Some interview questions

MAXIMOVSKAYA ANASTASIA

Feature Selection

What are some common Machine Learning problems that Unsupervised Learning can help with?

Insufficient labeled data

- For supervised learning, there is a requirement for a lot of labeled data for the model to perform well.
- Unsupervised learning can automatically label unlabeled examples.
- This would work by clustering all the data points and then applying the labels from the labeled ones to the unlabeled ones..

What are some common Machine Learning problems that Unsupervised Learning can help with?

Overfitting

- Machine learning algorithms can sometimes overfit the training data by extracting too much from the noise in the data.
- When this happens, the algorithm is memorizing the training data rather than learning how to generalize the knowledge of the training data.
- Unsupervised learning can be introduced as a regularizer. Regularization is a process that helps to reduce the complexity of a machine learning algorithm, helping it capture the signal in the data without adjusting too much to the noise.

What are some common Machine Learning problems that Unsupervised Learning can help with?

Outliers

- The quality of data is very important. If machine learning algorithms train on outliers (rare cases) then their generalization error will be lower than if they are ignored.
- Unsupervised learning can perform outlier detection using dimensionality reduction and create solutions specifically for the outliers, and separately, a solution for the normal data.

What are some common Machine Learning problems that Unsupervised Learning can help with?

Feature Engineering

- Feature engineering is a vital task for data scientists to perform, but feature engineering is very labor-intensive, and it requires a human to creatively engineer the features.
- Representation learning from unsupervised learning can be used to automatically learn the right type of features to help the task at hand.

What's the difference between Feature Engineering vs. Feature Selection?

Feature engineering allows us to create new features from the ones we already have in order to help the machine learning model make more effective and accurate predictions. Some of the tasks that imply feature engineering are:

- Filling missing values within a variable.
- Encoding categorical variables into numbers.
- Variable transformation.
- Creating or extracting new features from the ones available in the dataset.

What's the difference between Feature Engineering vs. Feature Selection?

Feature selection, on the other hand, allows us to select features from the feature pool (including any newly-engineered ones) that will help machine learning models make predictions on target variables more efficiently.

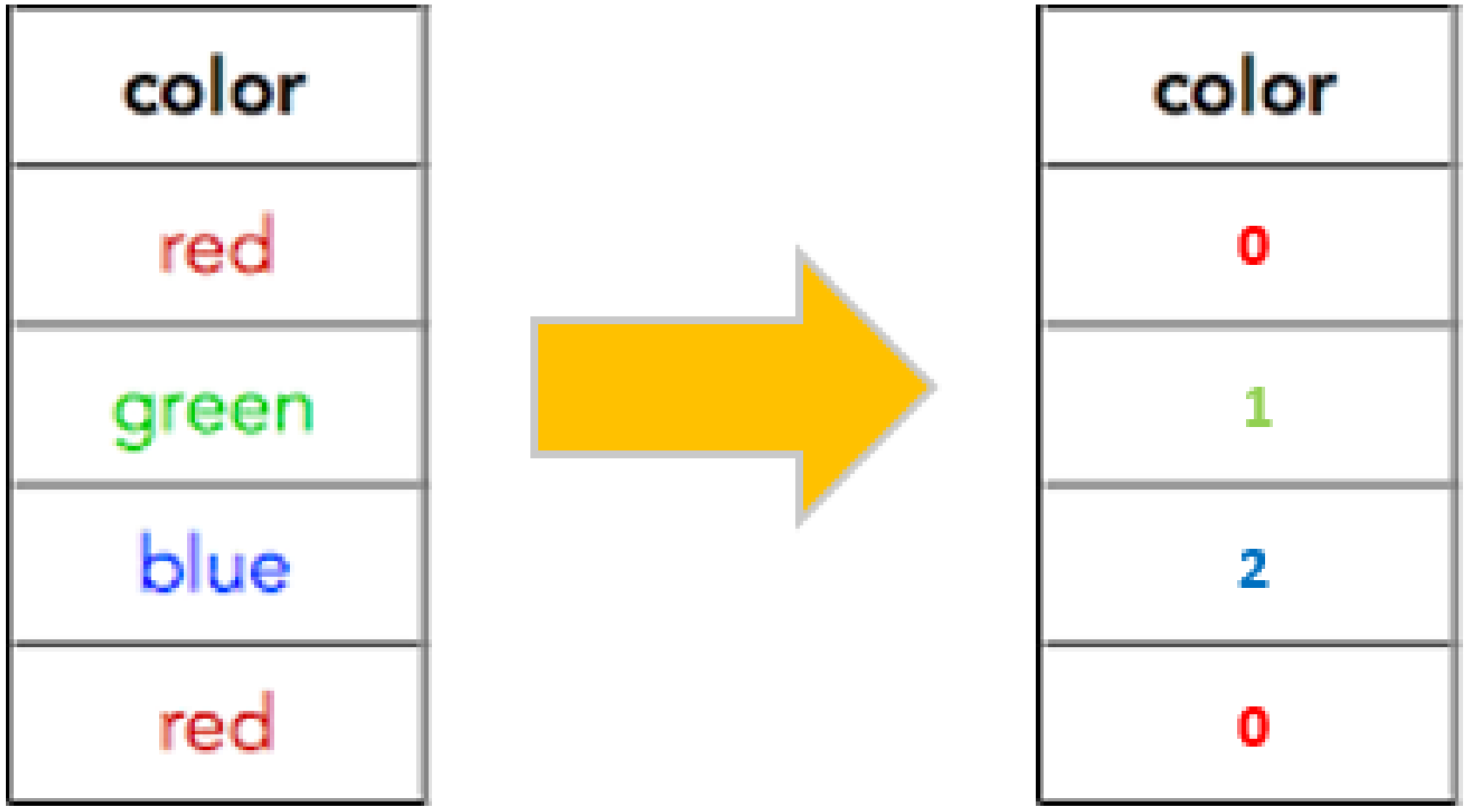
For this, two common methods used are wrapper and filter methods. These methods are almost always supervised and are evaluated based on the performance of a resulting model on a holdout dataset.

