	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	The machine learns by using labelled data	The machine is trained on unlabelled data without any guidance	An agent interacts with its environment by producing actions & discovers errors or rewards
Type of problems	Regression & Classification	Association & Clustering	Reward based
Type of data	Labelled data	Unlabelled data	No pre-defined data
Training	External supervision	No supervision	No supervision
Approach	Map labelled input to known output	Understand patterns and discover output	Follow trail and error method
Popular algorithms	Linear regression, Logistic regression, Support Vector Machine, KNN, etc	K-means, C-means, etc	Q-Learning, SARSA, etc

1. **Research**: Spend the time to understand what the data science team in each organization is working on. You'll do better in the interview process, and you'll be able to relate better to future colleagues. You'll be asked a lot of situational and product questions that have to do with current work the company is undertaking, whether it's People You May Know with Linkedin or determining how drivers should be matched with passengers with Uber.
2. **Prepare** for four categories of data science questions:

- **Statistics**
- **Probability**
- **Programming**
- **Business** thinking
- **Current Role**

Practice statistical modeling/reasoning, describing machine learning concepts, work in SQL, R, and Python from the basics to more advanced work under time constraints. The data science interview process is pretty standard across companies: **phone screens**, **tests**, and then **on-site interviews.** You'll want to make sure you come off well in interviews and time-constrained assignments.

3. **Practice** using SQL, R, and Python under time constraints. A lot of take-home assignments try to catch you by surprise on this and test your familiarity with the languages with very little time. Showing you can think in frameworks like Hadoop at speed is impressive for these hiring companies, but don't forget the basics too! Sometimes companies will ask basic statistical questions to make sure you're on top of your game.
4. **Get a referral**. Four out of nine companies we surveyed had internal referral as the top source of interviews (Google, Uber, Facebook, Airbnb), and overall, it was the

second largest source of interviews. You'll want to get to know people in the company and get them to advocate for you rather than just applying online.

5. **Prepare your story**. You'll be asked to go over your past work in detail. Be prepared to run over everything you've done with as much specificity as possible, from the tools you used, to why you made different decisions. Be ready to weave a coherent narrative of how the amazing things you did improved business outcomes. Prepare for a long, drawn-out process. Interviewing for a data science position can take months and multiple stages. Make sure you're ready for the wait.

Data Preprocessing and Feature Engineering:

1. What is the importance of data cleaning?
    a. Methods and practices
    b. Can you provide an example of where you have worked with data that had not been cleaned properly and explain what actions you took to rectify the issue?
2. What is data sampling?
3. Missing Values
    a. Reason
    b. Method to handle
    c. Project example
4. Outliers
    a. Reason
    b. Method to handle
    c. Project example
5. Feature engineering
    a. Kind of Feature engineering
    b. Method of Feature engineering
    c. Project example
6. How to handle skewed data
    a. Training
    b. Cross validation
    c. Project example
7. What is Hypothesis Generation? What is the difference between Null Hypothesis (Ho) and Alternate Hypothesis (Ha)?

    Answer:

    Hypothesis generation is a process of creating a set of features which could influence the target variable given a confidence interval (taken as 95% all the time). We can do this before looking at the dataset to avoid biased thoughts. This step often helps in creating new features.
    Defining a hypothesis has two parts:

    1. Null Hypothesis (Ho)

2. Alternate Hypothesis(Ha).

Ho - There exists no impact of a particular feature on the dependent variable.
Ha - There exists a direct impact of a particular feature on the dependent variable.

Based on a decision criterion (say, 5% significance level), we always 'reject' or 'fail to reject' the null hypothesis in statistical parlance. Practically, while model building, we look for probability (p) values. If p value < 0.05, we reject the null hypothesis. If p > 0.05, we fail to reject the null hypothesis.

8. Explain various plots and grids available for data exploration in seaborn and matplotlib libraries?
   Answer: Joint Plot, Distribution Plot, Box Plot, Bar Plot, Regression Plot, Strip Plot, Heatmap, Violin Plot, Pair Plot and Grid, Facet Grid
9. Which Machine Learning Algorithms require Feature Scaling (Standardization and Normalization) and which not?
   Answer:
   Feature Scaling (Standardization and Normalization) is one of the important steps while preparing the data. Whether to use feature scaling or not depends upon the algorithm you are using. Some algorithms require feature scaling and some don't.

