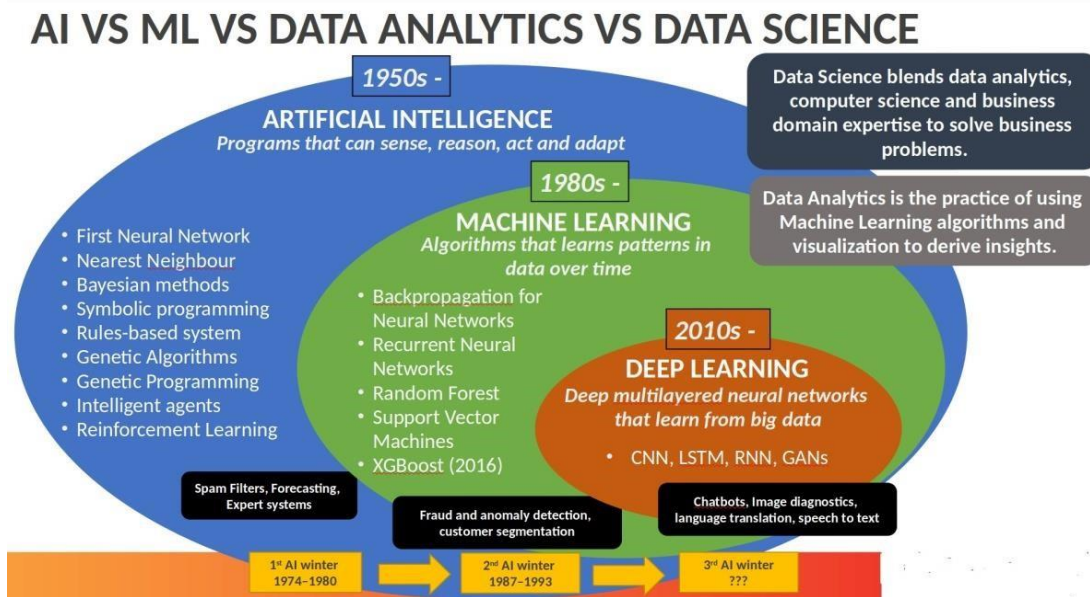# Day 1:
Data Science Interview Questions

Q1. What is the difference between AI, Data Science, ML, and DL?

Ans 1 :

Artificial Intelligence: AI is purely math and scientific exercise, but when it became computational, it started to solve human problems formalized into a subset of computer science. Artificial intelligence has changed the original computational statistics paradigm to the modern idea that machines could mimic actual human capabilities, such as decision making and performing more "human" tasks. Modern AI into two categories

1. General AI - Planning, decision making, identifying objects, recognizing sounds, social & business transactions.
2. Applied AI - driverless / Autonomous car or machine smartly trade stocks.

Machine Learning: Instead of engineers "teaching" or programming computers to have what they need to carry out tasks, that perhaps computers could teach themselves – learn something without being explicitly programmed to do so. ML is a form of AI were based on more data, and they can change actions and response, which will make more efficient, adaptable and scalable. e.g., navigation apps and recommendation engines. Classified into: -

1. Supervised
2. Unsupervised
3. Reinforcement learning

Data Science: Data science has many tools, techniques, and algorithms called from these fields, plus others –to handle big data.

The goal of data science, somewhat like machine learning, is to make accurate predictions and to automate and perform transactions in real-time, such as purchasing internet traffic or automatically generating content.

Data science relies less on math and coding and more on data and building new systems to process the data. Relying on the fields of data integration, distributed architecture, automated machine learning, data visualization, data engineering, and automated data-driven decisions, data science can cover an entire spectrum of data processing, not only the algorithms or statistics related to data.

Deep Learning: It is a technique for implementing ML.

ML provides the desired output from a given input, but DL reads the input and applies it to another data. In ML, we can easily classify the flower based upon the features. Suppose you want a machine to look at an image and determine what it represents to the human eye, whether a face, flower, landscape, truck, building, etc.

