**Note: Data Analysis with Python**

Source: <https://dev.to/mage_ai/10-steps-to-build-and-optimize-a-ml-model-4a3h>

The above link will give emphasis on how to build a machine learning model and will help you understand all the processes required to build a machine learning model.

**Week 1:**

1. Defining Problem Statement
   1. Setting goals and objectives
   2. Analytical Approach
2. Data Requirements & Data Collection

**Week 2:**

**Data Wrangling:**

Data wrangling is the process of converting data from the initial format to a format that may be better for analysis.

1. Identify and handle missing values
2. Data Formatting
3. **Data Normalization**: (centering/scaling):
   1. Data normalization can be defined as a process designed to facilitate a more cohesive form of data entry, essentially ‘cleaning’ the data. When you normalize a data set, you are reorganizing it to remove any unstructured or redundant data to enable a superior, more logical means of storing that data.
   2. The main goal of data normalization is to achieve a standardized data format across your entire system. This allows the data to be queried and analyzed more easily, which can lead to better business decisions.
   3. Examples
      1. Miss ANNA will be written Ms. Anna
      2. 4158488400 will be written 415-848-8400
      3. 37 buttercup AVE will be written 37 Buttercup Avenue
      4. Amazon will be written Amazon.com, Inc.
      5. VP product will be written Vice President of Product
      6. Price range should be between 5000 – 10000. Over/under these values, column won’t store the values.
4. **Data Binning :** Data binning, also called data discrete binning or data bucketing, is a data pre-processing technique used to reduce the effects of minor observation errors. The original data values which fall into a given small interval, a bin, are replaced by a value representative of that interval, often a central value.
5. **Turning Categorical values into numeric variables**

**Data Normalization Methods/Techniques:**

This technique is very important to understand the data in data pre-processing process.

When it comes to build a model that predict the outcome of variables, if needed, the data normalization is very important and it can be achieved through several techniques such as;

1. **Simple Feature Scaling**

In this method, simply divide each value by maximum value of the feature and formula would be:

**Xnew = Xold / Xmax**

1. **Min-Max Method**

In this method, we simply divide the difference of Xold and Xmin to the difference of Xmax and Xmin.

**Xnew = (Xold – Xmin) / (Xmax – Xmin)**

1. **Z-score or Standard Score (Data Standardization)**

In this method, we divide the difference of Xold and Xavg with the standard deviation of the feature variable.

**Xnew = (Xold – Xavg) / Xstd**

**Or** ( **Note**: The resulting values hover around zero, and typically range between -3 to +3 but can be higher or lower.) This process is also called as **Standardization**.

**Xnew = (Xold – Xavg) / standard deviation**

**Note:** All these methods result in the providing the Xnew value between 0 to 1.

**Data Binning:**

**Data Standardization**

Data is usually collected from different agencies in different formats. (Data standardization is also a term for a particular type of data normalization where we subtract the mean and divide by the standard deviation.)

**What is standardization?**

Standardization is the process of transforming data into a common format, allowing the researcher to make the meaningful comparison.

**Example**

Transform mpg to L/100km:

In our dataset, the fuel consumption columns "city-mpg" and "highway-mpg" are represented by mpg (miles per gallon) unit. Assume we are developing an application in a country that accepts the fuel consumption with L/100km standard.

We will need to apply **data transformation** to transform mpg into L/100km.

## Data Normalization

**Why normalization?**

Normalization is the process of transforming values of several variables into a similar range. Typical normalizations include scaling the variable so the variable average is 0, scaling the variable so the variance is 1, or scaling the variable so the variable values range from 0 to 1.

**Example**

To demonstrate normalization, let's say we want to scale the columns "length", "width" and "height".

**Target:** would like to normalize those variables so their value ranges from 0 to 1

**Approach:** replace original value by (original value)/(maximum value)

**Binning**

**Why binning?**

Binning is a process of transforming continuous numerical variables into discrete categorical 'bins' for grouped analysis.

