this:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 \ldots\ldots + \beta_n x_n + \varepsilon$$

Here, ε is the random error.

Error Term has a Population Mean of Zero: Error term is the difference between the observed and predicted valued of the target variable. The individual data point level differences are random chance errors and not explained by the predictors. On an average, the error term should be zero. E.g. if the mean of the error term is +5 (actual – predicted), that indicates that the model consistently under-predicts the target variable and that's undesirable.

All Independent (Predictor) Variables are Uncorrelated to the Error Term: If the Independent variable is correlated to the error term then we can use the independent variable to control the error term. This violates the notion of Error Term being random chance errors. This assumption is called Exogeneity. Existence of this kind of correlation is called Endogeneity. Endogeneity leads to bias in the calculated coefficients.

OLS Assumptions

Error Terms are Uncorrelated to each other: there should not be any pattern of dependency between the values of the error terms, i.e. the value of an individual random discrepancy between observed and predicted values must not be in any way correlated to another value of discrepancy. Existence of such serial correlation in error terms reduces the accuracy of the model.

The Error Term has a Constant Variance: The variance of errors should be consistent for all observations, i.e. the variance should not change for each predicted value of the target variable or there should not be a dependency pattern between the error term variance and predicted values. This condition or assumption is called **Homoscedasticity** (meaning constant variance). The way to determine Homoscedasticity is to plot the error terms or residuals against the fitted values.

Absence of Multicollinearity: In a Linear regression model, there mustn't be predictors that explain other predictors in a Linear function. The presence of this phenomena is called multicollinearity and is confusing for the model. E.g. Presence of a monetary value in two different currencies is multicollinearity and one must be removed for the model to be able to work effectively.

Residual Normality: The Error terms should be Normally Distributed. However, this assumption is optional. Compliance to this assumption helps in performing Hypothesis Testing on the predictor coefficients.

Interpreting Model Results

- **R-Squared**: This is also known as the 'Coefficient of Determination'. R-Squared is the measure of how well the regression model approximates the observed values. It has a value range of 0 – 1, where 0 indicates no explaining power of the model and 1 indicates perfect explaining power. A value closer to 1 is desirable. Higher the R-Squared, higher the variations explained by the input variables and better the model.

- **Adjusted R-Squared**: One of the issues with R-Squared measure is that it will either remain the same or increase with every increase of the number of input variables. However, increase in variable is not always desirable as it may tend to make the model more complex and less generalizable. Adjusted R-Square is a measure that penalizes the increase in variables that do not improve your model. Hence, the Adjusted R-Squared is a better measure of explainability of the model.

$$Adjusted\ R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

[where N is the Sample size and p is number of predictors]

- **F-Statistic:** The F-Statistic is the ratio of the mean regression sum of squares divided by the mean error sum of squares. Its value can range from zero to an arbitrarily large number.

- **Probability F-Statistic:** This is the probability that the Null-Hypothesis on the model is True. The Null-Hypothesis on the model is that the coefficients of all predictor parameters are zero. In our case, the Prob (F-Statistic) is 0 indicating that the Null hypothesis on the model should be rejected.

Interpreting Model Results

- **‘coeff’**: These are the coefficient values against each individual predictor. These coefficients make up the Linear Equation that defines the model.

- **P-Value (P > |t|):** These are individual Probability values against the Null Hypothesis on the concerned predictor. The Null Hypothesis on the predictor is that the coefficient is 0, meaning that the predictor doesn’t explain the target variable values. If the P-Value is lesser than the confidence level (commonly a benchmark value of 0.05 is taken), it indicates that this predictor is statistically significant to explain the target variable values (that the Null Hypothesis can be rejected, and the coefficient / relationship is statistically significant).

Eliminating Multicollinearity using VIF

Multicollinearity is the phenomena of **how well a predictor variable is explained by other predictor variables in the dataset**. Multicollinearity is undesirable in a Linear Regression model as in presence of it, the model is unable to determine and isolate the influence of each variable over the target variable values. E.g. The presence of Income of households and monthly expenditure will be confounding as we expect expenditure to be high if income is high. In such case, only one of the variables must be selected as a predictor.

