MACHINE LEARNING

Q1) C

Q2) C

Q3) C

Q4) A

Q5) C

Q6) A & D

Q7) A & B

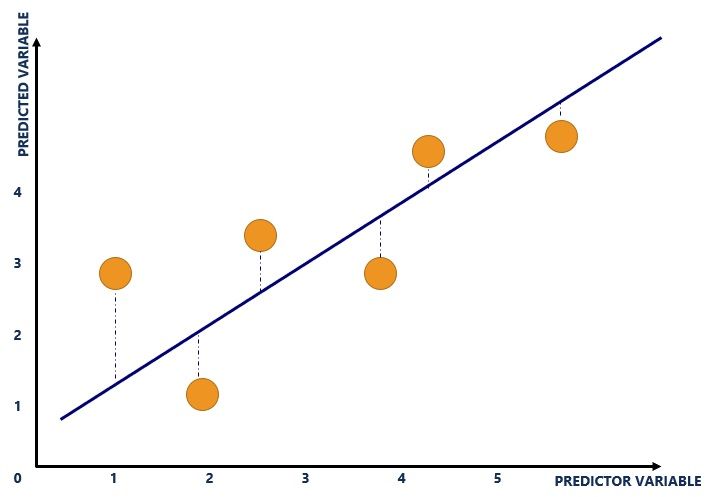
Q8) A &B

Q9) A & C

**Q10)**

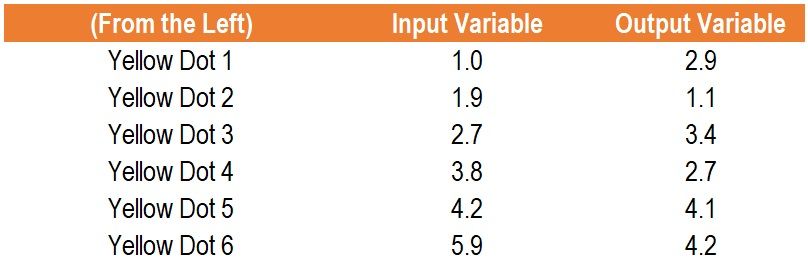
**What is the R-squared?**

The R-squared, also called the[coefficient of determination](https://corporatefinanceinstitute.com/resources/knowledge/other/coefficient-of-determination/), is used to explain the degree to which input variables (predictor variables) explain the variation of output variables (predicted variables). It ranges from 0 to 1. For example, if the R-squared is 0.9, it indicates that 90% of the variation in the output variables are explained by the input variables. Generally speaking, a higher R-squared indicates a better fit for the model. Consider the following diagram:



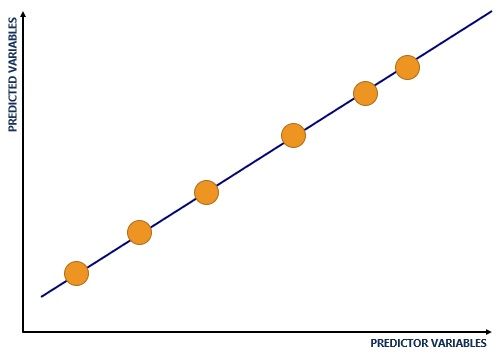
The blue line refers to the line of best fit and shows the relationship between variables. The line is calculated through [regression analysis](https://corporatefinanceinstitute.com/resources/knowledge/finance/regression-analysis/) and is plotted where the vertical distances (blue dotted lines) of the yellow dots to the line of best fit is minimized.

The yellow dots refer to the plot of input and output variables. The input variable is plotted on the x-axis while the output variable is plotted on the y-axis. For example, the graph above consists of the following dataset:



The blue dotted lines refer to the distance of the plot of input and output

. The R-squared is derived from the distance of all the yellow dots from the line of best fit (the blue line). For example, the following diagram would illustrate an R-squared of 1:

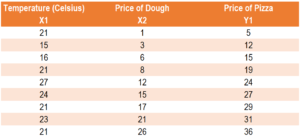


**Problems with the R-squared**

R-squared comes with an inherent problem – additional input variables will make the R-squared stay the same or increase (this is due to how the R-squared is calculated mathematically). Therefore, even if the additional input variables show no relationship with the output variables, the R-squared will increase. An example that explains such an occurrence is provided below.

**Understanding the Adjusted R-squared**

Essentially, the adjusted R-squared looks at whether additional input variables are contributing to the model. Consider an example using data collected by a pizza owner, as shown below:



Assume the pizza owner runs two regressions:

**Regression 1: Price of Dough (input variable), Price of Pizza (output variable)**

Regression 1 yields an R-squared of 0.9557 and an adjusted R-squared of 0.9493.