**Example:**

In our dataset, "horsepower" is a real valued variable ranging from 48 to 288 and it has 59 unique values. What if we only care about the price difference between cars with high horsepower, medium horsepower, and little horsepower (3 types)? Can we rearrange them into three ‘bins' to simplify analysis?

We will use the pandas method 'cut' to segment the 'horsepower' column into 3 bins.

**Indicator Variable (or Dummy Variable)**

**What is an indicator variable?**

An indicator variable (or dummy variable) is a numerical variable used to label categories. They are called 'dummies' because the numbers themselves don't have inherent meaning.

**Why we use indicator variables?**

We use indicator variables so we can use categorical variables for regression analysis in the later modules.

**Example**

We see the column "fuel-type" has two unique values: "gas" or "diesel". Regression doesn't understand words, only numbers. To use this attribute in regression analysis, we convert "fuel-type" to indicator variables.

We will use pandas' method 'get\_dummies' to assign numerical values to different categories of fuel type.

**Week 3:**

**Chi-Square Test:**

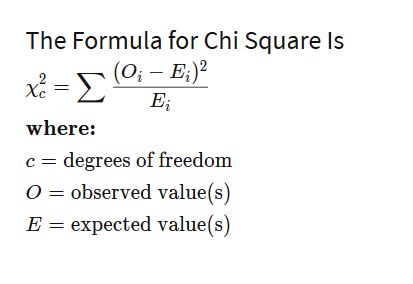
The Chi-square test is intended to test how likely it is that an observed distribution is due to chance. It measures how well the observed distribution of data fits with the distribution that is expected if the variables are independent.

The Chi-square tests a null hypothesis that the variables are independent. The test compares the observed data to the values that the model expects if the data was distributed in different categories by chance.

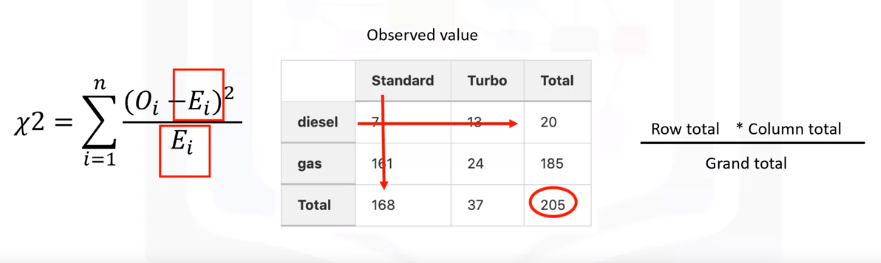
Anytime the observed data doesn't fit within the model of the expected values, the probability that the variables are dependent becomes stronger, thus proving the null hypothesis incorrect.

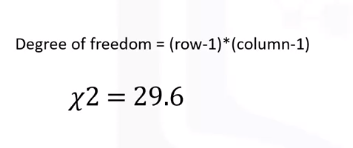
The Chi-square does not tell you the type of relationship that exists between both variables, but only that a relationship exists.

A crosstab is a table showing the relationship between two or more variables. When the table only shows the relationship between two categorical variables, the crosstab is also known as a contingency table.



How to calculate expected values for Chi-square test:





**If p <0.05, then we reject the NULL Hypothesis, meaning that two variables are independent to each other and conclude that there is evidence of association between these two variables.**

## Continuous Numerical Variables:

Continuous numerical variables are variables that may contain any value within some range. They can be of type "int64" or "float64". A great way to visualize these variables is by using scatterplots with fitted lines.

In order to start understanding the (linear) relationship between an individual variable and the price, we can use "regplot" which plots the scatterplot plus the fitted regression line for the data.

### **ANOVA: Analysis of Variance**

The Analysis of Variance (ANOVA) is a statistical method used to test whether there are significant differences between the means of two or more groups. ANOVA returns two parameters:

**F-test score**: ANOVA assumes the means of all groups are the same, calculates how much the actual means deviate from the assumption, and reports it as the F-test score. A larger score means there is a larger difference between the means.