There are several ways multicollinearity can creep into datasets. Some of them are:

- The input dataset already contains such variables that have association.
- Creating derived variables and retaining the original ones. E.g. Length in ‘cm’ is present in the dataset and another field is created for Length in ‘inches’. Only one of them must be retained.
- Not rationalizing the Dummy variables. Refer to the section on Dummy variables and the need to remove one of the Dummy variables from the fields created due to the complimentary nature of the values.

One of the common ways to eliminate Multicollinearity is VIF or Variance Inflation Factor. The VIF method takes up each field from the dataset at a time and regress it against the other variables present thereby calculating the R-Squared value of the variable. VIF is then calculated as:

$$VIF = \frac{1}{1 - R^2}$$

Here, the R-Squared value is not the R-Squared value of the model but of the Variable being regressed to get the VIF value. Accordingly, each variable will have an individual VIF value.

Commonly, a value of 5 and below is considered to say that it does not have Multicollinearity.

Recursive Feature Elimination

Recursive Feature Elimination, or RFE for short, is a feature selection algorithm.

Feature selection refers to techniques that select a subset of the most relevant features (columns) for a dataset. Fewer features can allow machine learning algorithms to run more efficiently (less space or time complexity) and be more effective. Some machine learning algorithms can be misled by irrelevant input features, resulting in worse predictive performance.

RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains.

This is achieved by

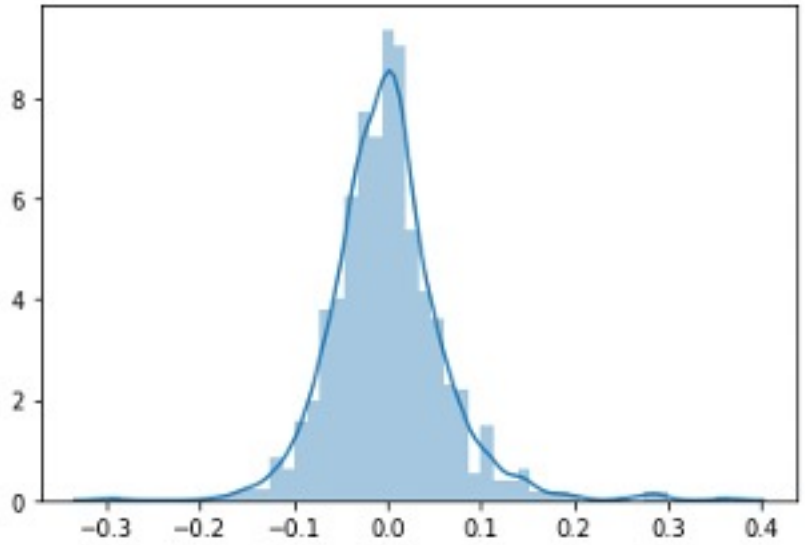
- **Fitting the given machine learning algorithm used in the core of the model**
- **Ranking features by importance, discarding the least important features, and re-fitting the model**

This process is repeated until a specified number of features remains.

Testing your Linear Regression Model

Residual Analysis:

One of the assumptions of Linear Regression is that the Residuals (differences between observed and predicted values for each individual data point) are normally distributed. In the following code, we create a Numpy Series object with the residual values of our example and plot a distribution using Seaborn library - plot method.