**Regression 2: Temperature (input variable 1), Price of Dough (input variable 2), Price of Pizza (output variable)**

Regression 2 yields an R-squared of 0.9573 and an adjusted R-squared of 0.9431.

Although temperature should not exert any predictive power on the price of a pizza, the R-squared increased from 0.9557 (Regression 1) to 0.9573 (Regression 2). A person may believe that Regression 2 carries higher predictive power since the R-squared is higher. Even though the input variable of temperature is useless in predicting the price of a pizza, it increased the R-squared. Here, the adjusted R-squared comes in.

The adjusted R-squared looks at whether additional input variables are contributing to the model. The adjusted R-squared in Regression 1 was 0.9493 compared to the adjusted R-squared in Regression 2 of 0.9493. Therefore, the adjusted R-squared is able to identify that the input variable of temperature is not helpful in explaining the output variable (the price of a pizza). In such a case, the adjusted R-squared would point the model creator to using Regression 1 rather than Regression 2.

**Example of the Adjusted R-squared**

Consider two models:

* Model 1 uses input variables X1, X2, and X3 to predict Y1.
* Model 2 uses input variables X1 and X2 to predict Y1.

Which model should be used? Information regarding both models are provided below:



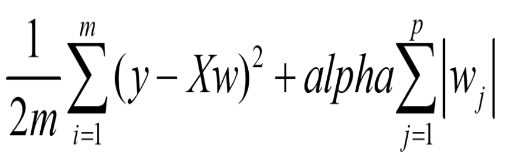
Comparing the R-squared between Model 1 and Model 2, the R-squared predicts that Model 1 is a better model as it carries greater explanatory power (0.5923 in Model 1 vs. 0.5612 in Model 2).

Comparing the R-squared between Model 1 and Model 2, the adjusted R-squared predicts that the input variable X3 contributes to explaining output variable Y1 (0.4231 in Model 1 vs. 0.3512 in Model 2).

As such, Model 1 should be used, as the additional X3 input variable contributes to explaining the output variable Y1.

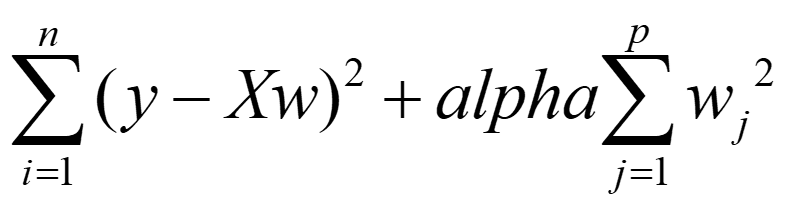
**Q11)**

**Lasso**is a modification of linear regression, where the model is penalized for the sum of absolute values of the weights. Thus, the absolute values of weight will be (in general) reduced, and many will tend to be zeros. During training, the objective function become:



As you see, Lasso introduced a new hyperparameter, *alpha*, the coefficient to penalize weights.

**Ridge** takes a step further and penalizes the model for the sum of squared value of the weights. Thus, the weights not only tend to have smaller absolute values, but also really tend to penalize the extremes of the weights, resulting in a group of weights that are more evenly distributed. The objective function becomes:



**Q12)**

The Variance Inflation Factor (VIF) measures the severity of multicollinearity in [regression analysis](https://corporatefinanceinstitute.com/resources/knowledge/finance/regression-analysis/). It is a statistical concept that indicates the increase in the variance of a regression coefficient as a result of collinearity.

**Q13)**

**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. Having features on a similar scale will help the gradient descent converge more quickly towards the minima.

**Q14)**

# Evaluation metrics for a linear regression model

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Let us look at**MSE, MAE, R-squared, Adjusted R-squared, and RMSE.**

## Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

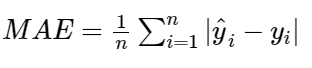


Mean squared error. Image by the author.

MSE penalizes large errors.

## Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.



MAE does not penalize large errors.

## **R-squared or Coefficient of Determination**

This metric represents the part of the variance of the dependent variable explained by the independent variables of the model. It measures the strength of the relationship between your model and the dependent variable.

To understand what R-square really represents let us consider the following case where we measure the error of the model with and without the knowledge of the independent variables.

**Calculating regression error**When we know the values of the independent variables, we can calculate the regression error.

We know that residual is the difference between actual and predicted value. Thus, RSS (Residual sum of squares) can be calculated as follows.