   Algorithms which require Feature Scaling (Standardization and Normalization)
   Any machine learning algorithm that computes the distance between the data points needs Feature Scaling (Standardization and Normalization). This includes all curve based algorithms.

   Example:
   1. KNN (K Nearest Neigbors)
   2. SVM (Support Vector Machine)
   3. Logistic Regression
   4. K-Means Clustering

   Algorithms that are used for matrix factorization, decomposition and dimensionality reduction also require feature scaling.

   Example:
   1. PCA (Principal Component Analysis)
   2. SVD (Singular Value Decomposition)

   Why is Feature Scaling required?
   Consider a dataset which contains the age and salary of the employee. Now age is a 2 digit number while salary would be of 5 to 6 digit. If we don't scale both the features (age and salary), salary will adversely affect the accuracy of the algorithm (if the algorithm is distance based). So, if we don't scale the features, then large scale features will dominate the small scale features due to which algorithm will produce wrong predictions.

Algorithms which don't require Feature Scaling (Standardization and Normalization)
The algorithms which rely on rules like tree based algorithms don't require Feature Scaling (Standardization and Normalization).

Example:
1. CART (Classification and Regression Trees)
2. Random Forests
3. Gradient Boosted Decision Trees

Algorithms that rely on distributions of the variables also don't need feature scaling.
Example:
1. Naive Bayes

Machine Learning Questions:

10. **Mention the difference between Data Mining and Machine learning?**
11. **What is hypothesis function?**
12. Name a few libraries in Python used for Data Analysis and Scientific Computations.
13. **What is 'Overfitting' in Machine learning?**
14. **What is 'Training set' and 'Test set'?**
15. What is curse of dimensionality?
16. **How to handle categorical variables?**
17. **How to handle Outlier in your data set?**
18. Assumptions of linear regression?
19. **Methods or cost function to find the best fit line for data in Linear Regression?**
20. **Which algorithm is Lazy learning algorithm?**
21. **What is ensembling and when to use?**
22. **Bagging and boosting**
23. Describe the working of gradient boost.
24. absolute error instead of squared error

Answer:
Minimizing the squared error (L2) over a set of numbers results in finding its mean, and minimizing the absolute error (L1) results in finding its median. (And minimizing the L0 error results in finding the modes.)

As for when each loss function is most appropriate, the most basic differences are the using the squared error is easier to solve for and using the absolute error is more robust to outliers. More specifically, if we have a set of N numbers y0,…,yN, the value of z that minimizes the equation $\sum i(yi-z)2$ is the mean of the y's: $z=\sum_i y^i/N$
The value of z that minimizes the equation $\sum_i |y^i-z|$ is the median of the y's.

Again, among the main differences between the two are that using the squared error is easier to solve for and using the absolute error is more robust to outliers.

The reason that the squared error is easier to solve for is that the derivatives are continuous. In the case of linear regression, this means that you can find the solution in closed form (by setting the derivative to zero). Linear regression with absolute error requires an iterative approach, which is more complicated and isn't as efficient.

Squared error approach penalizes large errors more as compared to absolute error approach If you think that outliers are merely corrupted data that should be somewhat ignored, then absolute error might be better to use. If you want to avoid very large errors and still fit outliers somewhat reasonably, then squared error might be better to use.

25. What are benefits and weaknesses of various binary classification metrics?
26. What is an intuitive explanation of regularization?
27. Why might it be preferable to include fewer predictors over many?

Answer: Too many features in your hypothesis can lead to over-fitting and the model will not do well with new data points.

Also using too many features means getting more data ! It is not always possible to get all the data , so missing/sparse data can be very dangerous for your model's performance.

To avoid the problem of over-fitting

-Reduce the number of feature by manually selecting only required features or using a model selection algorithm.