Machine learning is not sufficient for this task because machine learning can only produce an output from a data set – whether according to a known algorithm or based on the inherent structure of the data. You might be able to use machine learning to determine whether an image was of an "X" – a flower, say – and it would learn and get more accurate. But that output is binary (yes/no) and is dependent on the algorithm, not the data. In the image recognition case, the outcome is not binary and not dependent on the algorithm.

The neural network performs MICRO calculations with computational on many layers. Neural networks also support weighting data for 'confidence. These results in a probabilistic system, vs. deterministic, and can handle tasks that we think of as requiring more 'human-like' judgment.

Q2. What is the difference between Supervised learning, Unsupervised learning and Reinforcement learning?

Ans 2:

Machine Learning

Machine learning is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead.

Building a model by learning the patterns of historical data with some relationship between data to make a data-driven prediction.

Types of Machine Learning

• Supervised Learning

• Unsupervised Learning

• Reinforcement Learning

Supervised learning

In a supervised learning model, the algorithm learns on a labeled dataset, to generate reasonable predictions for the response to new data. (Forecasting outcome of new data)

- Regression
- Classification

Unsupervised learning

An unsupervised model, in contrast, provides unlabelled data that the algorithm tries to make sense of by extracting features, co-occurrence and underlying patterns on its own. We use unsupervised learning for

- Clustering
- Anomaly detection
- Association
- Autoencoders

Reinforcement Learning

Reinforcement learning is less supervised and depends on the learning agent in determining the output solutions by arriving at different possible ways to achieve the best possible solution.

Q3. Describe the general architecture of Machine learning.
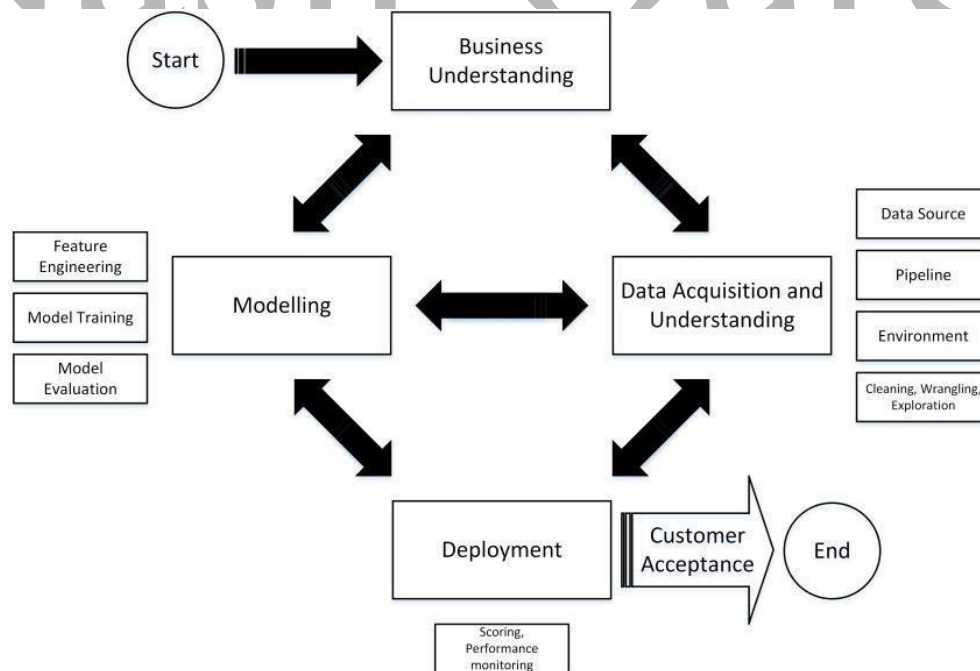
Explanation of the various steps:

Business understanding: Understand the give use case, and, it's good to know more about the domain for which the use cases are built.

Data Acquisition and Understanding: Data gathering from different sources and understanding the data. Cleaning the data, handling the missing data if any, data wrangling, and EDA (Exploratory data analysis).

Modeling:

Feature Engineering - scaling the data, feature selection - not all features are important. We use the backward elimination method, correlation factors, PCA and domain knowledge to select the features.

Model Training based on trial-and-error method or by experience, we select the algorithm and train with the selected features.

Model evaluation Accuracy of the model, confusion matrix and cross-validation.