Residual sum of squares. Image by the author.

**Calculating squared residual error**Consider the case where we don't know the values of the independent variables. We only have the **y** values. With this, we calculate the **mean of the y values.** This point can be represented as a horizontal line. Now we calculate the sum of squared error between the **mean y value** and that of every other **y** value.

The total variation in Y can be given as a sum of squared differences of the distance between every point and the arithmetic mean of Y values. This can be termed as **TSS**(Total sum of squares).



**SQL**

Q1) A& C

Q2) A, C & D

Q3) B

Q4) C

Q5) B

Q6) B

Q7) A

Q8) C

Q9) D

**Q10) A**

**Q11) Denormalization**

When we normalize tables, we break them into multiple smaller tables. So when we want to retrieve data from multiple tables, we need to perform some kind of join operation on them. In that case, we use the denormalization technique that eliminates the drawback of normalization.

Denormalization is a technique used by database administrators to optimize the efficiency of their database infrastructure. This method allows us to add redundant data into a normalized database to alleviate issues with database queries that merge data from several tables into a single table. The denormalization concept is based on the definition of normalization that is defined as arranging a database into tables correctly for a particular purpose.

**Pros of Denormalization**

The following are the advantages of denormalization:

**1. Enhance Query Performance**

Fetching queries in a normalized database generally requires joining a large number of tables, but we already know that the more joins, the slower the query. To overcome this, we can add redundancy to a database by copying values between parent and child tables, minimizing the number of joins needed for a query.

**2. Make database more convenient to manage**

A normalized database is not required calculated values for applications. Calculating these values on-the-fly will take a longer time, slowing down the execution of the query. Thus, in denormalization, fetching queries can be simpler because we need to look at fewer tables.

**3. Facilitate and accelerate reporting**

Suppose you need certain statistics very frequently. It requires a long time to create them from live data and slows down the entire system. Suppose you want to monitor client revenues over a certain year for any or all clients. Generating such reports from live data will require "searching" throughout the entire database, significantly slowing it down.

**Cons of Denormalization**

The following are the disadvantages of denormalization:

* It takes large storage due to data redundancy.
* It makes it expensive to updates and inserts data in a table.
* It makes update and inserts code harder to write.
* Since data can be modified in several ways, it makes data inconsistent. Hence, we'll need to update every piece of duplicate data. It's also used to measure values and produce reports. We can do this by using triggers, transactions, and/or procedures for all operations that must be performed together.

**Q12)**

A cursor in SQL Server is a d**atabase object that allows us to retrieve each row at a time and manipulate its data**. A cursor is nothing more than a pointer to a row. It's always used in conjunction with a SELECT statement. It is usually a collection of [SQL](https://www.javatpoint.com/sql-tutorial) logic that loops through a predetermined number of rows one by one. A simple illustration of the cursor is when we have an extensive database of worker's records and want to calculate each worker's salary after deducting taxes and leaves.

The [SQL Server](https://www.javatpoint.com/sql-server-tutorial) **cursor's purpose is to update the data row by row, change it, or perform calculations that are not possible when we retrieve all records at once**. It's also useful for performing administrative tasks like SQL Server database backups in sequential order. Cursors are mainly used in the development, DBA, and ETL processes.

**Types of Cursors in SQL Server**

The following are the different types of cursors in SQL Server listed below:

* Static Cursors
* Dynamic Cursors
* Forward-Only Cursors
* Keyset Cursors

**Q13)**

### ****Types of SQL Queries****

#### **Select Query**

The select query is the least difficult kind of inquiry and thus, it is likewise the most ordinarily utilized one in Microsoft Access databases. It very well may be utilized to choose and show information from possibly one table or a progression of them relying upon what is required.

At last, it is the client decided criteria that tell the database what the determination is to be founded on. After the select query is called, it makes a "virtual" table where the information can be changed, however at close to one record at any given moment.

#### **Action Query**

At the point when the activity question is called, the database experiences a particular activity relying upon what was indicated in the query itself. This can incorporate such things as making new tables, erasing lines from existing ones and refreshing records or making totally new ones.

Action queries are extremely famous in information the board since they take into account numerous records to be changed at one time rather than just single records like in a select query.

Four types of action queries are:

1. **Append Query** – takes the set consequences of a query and "adds" (or includes) them to a current table.
2. **Delete Query** – erases all records in a hidden table from the set results of a query.
3. **Make Table Query** – as the name proposes, it makes a table dependent on the set consequences of a query
4. **Update Query** – takes into account at least one field in your table to be refreshed.

