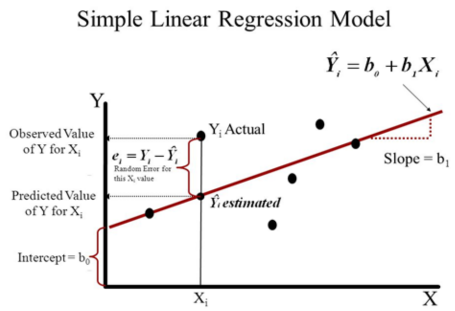
# Interview Questions

16 August 2023 15:54

**Q1.** What is linear regression ?

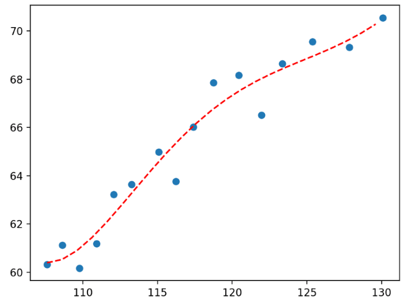
**Ans. Linear Regression** is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It’s used to predict values within a *continuous range*, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).


Simple Linear Regression Model
$\hat{Y}_i = b_o + b_1 X_i$
Y
Observed Value of Y for $X_i$
$Y_i$ Actual
$e_i = Y_i - \hat{Y}_i$
Random Error for this $X_i$ value
Slope = $b_1$
Predicted Value of Y for $X_i$
$\hat{Y}_i$ estimated
Intercept = $b_0$
$X_i$
X

**Q2**. How does a non-linear regression differ from linear regression analysis ?

**Ans**. ***Non-linear*** functions have variables with powers greater than 1. Like x^2. If these non-linear functions are graphed, they do not produce a straight line (their direction changes constantly).

- ***Linear*** functions have variables with only powers of 1. They form a straight line if it is graphed.
- ***Non-linear*** regression analysis tries to model a non-linear relationship between the independent and dependent variables.
- A simple non-linear relationship is shown below:



***Linear*** regression analysis tries to model a linear relationship between the independent and dependent variables.

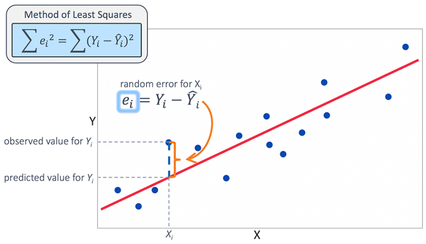
**Q3.** How the error calculated in a linear regression model ?

**Ans**. Measuring the distance of the observed *y-values* from the predicted *y-values* at each value of *x*.

1. Squaring each of these distances.
2. Calculating the *mean* of each of the squared distances.

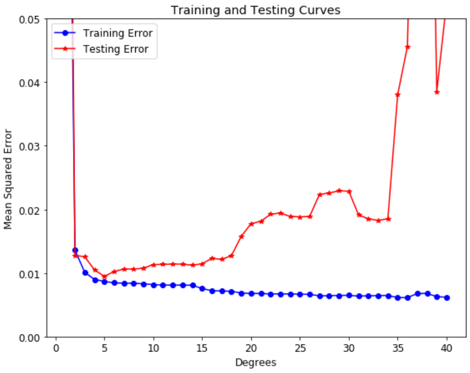
MSE = (1/n) * Σ(actual – predicted)^2

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.



**Q4.** How would you detect overfitting in linear models ?

**Ans.** 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**.

**Q5.** What is the difference between Mean Squared Error and Mean Absolute Error ?

**Ans.**

- 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:

$$MSE = (1 / N) \quad * \quad \sum (yi - y')\,^\wedge 2$$

- 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:

$$MAE = (1 / N) \quad * \quad \sum \ | \ yi - y' \,|$$

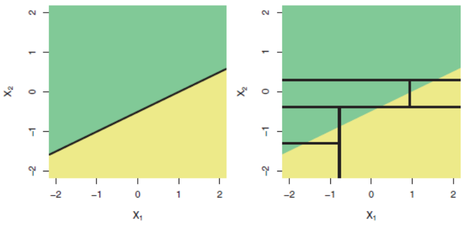
[Equation]

*Q6. Compare linear regression with Decision Trees ?*

[Equation]

*Ans.*

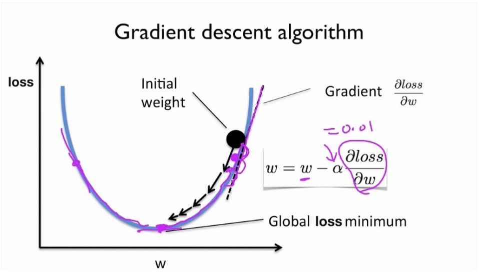
- **Linear regression** is used to predict *continuous* outputs where there is a linear relationship between the features of the dataset and the output variable.
- **Decision trees** work by splitting the dataset, in a tree-like structure, into smaller and smaller subsets and make predictions based on which subset the new example falls into.
- **Linear regression** is used for *regression* problems where it predicts something with infinite possible answers such as the price of a house.
- **Decision trees** can be used to predict both *regression* and *classification* problems.
- **Linear regression** is prone to *underfitting* the data. Switching to *polynomial regression* will sometimes help in countering underfitting.
- **Decision trees** are prone to *overfit* the data. *Pruning* helps with the overfitting problem.