If accuracy is not high, to achieve higher accuracy, we tune the model...either by changing the algorithm used or by feature selection or by gathering more data, etc.

Deployment - Once the model has good accuracy, we deploy the model either in the cloud or Raspberry py or any other place. Once we deploy, we monitor the performance of the model. If it found to be good...we go live with the model or reiterate the all process until our model performance is good.

It's not done yet!!!

What if , after a few days, our model performs badly because of new data. In that case, we do all the process again by collecting new data and redeploy the model.

Q4. What is Linear Regression? Ans 4:

Linear Regression explains a relationship between a dependent variable(Y) and one or more independent variable(X) by finding the best fit of the straight line.

The equation for the Linear model is $Y = mx + c$, where m is the slope and c is the intercept value.
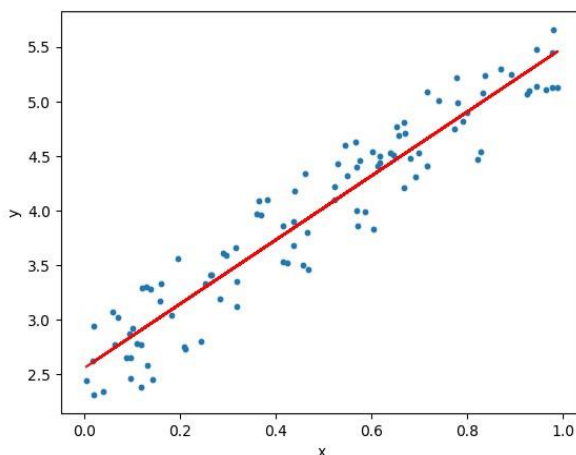
In this diagram,

The blue dots we see are the distribution of 'y' w.r.t 'x.'

There is no straight line that runs through all the data points.

So, the objective here is to fit the best fit of a straight line that will try to minimize the error between the expected and actual.

Q5. OLS Stats Model (Ordinary Least Square)

Ans 5:

OLS is a stats model, which will help us in identifying the more significant features that can has an influence on the output. OLS model in python is executed as:

lm = smf.ols(formula = 'Sales ~ am+constant', data = data).fit()

lm.conf_int()

lm.summary()

And we get the output as below,

```
                         OLS Regression Results
==============================================================================
Dep. Variable:                   mpg   R-squared:                       0.360
Model:                           OLS   Adj. R-squared:                  0.338
Method:                Least Squares   F-statistic:                     16.86
Date:               Wed, 17 Jan 2018   Prob (F-statistic):           0.000285
Time:                       14:07:51   Log-Likelihood:                -95.242
No. Observations:                 32   AIC:                             194.5
Df Residuals:                     30   BIC:                             197.4
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
constant       17.1474      1.125     15.247      0.000      14.851      19.444
am              7.2449      1.764      4.106      0.000       3.642      10.848
==============================================================================
Omnibus:                       0.480   Durbin-Watson:                   1.065
Prob(Omnibus):                 0.787   Jarque-Bera (JB):                0.589
Skew:                          0.051   Prob(JB):                        0.745
Kurtosis:                      2.343   Cond. No.                         2.46
==============================================================================
```

The higher the t-value for the feature, the more significant the feature is to the output variable. And also, the p-value plays a rule in rejecting the Null hypothesis (Null hypothesis stating the features has zero significance on the target variable.).

Note:

If the p-value is less than 0.05(95% confidence interval) for a feature, then we can consider the feature to be significant.

Q6. What is R-square (where to use and where not)?

Ans 8. R-squared 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.

The definition of R-squared is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%.

0% indicates that the model explains none of the variability of the response data around its mean.

100% indicates that the model explains all the variability of the response data around its mean.

<u>In general, the higher the R-squared, the better the model fits your data.</u>

Sum Squared Regression Error

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Sum Squared Total Error

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

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

where

R^2 = sample R-square
p = Number of predictors
N = Total sample size.

There is a problem with the R-Square. The problem arises when we ask this question to ourselves. ** Is it good to help as many independent variables as possible? **

The answer is No.

Because we understood that each independent variable should have a meaningful impact.
But, even** if we add independent variables which are not meaningful**, will it improve R-Square value?