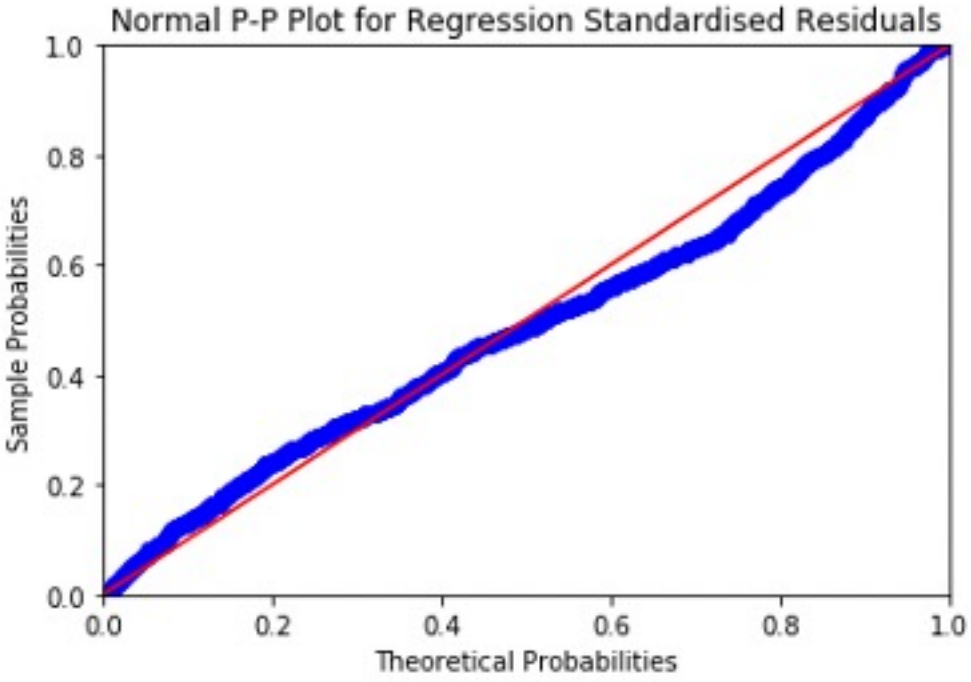
Testing the Model

Residual Normality

The P-P (Probability-Probability) plot further establishes the normality of the Residuals.

The normal probability plot or P-P Plot is a graphical technique for assessing whether or not a data set is approximately normally distributed. (here the dataset indicates the error terms.

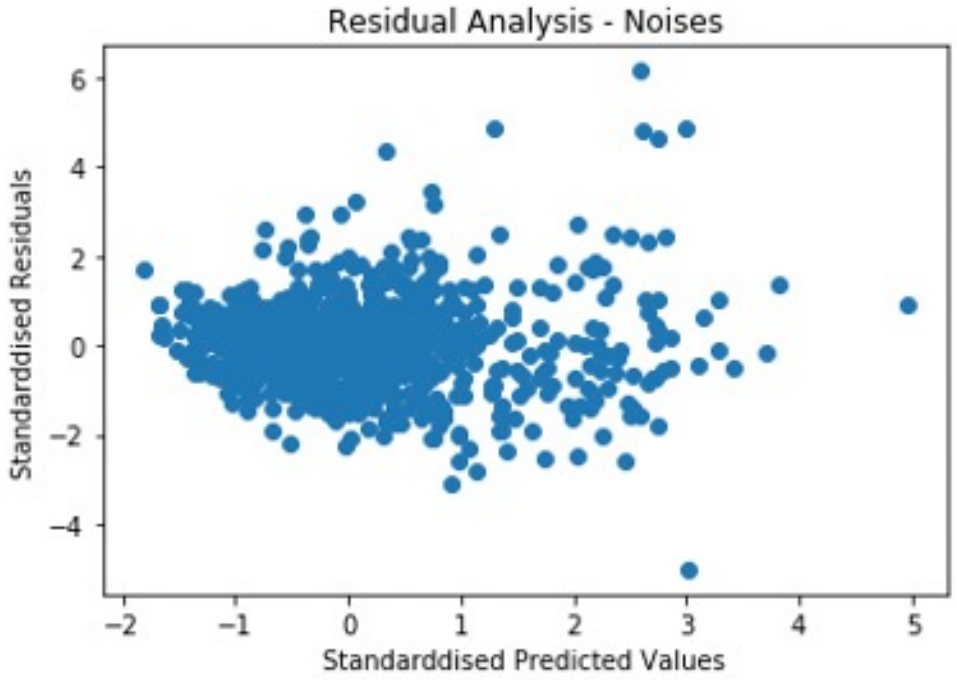
The data (Error terms) are plotted against a theoretical normal distribution in such a way that the points should form an approximate straight line.



Testing the Model

Test of Homoscedasticity of residuals:

Variance of the residuals should not change for each predicted value of the target variable or there should not be a dependency pattern between the error term variance and predicted values. This is determined by plotting the residuals against the predicted values.



Practice

PYTHON DEMO
CAR PRICE PREDICTION