In a typical machine learning pipeline, we perform feature selection after completing feature engineering.

Name some benefits of Feature Selection

1. Many features and low samples/features ratios will introduce noise into your dataset. In such a case your algorithm is likely to overfit and give you a false feeling of good performance.
2. Reducing the number of features will reduce the running time in the later stages. That in turn will enable you to use algorithms of higher complexity, search for more hyperparameters or do more evaluations.
3. A smaller set of features is more comprehendible to humans. That will enable you to focus on the main sources of predictability and do more exact feature engineering. If you will have to explain your model to a client, you are better at presenting a model with 5 features than a model with 200 features.

Explain One-Hot Encoding and Label Encoding. Does the dimensionality of the dataset increase after applying them?



Explain One-Hot Encoding and Label Encoding. Does the dimensionality of the dataset increase after applying them?

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

How do you use the F-test to select features?

F-Test is a statistical test used to compare models and check if the difference is significant between them. The hypothesis testing uses a model X and Y, where:

- X is a model created by just a constant.
- Y is the model created by a constant and a feature.

We calculate the least square errors in both models and compare and check if the difference in errors between model X and Y are significant or introduced by chance.

This significance returns the importance of each feature, so we select the features that return high F-values and use those for further analysis.

What are some recommended choices for Imputation Values?

For numeric features:

- If the data is normally distributed, use the mean value.
- If the data is skewed or has a lot of outliers, use the median value.