**P-value**: P-value tells how statistically significant our calculated score value is.

If our price variable is strongly correlated with the variable we are analyzing, we expect ANOVA to return a sizeable F-test score and a small p-value.

**Pearson Correlation Co-efficient and P-Value:**

**P-value**

What is this P-value? The P-value is the probability value that the correlation between these two variables is statistically significant. Normally, we choose a significance level of 0.05, which means that we are 95% confident that the correlation between the variables is significant.

By convention, when the

**p-value is <0.001:** we say there is strong evidence that the correlation is significant

**the p-value is < 0.05:** there is moderate evidence that the correlation is significant.

**the p-value is < 0.1:** there is weak evidence that the correlation is significant.

**the p-value is > 0.1:** there is no evidence that the correlation is significant.

We can obtain this information using "stats" module in the "scipy" library.

**Week 4:**

**Model Development:**

A model or estimator can be thought of as a mathematical equation used to predict the value given one or more other values. Relating one or more independent variables or features to dependent variables.

Usually, the more relevant data you have, the more accurate your model is.

Simplest form of regression models are :   
  
Simple Linear Regression Model

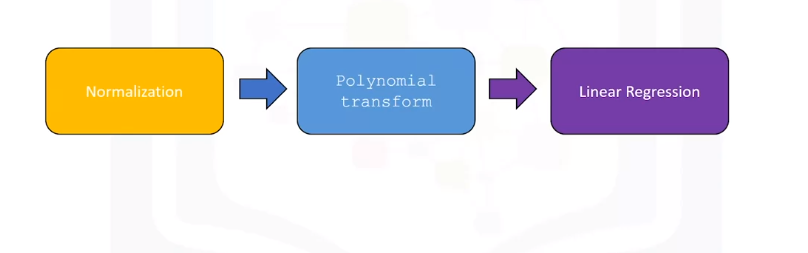
Multiple Linear Regression Model

Polynomial Regressions Model

**Polynomial Regression:**

Polynomial regression is a special case of general linear regression. For this type of regression, we transform our data into a polynomial, then use linear regression to fit the parameters.

There are many steps involved when building an accurate machine learning model such as ;



**Determining a Good Model Fit:**

How can we determine if the predictive model is correct?

When building a best-fit predictive model, do consider the following points;

1. Make sure the predicted values make sense
2. Do the visualization of results such as visualization of residual errors, normal distribution curve or bell-curved visualization, visualization of actual values, predicted values, and mean along with residuals and such.
3. Make sure to create a scatter plot along with best-fitted line which will help in determining the best fit model.
4. Comparing between different models

**SLOPE in Regression Model:**

In a regression context, the slope is the heart and soul of the equation because it tells you how much you can expect Y to change as X increases. ***In general, the units for slope are the units of the Y variable per units of the X variable. It's a ratio of change in Y per change in X.***

* a refers to the **intercept** of the regression line, in other words: the value of Y when X is 0
* b refers to the **slope** of the regression line, in other words: the value with which Y changes when X increases by 1 unit

For example: Predicting the price(dependent variable/target variable/response variable or predicated values) of a car based on highway-mpg(independent variable/predictor variable/explanatory variable).

Price = 38423.31 – 821.75 \* highway-mpg (Simple linear model equation)

How to interpret the above equation:

This value (821.75) corresponds to the multiple of the highway miles per gallon feature. As such, an increase of one unit in highway miles per gallon, the value of the car decreases approximately 821 dollars. This value also seems reasonable.

***Simply: one unit of increase in X (explanatory variable) responds to unit increase or decrease in Y.***

**Note:** The mean square error is perhaps the most intuitive numerical measure for determining if a model is good or not.

**R-Squared:**

It tells you how well your line fits into the model. R - squared values range from 0 - 1. R - squared tells us what percent of the variability in the dependent variable is accounted for by the regression on the independent variable.