**Q7.** Explain how gradient descent work in linear regression ?

**Ans.** The **Gradient Descent** works by starting with random values for each coefficient in the linear regression model.

- After this, the *sum of the squared errors* is calculated for each pair of input and output values (loss function), using a *learning rate* as a scale factor.
- For each iteration, the coefficients are updated in the direction towards *minimizing the error*,
- then we keep repeating the iteration process until a *minimum sum squared error* is achieved or no further improvement is possible.


Gradient descent algorithm
loss
Initial weight
Gradient $\frac{\partial loss}{\partial w}$
=0.01
$w = w - \alpha \frac{\partial loss}{\partial w}$
Global loss minimum
w

**Q8.** How would you decide the importance of variable for the multivariate linear regression model ?

**Ans.** A way to perform the **variable selection** is trying out different models, each containing a different subset of the predictors. For instance, if the number of predictors is 2, then we can consider 4 models:

1. A model containing no variables.
2. A model containing X1 only.
3. A model containing X2 only.
4. A model containing both X1 and X2.

We can then select the best model out of all of the models that we have considered by computing some statistics like **Adjusted R-squared**. However, if the number of predictors is high, we must use some more elaborated methods for feature selection, like:

- **stepwise regression**,
- **forward selection**, and
- **backward elimination**.

**Q9.** Name a disadvantage of R-Squared and how would you address it ?

**Ans. R-squared ($R^2$)** is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

**R-squared** takes values between 0 and 1, with 0 indicating that the proposed model does not improve prediction over the mean model and 1 indicating the perfect prediction. However, one **drawback** of R-squared is that its values can increase if we add predictors to the regression model, leading to a possible *overfitting*.

To address this issue, we can use **Adjusted R-squared**: a modified version of *R-squared* that has been adjusted for the number of predictors in the model. The adjusted R-squared increases when the new term improves the model more than would be expected by chance, and it decreases when a predictor improves the model by less than expected.

**Q10.** What are the assumptions of linear regression ?

**Ans.** We make a few assumptions when we use linear regression to model the relationship between a response and a predictor. These assumptions are essential conditions that should be met before we draw inferences regarding the model estimates or before we use a model to make a prediction.

There are **four principal assumptions** which justify the use of linear regression models for purposes of inference or prediction:

**(i) Linearity and Additivity** of the relationship between dependent and independent variables:

- (a) The expected value of dependent variable is a straight-line function of each independent variable, holding the others fixed.
- (b) The slope of that line does not depend on the values of the other variables.
- (c) The effects of different independent variables on the expected value of the dependent variable are additive.

**(ii) Statistical Independence** of the errors (in particular, no correlation between consecutive errors in the case of time series data)

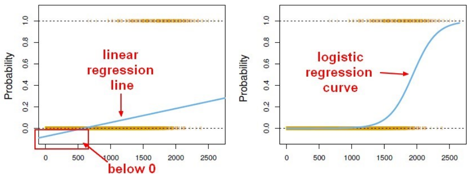
**(iii) Homoscedasticity** (constant variance) of the errors

- (a) versus time (in the case of time series data)
- (b) versus the predictions
- (c) versus any independent variable

**(iv) Normality** of the error distribution.

**Q11.** What is the difference between Linear Regression and Logistic Regression ?

**Ans. Linear regression output as probabilities** In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In **logistic regression**, the outcome (dependent variable) has only a limited number of possible values.



- **Outcome** In linear regression, the outcome (dependent variable) is continuous. It can have any one of an infinite number of possible values. In logistic regression, the outcome (dependent variable) has only a limited number of possible values.
- **The dependent variable** Logistic regression is used when the response variable is categorical in nature. For instance, yes/no, true/false, red/green /blue, 1st/2nd/3rd/4th, etc. Linear regression is used when your response variable is continuous. For instance, weight, height, number of hours, etc.
- **Equation** Linear regression gives an equation which is of the form Y = mX + C, means equation with degree 1. However, logistic regression gives an equation which is of the form Y = eX + e-X
- **Coefficient interpretation** In linear regression, the coefficient interpretation of independent variables are quite straightforward (i.e. holding all other variables constant, with a unit increase in this variable, the dependent variable is expected to increase/decrease by xxx). However, in logistic regression, depends on the family (binomial, Poisson, etc.) and link (log, logit, inverse-log, etc.) you use, the interpretation is different.
- **Error minimization technique** Linear regression uses *ordinary least squares* method to minimise the errors and arrive at a best possible fit, while logistic regression uses *maximum likelihood* method to arrive at the solution. Linear regression is usually solved by minimizing the least squares error of the model to the data, therefore large errors are penalized quadratically. Logistic regression is just the opposite. Using the logistic loss function causes large errors to be penalized to an asymptotically constant. Consider linear regression on categorical {0, 1} outcomes to see why this is a problem. If your model

predicts the outcome is 38, when the truth is 1, you've lost nothing. Linear regression would try to reduce that 38, logistic wouldn't (as much)2.