Yes, this is the basic problem with R-Square. How many junk independent variables or important independent variable or impactful independent variable you add to your model, the R-Squared value will always increase. It will never decrease with the addition of a newly independent variable, whether it could be an impactful, non-impactful, or bad variable, so we need another way to measure equivalent RSquare, which penalizes our model with any junk independent variable.

So, we calculate the **Adjusted R-Square** with a better adjustment in the formula of generic R-square.

Q9. What is Mean Square Error?

The mean squared error tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them.

Giving an intuition
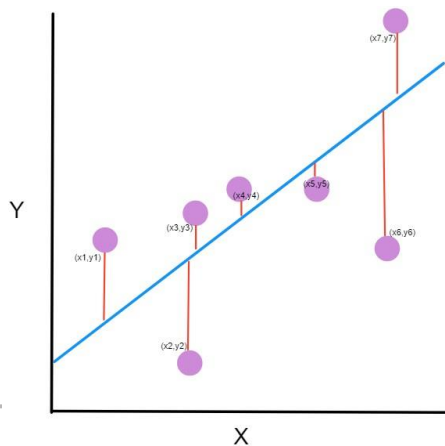
The line equation is **y= mx + b**. We want to find **m (slope)** and **b (y-intercept)** that minimizes the squared error.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Advanced

Q6. What is L1 Regularization (L1 = lasso)?

Ans 6: The main objective of creating a model (training data) is making sure it fits the data properly and reduce the loss. Sometimes the model that is trained which will fit the data, but it may fail and give a poor performance during analysing of data (test data). This leads to overfitting. Regularization came to overcome overfitting.

__Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds "Absolute value of magnitude" of coefficient, as penalty term to the loss function.__

Lasso shrinks the less important feature's coefficient to zero; thus, removing some feature altogether. So, this works well for feature selection in case we have a huge number of features.

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M}x_{ij}W_j)^2 + \lambda\sum_{j=0}^{M}|W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M}x_{ij}W_j)^2 + \lambda\sum_{j=0}^{M}W_j^2$$

Loss function   Regularization Term

Methods like Cross-validation, Stepwise Regression are there to handle overfitting and perform feature selection work well with a small set of features. These techniques are good when we are dealing with a large set of features.

Along with shrinking coefficients, the **lasso performs feature selection**, as well.

(Remember the 'selection in the lasso full-form?)
Because some of the coefficients become exactly zero, which is equivalent to the feature being excluded from the model.

Q7. What is L2 Regularization (L2 = Ridge Regression)

**Ans 7: To avoid overfitting, we perform Regularization**.

As we saw in previous question, Cost function for model accurate behaviour is:

$$Cost\ function\ =\ Loss\ +\frac{\lambda}{2m}\ *\ \sum \|w\|^2$$

Overfitting happens when the model learns signal as well as noise in the training data and would not perform well on new/unseen data on which model wasn't trained on.

To avoid overfitting your model on training data like **cross-validation sampling**, **reducing the number of features, pruning**, **regularization**, etc.

So, to avoid overfitting, we perform Regularization.
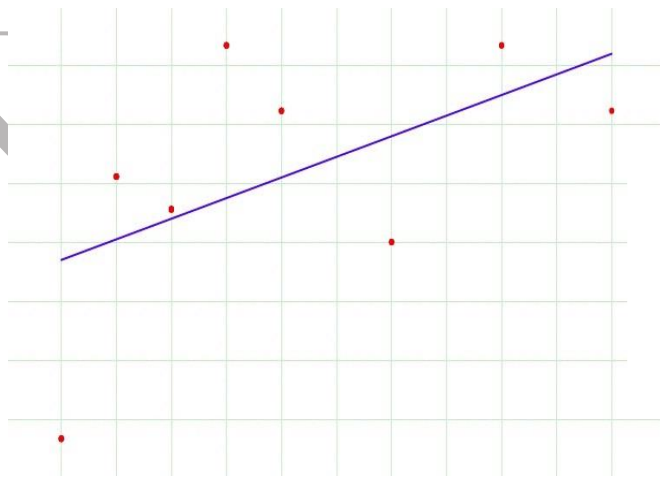
The Regression model that uses L2 regularization is called Ridge Regression. The formula for Ridge Regression: -

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda \sum_{j=1}^{n}\theta_j^2}\right]$$

$$\min_{\theta} J(\theta)$$

Regularization adds the penalty as model complexity increases. The regularization parameter (lambda) penalizes all the parameters except intercept so that the model generalizes the data and won't overfit.