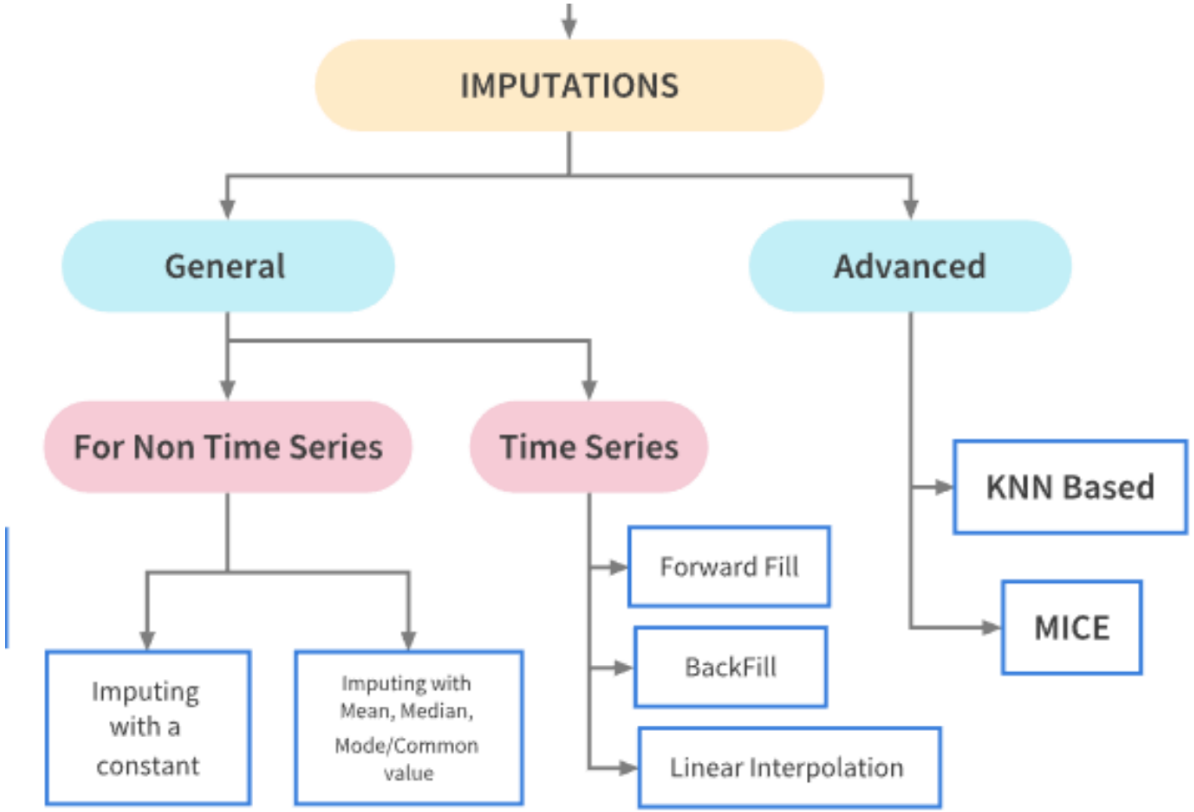
For categorical features:

- If the data is sortable, use the median value.
- If the data is not sortable, use the mode.

For boolean features:

- Use the frequency of the feature being true .

What are some recommended choices for Imputation Values?



Gradient Descent

What is the difference between Cost Function vs Gradient Descent?

- **A Cost Function** is something we want to minimize. For example, our cost function might be the sum of squared errors over the training set.
- **Gradient Descent** is a method for finding the minimum of a function of multiple variables.

What is the idea behind the Gradient Descent?

- A Gradient Descent is a type of optimization algorithm used to find the local minimum of a differentiable function.
- The main idea behind the gradient descent is to take steps in the negative direction of the gradient.
- This will lead to the steepest descent and eventually it will lead to the minimum point. It is shown as an equation by:

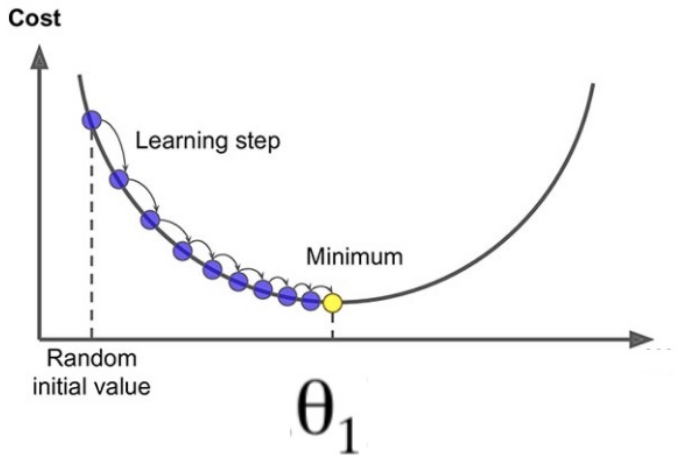
What is the idea behind the Gradient Descent?