#### **Parameter Query**

In Microsoft Access, a parameter query works with different sorts of queries to get whatever outcomes you are after. This is on the grounds that, when utilizing this kind of query, you can pass a parameter to an alternate query, for example, an activity or a select query. It can either be esteem or a condition and will basically tell the other query explicitly what you need it to do.

Usually picked in light of the fact that it takes into account an exchange box where the end client can enter whatever parameter value, they wish each time the query is being run. The parameter query is only an altered select query.

#### **Aggregate Query**

A unique kind of query is known as an aggregate query. It can chip away at different queries, (for example, choice, activity or parameter) simply like the parameter query does, yet as opposed to passing a parameter to another query it aggregates up to the things by chosen by the various groups.

It basically makes a summation of any chosen property in your table. This can be additionally created into measurable sums, for example, midpoints and standard deviation, just to name a couple.

**Q14)**

**Constraints in SQL**

Constraints in SQL means we are applying certain conditions or restrictions on the database. This further means that before inserting data into the database, we are checking for some conditions. If the condition we have applied to the database holds true for the data which is to be inserted, then only the data will be inserted into the database tables.

Constraints in SQL can be categorized into two types:

1. **ColumnLevelConstraint:**  
   Column Level Constraint is used to apply a constraint on a single column.
2. **TableLevelConstraint:**  
   Table Level Constraint is used to apply a constraint on multiple columns.

Some of the real-life examples of constraints are as follows:

1. Every person has a unique email id. This is because while creating an email account for any user, the email providing services such as Gmail, Yahoo or any other email providing service will always check for the availability of the email id that the user wants for himself. If some other user already takes the email id that the user wants, then that id cannot be assigned to another user. This simply means that no two users can have the same email ids on the same email providing service. So, here the email id is the constraint on the database of email providing services.
2. Whenever we set a password for any system, there are certain constraints that are to be followed. These constraints may include the following:
   * There must be one uppercase character in the password.
   * Password must be of at least eight characters in length.
   * Password must contain at least one special symbol.

Constraints available in SQL are:

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK
6. DEFAULT
7. CREATE INDEX

Now let us try to understand the different constraints available in SQL in more detail with the help of examples. We will use MySQL database for writing all the queries.

**Q15)**

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table. Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

**STATISTICS**

Q1) D

Q2) A

Q3) A

Q4) C

Q5) C

Q6) B

Q7) C

Q8) B

Q9) B

**Q10)**

Well Box plot and Histogram can be used use Seaborn . where Box plot is Used using .boxplot and Histogram is used using .countplot.

Box plot - gives the quartiles and indicate the median data to compare easily

Histogram - gives only the count

**Q11)** Here are the key factors to consider as you work to improve performance**.**

1. Define your primary objective. ...
2. Choose your metric(s) - determine cause and effect. ...
3. Create relevant activities. ...
4. Evaluate periodically.

**Q12)** Statistical significance is a measure of reliability in the result of an analysis that allows you to be confident in your decision making.

Statistical significance is often calculated with statistical hypothesis testing, which tests the validity of a hypothesis by figuring out the probability that your results have happened by chance.

Here, a “hypothesis” is an assumption or belief about the relationship between your datasets. The result of a hypothesis test allows us to see whether this assumption holds under scrutiny or not.

A standard hypothesis test relies on two hypotheses.

* **Null hypothesis:** The default assumption of a statistical test that you’re attempting to disprove (e.g., an increase in cost won’t affect the number of purchases).
* **Alternative hypothesis:** An alternate theory that contradicts your null hypothesis (e.g., an increase in cost will reduce the number of purchases). This is the hypothesis you hope to prove.

The testing part of hypothesis tests allows us to determine which theory, the null or alternative, is better supported by data. There are many hypothesis testing methodologies, and one of the most common ones is the Z-test, which is what we’ll discuss here.  
  
But, before we get to the Z-test, it is important for us to visit some other statistical concepts the Z-test relies on.

**Q13)**

Exponential distributions do not have a log-normal distribution or a Gaussian distribution. In fact, any type of data that is categorical will not have these distributions as well. Example: Duration of a phone car, time until the next earthquake, etc.

**Q14)**

Income is the classic example of when to use the median instead of the mean because its distribution tends to be skewed. The median indicates that half of all incomes fall below 27581, and half are above it. For these data, the mean overestimates where most household incomes fall.

**Q15)**

The likelihood is the probability that a particular outcome is observed when the true value of the parameter is , equivalent to the probability mass on ; it is not a probability density over the parameter . The likelihood, should not be confused with , which is the posterior probability of given the data .