An R - squared of 1 means that all movements of dependent variable are completely explained by movements in the independent variables.

# Lesson Summary

In this lesson, you have learned how to:

**Define the explanatory variable and the response variable:** Define the response variable (y) as the focus of the experiment and the explanatory variable (x) as a variable used to explain the change of the response variable. Understand the differences between Simple Linear Regression because it concerns the study of only one explanatory variable and Multiple Linear Regression because it concerns the study of two or more explanatory variables.

**Evaluate the model using Visualization:** By visually representing the errors of a variable using scatterplots and interpreting the results of the model.

**Identify alternative regression approaches:** Use a Polynomial Regression when the Linear regression does not capture the curvilinear relationship between variables and how to pick the optimal order to use in a model.

**Interpret the R-square and the Mean Square Error:** Interpret R-square (x 100) as the percentage of the variation in the response variable y  that is explained by the variation in explanatory variable(s) x. The Mean Squared Error tells you how close a regression line is to a set of points. It does this by taking the average distances from the actual points to the predicted points and squaring them.

**Week 5: Model Evaluation**

In simple evaluation tells us how well our model will fit the data used to train it….but it does not give us an estimate of how well the train model can predict new data.

Then we implement the solution, which is to split our data up, use the **in-sample data or training data** to train the model. The rest of the data, called **Test Data**, is used as out-of-sample data.

This data is then used to approximate how the model performs in the real world. Separating data into training and testing sets is an important part of model evaluation.

We use the test data to get an idea how our model will perform in the real world. When we split a dataset, usually the larger portion of data is used for training and a smaller part is used for testing.

**For example, we can use 70 % of the data for training. We then use 30 % for testing.**

We use training set to build a model and discover predictive relationships. We then use a testing set to evaluate model performance. When we have completed testing our model, we should use all the data to train the model.

## Note:

The random state hyperparameter in the **train\_test\_split()** function controls the shuffling process.

With **random\_state=None** , we get different train and test sets across different executions and the shuffling process is out of control.

With **random\_state=0** , we get the same train and test sets across different executions.

With **random\_state = 1**, it will produce the same splitting datasets.

It doesn't matter if the random\_state is 0 or 1 or any other integer. What matters is that it should be set the same value, if you want to validate your processing over multiple runs of the code. By the way I have seen random\_state=42 used in many official examples of scikit as well as elsewhere also.

random\_state as the name suggests, is used for initializing the internal random number generator, which will decide the splitting of data into train and test indices in your case.

**Generalization Performance/Error:**

Generalization Error is a measure of how good our data is at predicting previously unknown/unseen data. The error we obtain using our testing data is an approximation of this error.

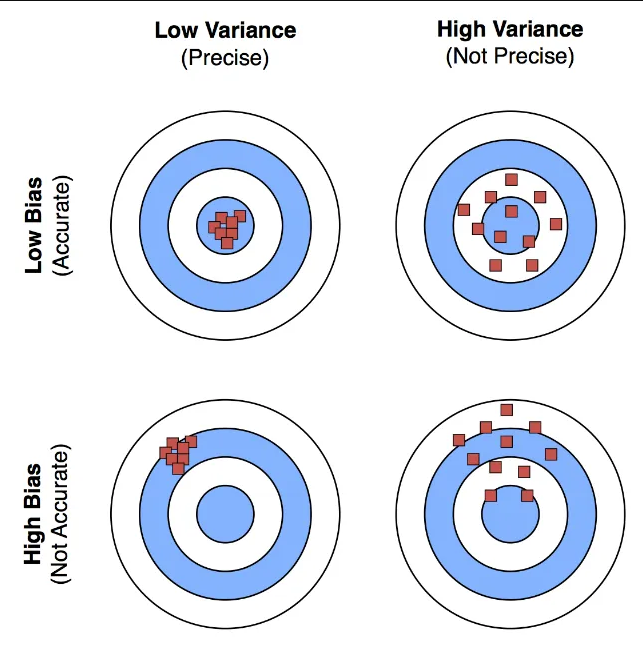
Finally, in order to obtain more predictive performance measures, we can use many different train/test partitions and average the results. This is called cross-validation.