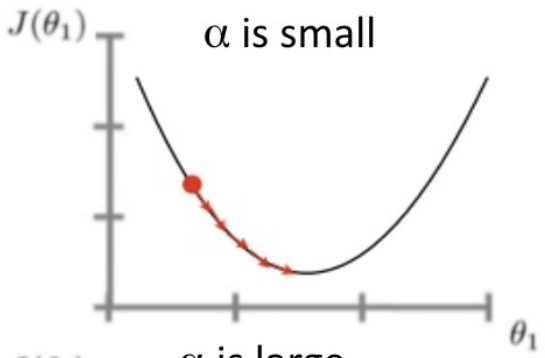
repeat until convergence {

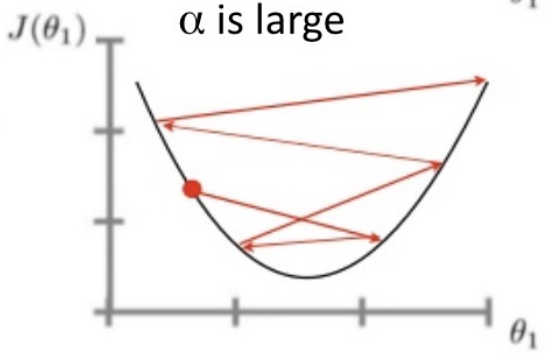
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j = 1$ and $j = 0$)

}







Explain the intuition behind Gradient Descent algorithm

- **Gradient descent** is an optimization algorithm that's used when training a machine learning model and is based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum (that is, slope = 0).
- For a start, we have to select a random bias and weights, and then iterate over the slope function to get a slope of 0.
- The way we change update the value of the bias and weights is through a variable called the learning rate. We have to be wise on the learning rate:
- A small leaning rate may lead to the model to take some time to learn
- A large learning rate will make the model converge as our pointer will shoot and we'll not be able to get to minima.

What is the difference between Maximum Likelihood Estimation and Gradient Descent?

Maximum likelihood estimation is a general approach to estimating parameters in a statistical model by maximizing the likelihood function defined as:

$$L(\theta|X) = f(X|\theta)$$

that is, the probability of obtaining data X given some value of parameter θ.

Knowing the likelihood function for a given problem you can look for such θ that maximizes the probability of obtaining the data you have.

What is the difference between Maximum Likelihood Estimation and Gradient Descent?

Sometimes we have known estimators, e.g. arithmetic mean is an MLE estimator for μ parameter for normal distribution, but in other cases, you can use different methods that include using optimization algorithms.

ML approach does not tell you how to find the optimal value of θ, it just tells you how you can compare if one value of θ is "more likely" than the other.

What is the difference between Maximum Likelihood Estimation and Gradient Descent?

- Gradient descent is an optimization algorithm. You can use this algorithm to find minimum (or maximum, then it is called gradient ascent) of many different functions.
- The algorithm does not care what is the function that it minimizes, it just does what it was asked for.
- So with using an optimization algorithm you have to know somehow how could you tell if one value of the parameter of interest is "better" than the other.
- You have to provide your algorithm some function to minimize and the algorithm will deal with finding its minimum.

What is the difference between Maximum Likelihood Estimation and Gradient Descent?

- You can obtain maximum likelihood estimates using different methods and using an optimization algorithm is one of them.
- On another hand, gradient descent can be also used to maximize functions other than the likelihood function.

Can Gradient Descent be applied to Non-Convex Functions?

- Gradient descent is a generic method for continuous optimization, so it can be, and is very commonly, applied to nonconvex functions.
- With a smooth function and a reasonably selected step size, it will generate a sequence of points $X_1, \dots, X_n$ with strictly decreasing values $f(x_1) > f(x_2), \dots$.
- Gradient descent will eventually converge to a stationary point of the function, regardless of convexity.
- If the function is convex, this will be a global minimum, but if not, it could be a local minimum or even a saddle point.