**Q12.** What is the difference between Ordinary Least Squares and Lasso Regression ?

**Ans.**

- **Ordinary least squares** fit a linear model by minimizing the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear approximation. Mathematically it solves a problem of the form:

[Equation]

$$\min_w ||Xw - y||_2^2$$

- The **Lasso** regression fits a linear model that estimates **sparse coefficients**. It is useful in some contexts due to its tendency to prefer solutions with fewer non-zero coefficients, effectively reducing the number of features upon which the given solution is dependent. Mathematically, it consists of a linear model with an added regularization term:

$$\min_w \frac{1}{2n} ||Xw - y||_2^2 + \alpha ||w||_1$$

**Q13.** Why use Root Mean Squared Error (RMSE) instead of Mean Absolute Error (MAE) ?

**Ans.** This depends on your **loss function**. In many circumstances it makes sense to give more weight to points further away from the mean:

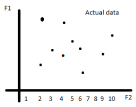
- If being off by 10 is more than *twice* as bad as being off by 5. In such cases, **RMSE** is a more appropriate measure of error.
- If being off by 10 is just twice as bad as being off by 5, then **MAE** is more appropriate.

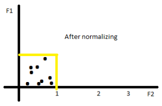
Another situation when you want to use (R)MSE instead of MAE: when your observations' conditional distribution is asymmetric and you want an unbiased fit. The (R)MSE is minimized by the conditional *mean*, the MAE by the conditional *median*. So if you minimize the MAE, the fit will be closer to the median and biased.

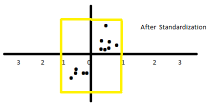
**Q14.** Why would you use Normalisation vs Standardisation for linear regression ?

**Ans.**

- **Normalization** transforms your data into a range between 0 and 1
- **Standardization** transforms your data such that the resulting distribution has a mean of 0 and a standard deviation of 1







**Q15.** Explain the stepwise regression technique ?

**Ans. Stepwise Regression** is a **feature selection** technique which objective is to reduce the number of features and, hence, reduce the computational complexity of the model. This technique is based on select models with the lowest **p-values**.

For illustrated this technique let's suppose we got 6 predictors in the dataset, so in order to perform stepwise regression we must follow the next steps:

1. We fit the model with **one predictor** and the target variable. We tried each predictor one by one for then compute its p-value. Let's say that among all predictors the model with the lowest p-value was the one that contains only X1, so we keep this model.
2. Now will fit the model with **two predictors**. One we have already selected in step 1 and for the second predictor, we will try one by one with all remaining predictors. In other words, we fit one model using X1 and X2, another model using X1 and X3, and so on. For each case we compute the p-value and once again we select the model with the lowest p-value.
3. Now will try to fit the model with **three predictors**. We take the predictors already selected in step 2 and the third could it be any of the remaining ones. But if in this process we found that for each possible model we no longer reach a p-value less than 0.05 we **stop** this process. A p-value greater than 0.05 means that the model is not significant so we can reject it.

By following the previous steps we can get the **smallest set** of features that have a significant impact on the final model fit, and at the same time, reduce computational cost and avoid overfitting.

**Q16.** How would you deal with overfitting in linear regression models ?

**Ans. Overfitting** is a synonym of a *complex model*, so we can solve this problem by trying to reduce its complexity. For this purpose, we can perform some **regularization techniques**, these techniques add a *penalty term* to the best fit derived from the trained data, in order to achieve a *lesser variance* with the tested data. It also restricts the influence of predictor variables over the output variable by compressing their coefficients and then reducing the complexity of the model. The regularization techniques are:

1. **Ridge Regression**: It works by adding bias to a multilinear regression model. The penalty term is known as ***L1*** and is defined as λ(m)² where m is the slope of the line.

$$y = \beta_0 + \beta_1 x_1 + \cdots \beta_k x_k + \lambda(m)^2$$

The penalty term restricts the coefficients of predictor variables but **never makes them zero**. In this way, we will have a better accurate regression with tested data at a cost of losing accuracy for the training data.

4. **Lasso Regression**: It works in a similar way that ridge regression but only differs in the penalty term ***L2***, which is equal to λ|m|. This regression is defined below as:

$$y = \beta_0 + \beta_1 x_1 + \cdots \beta_k x_k + \lambda|m|$$

In this case, the penalty term can remove the variables by making their **coefficients to zero** thus removing the variables that have *high covariance* with other predictor variables.

1. **ElasticNet regression**: This is a fancier combination of both Ridge and Lasso. A hyperparameter α with values between 0 and 1, is provided to assign how much weight is given to each of the ***L1*** and ***L2*** penalties:

$$y = \beta_0 + \beta_1 x_1 + \cdots \beta_k x_k + \alpha\lambda(m)^2 + (1-\alpha)\lambda|m|$$

The parameter α determines the mix of the penalties and is often pre-chosen on qualitative grounds. This technique can result in better performance than a model with either one or the other penalty depending on the problem and the complexity of the model.