A good approach to evaluating the generalization error is to experiment with training model with a different combination of samples.

The closer the values to the center of generalization error, the better the model will be but if the precision between the predicted values for different combinations of samples fall far apart then, then there will be a chance to improve the model.

There could be two scenarios based on the results;

1. **First Scenario:** Results that we get from experimenting with the training and testing data, could be **closer to the generalization error** but distinct from each other or the precision is poor. (Resulting in High Variance)
2. **Second Scenario:** Results that we get from experimenting with the training and testing data, could be **further from the generalization error** but are closer to each other (results from the samples are closer to each other); meaning that precision is poor. (Resulting in High Bias)

****

**Also look for more details :** [**https://medium.com/@yixinsun\_56102/understanding-generalization-error-in-machine-learning-e6c03b203036**](https://medium.com/@yixinsun_56102/understanding-generalization-error-in-machine-learning-e6c03b203036)

[**https://campus.datacamp.com/courses/machine-learning-with-tree-based-models-in-python/the-bias-variance-tradeoff?ex=1**](https://campus.datacamp.com/courses/machine-learning-with-tree-based-models-in-python/the-bias-variance-tradeoff?ex=1)

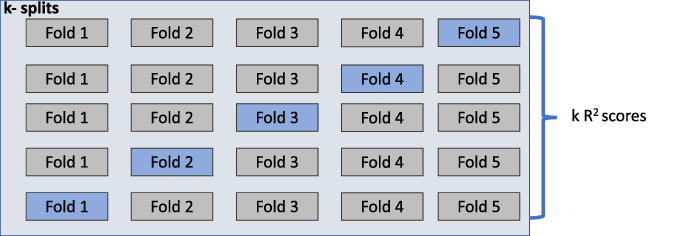
[**https://medium.com/@jwbtmf/generalization-error-in-machine-learning-4617141932b7**](https://medium.com/@jwbtmf/generalization-error-in-machine-learning-4617141932b7)

To overcome this issue, we need to use cross-validation.

**Cross-Validation: (cross\_val\_score / cross\_val\_predict):**

One of the most common out of sample evaluation metrics/techniques is cross-validation.

**Cross-validation** is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data. Use cross-validation to detect overfitting, ie, failing to generalize a pattern. Example given below for cv=5 (means 5 kfolds)



**Why Use Cross-Validation?**

*The purpose of cross–validation is to test the ability of a machine learning model to predict new data*. It is also used to flag problems like overfitting or selection bias and gives insights on how the model will generalize to an independent dataset.

**Model Fit: Overfitting, Underfitting, and Model Selection:**

Understanding model fit is important for understanding the root cause for poor model accuracy.

We can determine whether a predictive model is underfitting or overfitting the training data by looking at the prediction error on the training data and the evaluation data.

**Underfitting:** Your model is underfitting the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y).

**Overfitting:** Your model is overfitting your training data when you see that the model performs well on the training data but does not perform well on the evaluation data. This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.

One of the regression techniques is polynomial regression where order or degree of polynomial regression plays a crucial role in the analysis of prediction.

Selecting the accurate order of polynomial regression is very important and we can say; ***the goal of Model Selection is to determine the order of the polynomial to provide the best estimate of the function y(x).***

**For Underfitting:** Poor performance on the training data could be because the model is too simple (the input features are not expressive enough) to describe the target well. Performance can be improved by increasing model flexibility. To increase model flexibility, try the following:

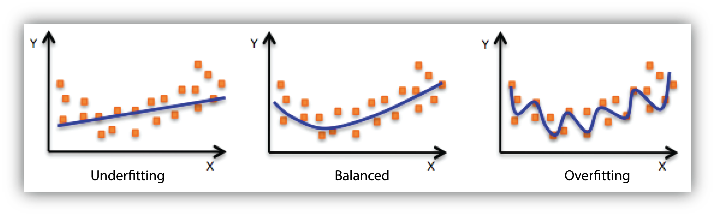
* Add new domain-specific features and more feature Cartesian products, and change the types of feature processing used (e.g., increasing n-grams size)
* Decrease the amount of regularization used

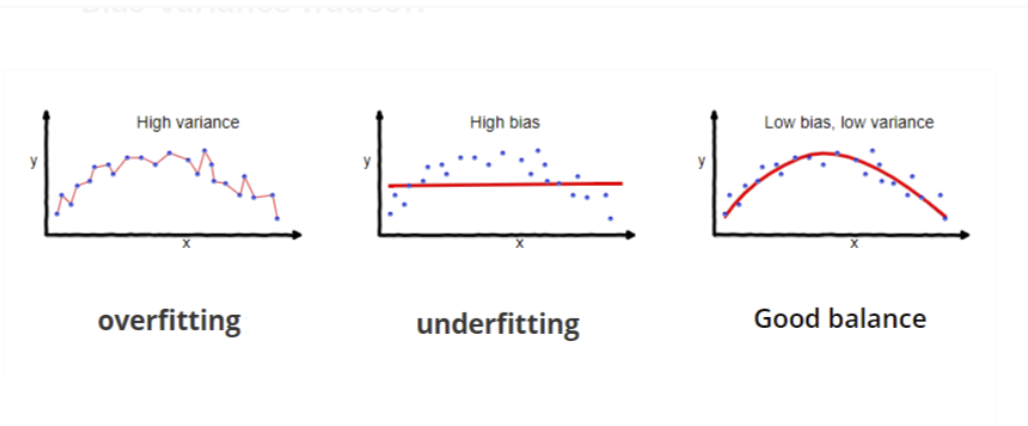
**For Overfitting:** If your model is overfitting the training data, it makes sense to take actions that reduce model flexibility. To reduce model flexibility, try the following:

* Feature selection: consider using fewer feature combinations, decrease n-grams size, and decrease the number of numeric attribute bins.
* Increase the amount of regularization used.

Accuracy on training and test data could be poor because the learning algorithm did not have enough data to learn from. You could improve performance by doing the following:

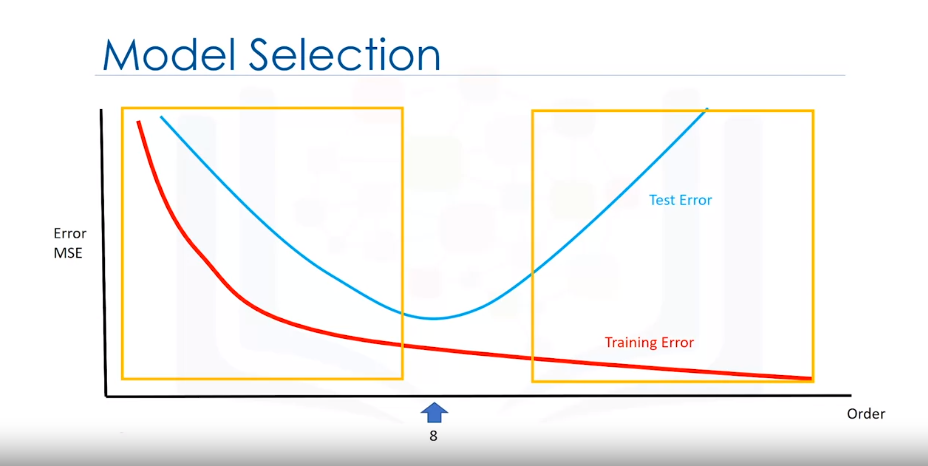
* Increase the amount of training data examples.
* Increase the number of passes on the existing training data.





When the model is too flexible and fits the noise but doesn’t follow path of actual function or doesn’t fit the actual function, then it is OVERFITTING.