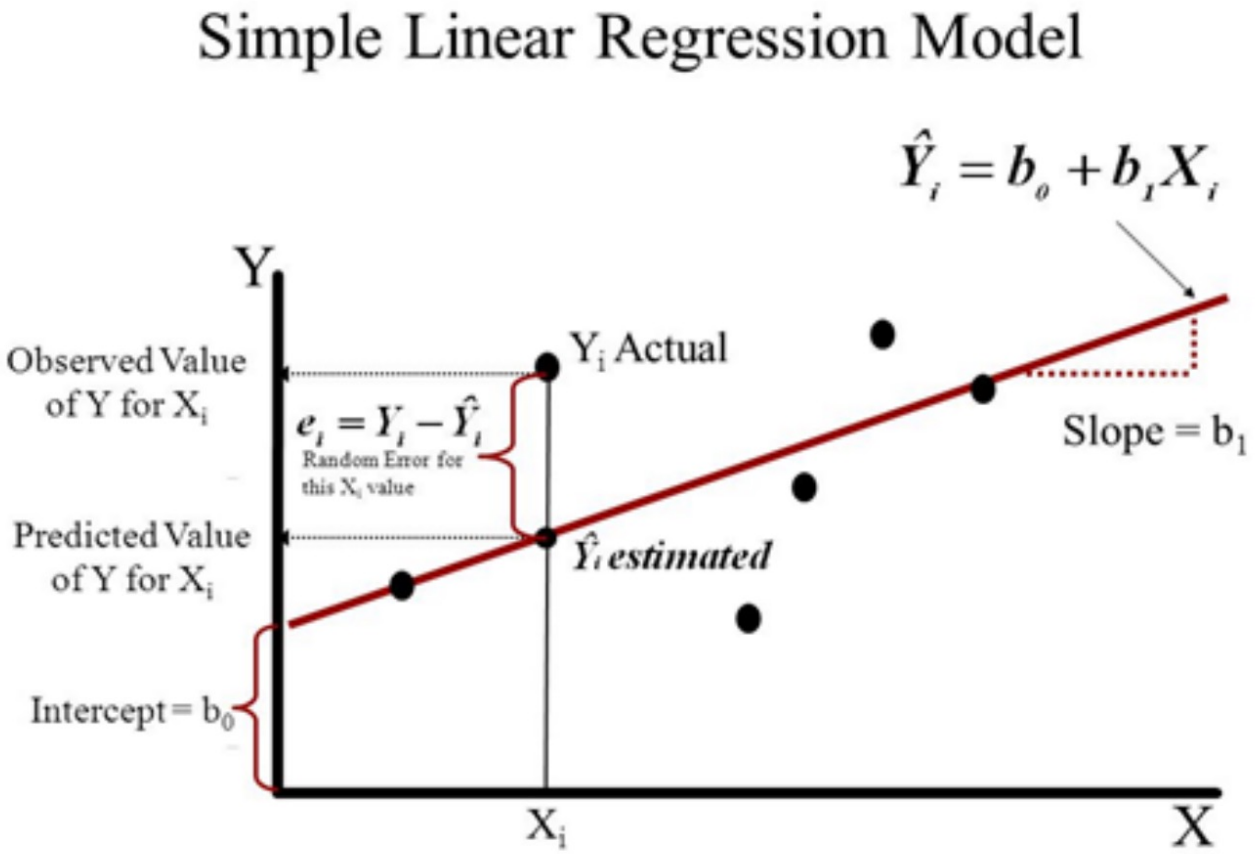
Linear Regression

What is Linear Regression?

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope.

It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

What is Linear Regression?



What are types of Linear Regression?

- Simple linear regression uses traditional slope-intercept form. x represents our input data and y represents our prediction.

$$y = mx + b$$

- A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

$$a(x) = w_0 + w_1 * x_1 + w_2 * x_2 + \ \ldots$$

What are types of Linear Regression?

- The variables $x1$, x2, x3 represent the attributes, or distinct pieces of information, we have about each observation.
- For sales predictions, these attributes might include a company's advertising spend on radio, TV, and newspapers.

$$a(x) = w_0 + w_1 * radio + w_2 * TV + w_3 * newspapers$$

How can you check if the Regression model fits the data well?