Ridge Regression: -

Ridge regression adds "squared magnitude of the coefficient" as penalty term to the loss function. Here the box part in the above image represents the L2 regularization element/term.

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Lambda is a hyperparameter.

If lambda is zero, then it is equivalent to OLS. But **if lambda is exceptionally large, then it will add too much weight, and it will lead to under-fitting**.

Ridge regularization **forces the weights to be small but does not make them zero** and **does not give the sparse solution**.

Ridge is **not robust to outliers** as square terms blow up the error differences of the outliers, and the regularization term tries to fix it by penalizing the weights.

Ridge regression performs better when all the input features influence the output, and all with **weights are of roughly equal size**.
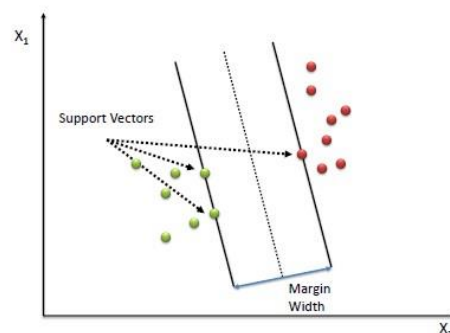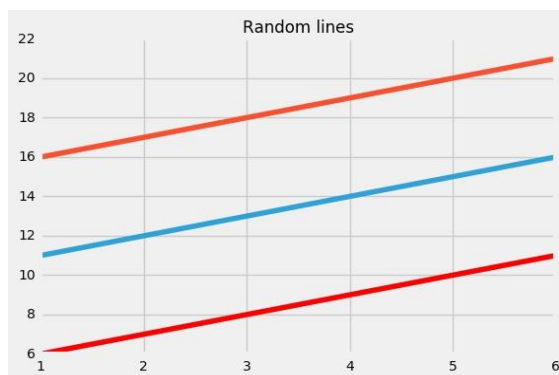
L2 regularization can **learn complex data patterns**.

Q10. **Why Support Vector Regression? Difference between SVR and a simple regression model?**

Ans 10: In simple linear regression, try to minimize the error rate. But in SVR, we try to fit the error within a certain threshold.
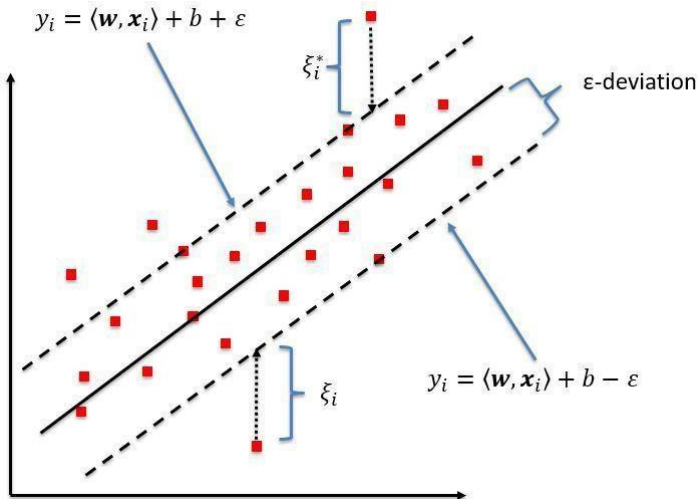
Main Concepts: -
1. **Boundary**
2. **Kernel**
3. **Support Vector**
4. **Hyper Plane**

Blueline: Hyper Plane; Red Line: Boundary-Line

Our best fit line is the one where the hyperplane has the maximum number of points.
We are trying to do here is trying to decide a decision boundary at 'e' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

$$y_i = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b + \varepsilon$$

ξ_i^*

ε-deviation

$$y_i = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b - \varepsilon$$

ξ_i