**How to select correct polynomial ORDER when choosing the models which avoid overfitting or underfitting?**



Overfitting

Underfitting

Let's look at a plot of the mean square error for the training and testing set of different order polynomials. The horizontal axis represents the order of the polynomial. The vertical axis is the mean square error.

**Red Line :** represents the metrics (MSE and polynomial order) for Training set.

**Blue Line:** represents the metrics for Testing set (Test dataset).

As we can see that the training error decreases with the order of the polynomial (for training set) while test error decreases till the best order of the polynomial is determined and then test error starts to increase again.

Here, we select the order of polynomial that minimizes the test error (blue line) .

**Ridge Regression:**

Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

The cost function for both ridge and lasso regression are similar. However, ridge regression takes the square of the coefficients and lasso takes the magnitude.

The difference between ridge and lasso regression is that it tends to make coefficients to absolute zero as compared to Ridge which never sets the value of coefficient to absolute zero.

The cost function for ridge regression:

**Min(||Y – X(theta)||^2 + λ||theta||^2)**

**Multicollinearity** is a statistical concept where several independent variables in a model are correlated. Two variables are considered perfectly collinear if their correlation coefficient is +/- 1.0. Multicollinearity among independent variables will result in less reliable statistical inferences.

**Ridge regression** is a regression that is employed in a Multiple regression model when Multicollinearity occurs. Multicollinearity is when there is a strong relationship among the independent variables. Ridge regression is very common with polynomial regression.

Ridge Regression prevents overfitting. Overfitting is also a big problem when you have multiple independent variables, or features.

**Ridge regression controls the magnitude of these polynomial coefficients by introducing the hyperparameter alpha.**

Alpha is a parameter we select before fitting or training the model. Alpha must be selected carefully. If alpha is too large, the coefficients will approach zero and underfit the data.

If alpha is zero, the overfitting is evident.

For alpha equal to 0.001, the overfitting begins to subside.

For Alpha equal to 0.01, the estimated function tracks the actual function.

When alpha equals one, we see the first signs of underfitting.

The estimated function does not have enough flexibility.

At alpha equals to 10, we see extreme underfitting. It does not even track the two points.

In order to select alpha, we use cross validation

A regression model that uses L1 regularization technique is called Lasso Regression and model which uses L2 is called Ridge Regression.

From a practical standpoint, L1 tends to shrink coefficients to zero whereas L2 tends to shrink coefficients evenly. L1 is therefore useful for feature selection, as we can drop any variables associated with coefficients that go to zero. L2, on the other hand, is useful when you have collinear/codependent features.

L2 regularization distributes the weights for all layers by gaussian distribution and shrinks their norm to small values. L2 Regularization shrinks all the weights to small values, preventing the model from learning any complex concept wrt. any particular node/feature, thereby preventing overfitting.

**L1 Regularization**, also called a lasso regression, adds the “absolute value of magnitude” of the coefficient as a penalty term to the loss function**. L2 Regularization**, also called a ridge regression, adds the “squared magnitude” of the coefficient as the penalty term to the loss function