1. **R-squared:** It is a statistical measure of how close the data points are to the fitted regression line. Its value is always between 0 and 1 . The closer to 1 , the better the regression model fits the observations.
2. **F-test:** It evaluates the null hypothesis that the data is described by an intercept-only model, which is a regression with all the coefficients equal to zero versus the alternative hypothesis that at least one is not. If the P-value for the F-test is less than the significance level, we can reject the null hypothesis and conclude that the model provides a better fit than the intercept-only model.

How can you check if the Regression model fits the data well?

3. **Root Mean Square Error (RMSE):** It measures the average deviation of the estimates from the observed value. How good this value is must be assessed for each project and context.

For example, an RMSE of 1,000 for a house price prediction is probably good as houses tend to have prices over $100,000 , but an RMSE of 1,000 for a life expectancy prediction is probably terrible as the average life expectancy is around 78 .

What is the difference between MAE vs MSE?

- The Mean Squared Error measures the variance of the residuals and is used when we want to punish the outliers in the dataset. It's defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

What is the difference between MAE vs MSE?

- The Mean Absolute Error measures the average of the residuals in the dataset. Is used when we don't want outliers to play a big role.
- It can also be useful if we know that our distribution is multimodal, and it's desirable to have predictions at one of the modes, rather than at the mean of them.
- It's defined as:

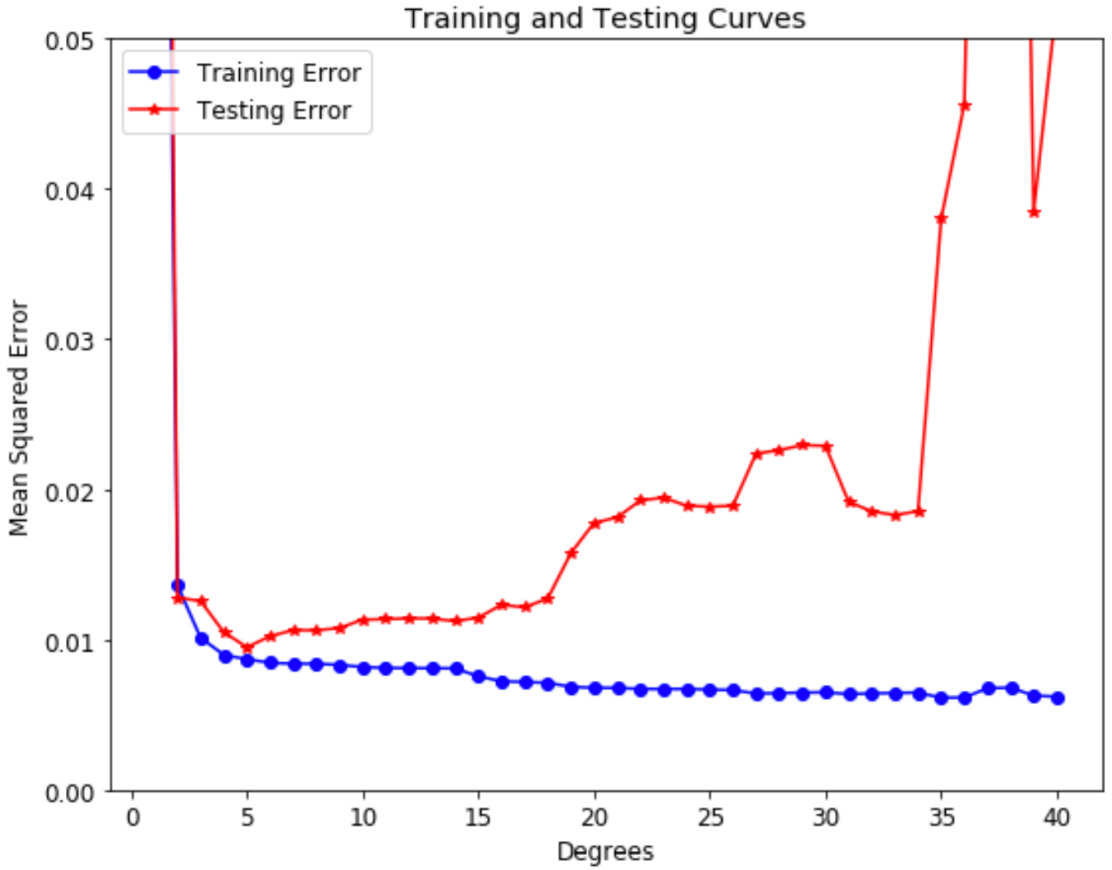
$$MAE = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}|$$

How would you detect Overfitting in Linear Models?

The common pattern for overfitting can be seen on learning curve plots, where model performance on the training dataset continues to improve (e.g. loss or error continues to fall) and performance on the test or validation set improves to a point and then begins to get worse.

So an overfit model will have extremely low training error but a high testing error.

How would you detect Overfitting in Linear Models?



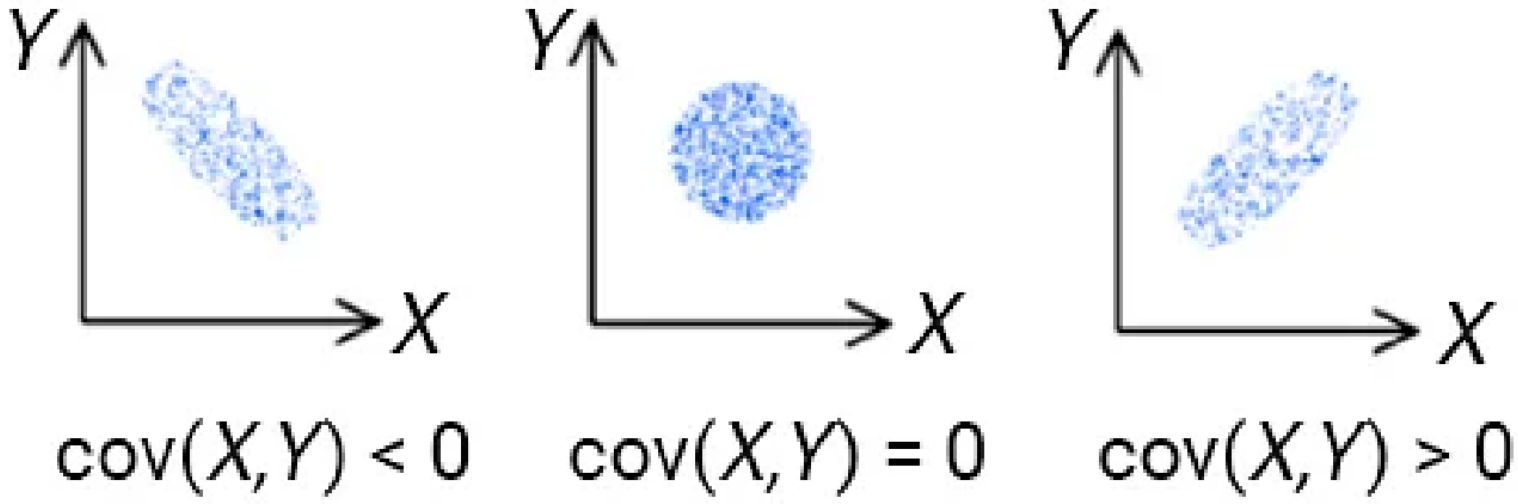
What's the difference between Covariance and Correlation?

- **Covariance** measures whether a variation in one variable results in a variation in another variable, and deals with the **linear relationship** of only 2 variables in the dataset.
- Its value can take range from $-\infty$ to $+\infty$.
- Simply speaking Covariance indicates the direction of the linear relationship between variables.

What's the difference between Covariance and Correlation?

- Correlation measures how strongly two or more variables are related to each other.
- Its values are between -1 to 1 .
- Correlation measures both the strength and direction of the linear relationship between two variables.
- Correlation is a function of the covariance.

What's the difference between Covariance and Correlation?