**Lasso Regression Stuff link:**

<https://www.youtube.com/watch?v=bPFjfZWWQO0>

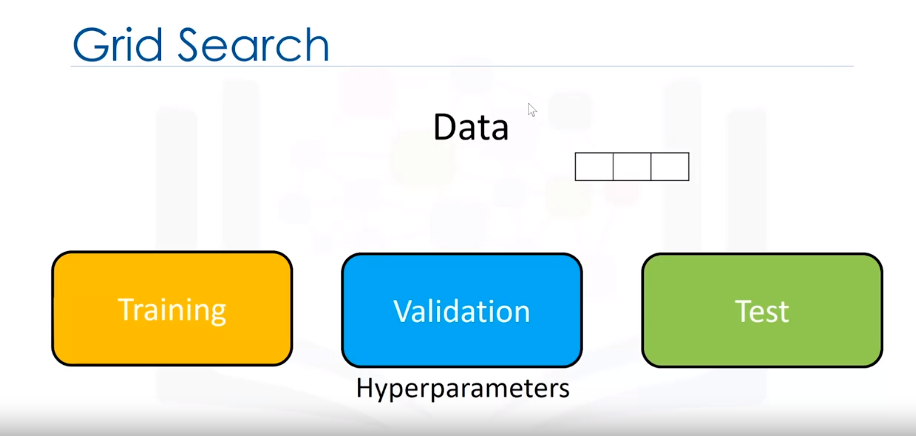
**Hyperparameters:**

Grid Search allows us to scan through multiple free parameters with few lines of code. Parameters like the **alpha** term are not part of the fitting or training process. *These values are called hyperparameters*.

**Grid Search** takes the model or objects you would like to train and different values of the hyperparameters. It then calculates the mean square error or R-squared for various hyperparameter values, allowing you to choose the best values.

Steps to perform the ridge regression using hyperparameters:

* 1. First, we need to select our data
  2. To select the correct hyperparameter, we split our dataset into three parts, **the training set, validation set, and test set.**
  3. We then train the model for different hyperparameters.
  4. Then we use the R-squared or mean square error for each model we previously trained using hyperparameters.
  5. and then we select the *hyperparameter that minimizes* the mean squared error or *maximizes the R-squared* on the validation set.
  6. and finally, we test our model performance using the test data



# Lesson Summary

In this lesson, you have learned how to:

**Identify over-fitting and under-fitting in a predictive model:** Overfitting occurs when a function is too closely fit to the training data points and captures the noise of the data. Underfitting refers to a model that can't model the training data or capture the trend of the data.

**Apply Ridge Regression to linear regression models:** Ridge regression is a regression that is employed in a Multiple regression model when Multicollinearity occurs.

**Tune hyper-parameters of an estimator using Grid search:** Grid search is a time-efficient tuning technique that exhaustively computes the optimum values of hyperparameters performed on specific parameter values of estimators.

**Feature Engineering:**

This process involves transforming the data and adjusting the features as per business problem statement.

Or Simple put:

A feature refers to a column in a dataset, while engineering can be manipulating, transforming, or constructing, together they’re known as Feature Engineering. Simply put, Feature Engineering is nothing but transforming existing features or constructing new features

Example: Handling missing values with average/max, handling outliers, handling skewness, encoding scaling down the features, creating new features from the existing features and etc.

### **Types of Features in Machine Learning**

Features in machine learning can roughly be termed as the building blocks of any machine learning model and the input variables that a machine learning algorithm uses to make predictions or decisions. Here are the different types of features in machine learning-

* **Numerical Features-** These features are continuous values that can be measured on a scale. Examples of numerical features include age, height, weight, and income. Numerical features can be used in machine learning algorithms directly.
* **Categorical Features-**These features are discrete values that can be grouped into categories. Examples of categorical features include gender, color, and zip code. Categorical features in machine learning typically need to be converted to numerical features before they can be used in machine learning algorithms. You can easily do this using one-hot, label, and ordinal encoding.
* **Time-series Features-**These features are measurements that are taken over time. Time-series features include stock prices, weather data, and sensor readings. You can use these features to train machine learning models that can predict future values or identify patterns in the data.
* **Text Features-** These features are text strings that can represent words, phrases, or sentences. Examples of text features include product reviews, social media posts, and medical records. You can use text features to train machine learning models that can understand the meaning of text or classify text into different categories.

Data Scientists spend 80% of their time doing automated feature engineering because a time-consuming and difficult. Understanding features and the various techniques to deconstruct this art can ease the complex automated feature engineering process.

**Feature Selection:**

**Supervised:** It refers to the method which uses the output label class for feature selection.

**Unsupervised:** it refers to the method which doesn’t need the output label class for feature selection.

There are three types of Supervised feature selection:

* Filter Method
* Wrapper Method
* Filter Method