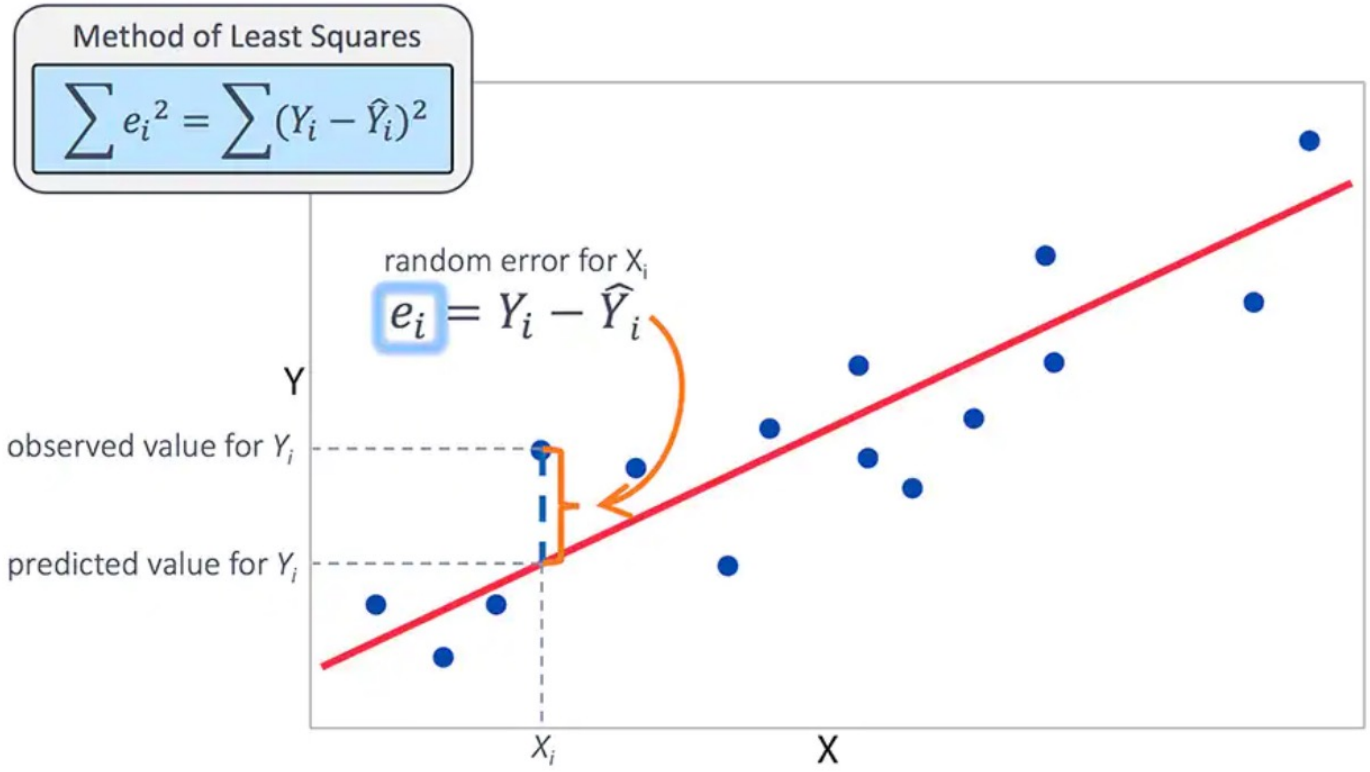
How is the Error calculated in a Linear Regression model?

1. Measuring the distance of the observed y-values from the predicted y-values at each value of x.
2. Squaring each of these distances.
3. Calculating the mean of each of the squared distances.

MSE = (1/n) * Σ(actual – forecast)^2

1. The smaller the Mean Squared Error, the closer you are to finding the line of best fit
2. How bad or good is this final value always depends on the context of the problem, but the main goal is that its value is so minimal as possible.

How is the Error calculated in a Linear Regression model?



Define Linear Regression and its structure

- Linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables.
- In a supervised linear regression, the model tries to find a linear relationship between the input and output data points.
- This linear relationship is a straight line if graphed.
- If there is only one explanatory variable it is called simple linear regression, and if there are more than one explanatory variable it is called multiple linear regression.

$$y = X * \beta + \epsilon$$

Define Linear Regression and its structure