- Use regularization: if you throw away lot of feature which are actually useful then regularization is much more helpful then just reducing the number of features.

28. Difference between Gradient Boosting | XG Boost |Light GBM and Cat boost?
29. Why XG boost is so fast?
30. How does parallel processing in XG Boost works? (Remember it is boosting so tree are dependent on the above tree)
31. Why Light GBM is faster than XG Boost?
32. When would you use GD over SDG, and vice-versa?

Answer: GD theoretically minimizes the error function better than SGD. However, SGD converges much faster once the dataset becomes large.That means GD is preferable for small datasets while SGD is preferable for larger ones.

In practice, however, SGD is used for most applications because it minimizes the error function well enough while being much faster and more memory efficient for large datasets.

33. p-value and Chi Square value

The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no linear relation or no correlation between target value and predictor). A low p-value (< 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable.
Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.
In the output below, we can see that the predictor variables of South and North are significant because both of their p-values are 0.000. However, the p-value for East (0.092) is greater than the common alpha level of 0.05, which indicates that it is not statistically significant.

Coefficients

Term	Coef	SE Coef	T	P
Constant	389.166	66.0937	5.8881	0.000
East	2.125	1.2145	1.7495	0.092
South	5.318	0.9629	5.5232	0.000
North	-24.132	1.8685	-12.9153	0.000

Typically, you use the coefficient p-values to determine which terms to keep in the regression model. In the model above, we should consider removing East.

chi-square test for independence. This Chi-Square test tells us whether two categorical variables depend on each other. The test is applied when you have two categorical variables from a single population. It is used to determine whether there is a significant association between the two variables. A small chi-square value means that data fits A high chi-square value means that data doesn't fit.

For example, in an election survey, voters might be classified by gender (male or female) and voting preference (Democrat, Republican, or Independent). We could use a chi-square test for independence to determine whether gender is related to voting preference

34. Z-test

In a z-test, the sample is assumed to be normally distributed. A z-score is calculated with population parameters such as **"population mean"** and **"population standard deviation"** and **is used to validate a hypothesis that the sample drawn belongs to the same population.**

35. T-test

A t-test is used to compare the mean of two given samples. Like a z-test, a t-test also assumes a normal distribution of the sample. A t-test is used when the population parameters (mean and standard deviation) are not known.

36. Multicollinearity occurs when independent variables in a regression model are correlated. This correlation is a problem because independent variables should be *independent*. If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.

    Multicollinearity causes the following two basic types of problems:

    - The coefficient estimates can swing wildly based on which other independent variables are in the model. The coefficients become very sensitive to small changes in the model.
    - Multicollinearity reduces the precision of the estimate coefficients, which weakens the statistical power of your regression model. You might not be able to trust the p-values to identify independent variables that are statistically significant.

    ## Testing for Multicollinearity with Variance Inflation Factors (VIF)

    If you can identify which variables are affected by multicollinearity and the strength of the correlation, you're well on your way to determining whether you need to fix it.There is a very simple test to assess multicollinearity in your regression model. The variance inflation factor (VIF) identifies correlation between independent variables and the strength of that correlation.

    If the VIF is equal to 1 there is no multicollinearity among factors, but if the VIF is greater than 1, the predictors may be moderately correlated. The output above shows that the VIF for the Publication and Years factors are about 1.5, which indicates some correlation, but not enough to be overly concerned about. A VIF between 5 and 10 indicates high correlation that may be problematic. And if the VIF goes above 10, you can assume that the regression coefficients are poorly estimated due to multicollinearity.

The potential solutions include the following:

- Remove some of the highly correlated independent variables.
- Linearly combine the independent variables, such as adding them together.
- Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

37. Matrix Factorization: A matrix decomposition is a way of reducing a matrix into its constituent parts.

    - It is an approach that can simplify more complex matrix operations that can be performed on the decomposed matrix rather than on the original matrix itself.
    - A common analogy for matrix decomposition is the factoring of numbers, such as the factoring of 10 into 2 x 5. For this reason, matrix decomposition is also called matrix factorization. Like factoring real values, there are many ways to decompose a matrix, hence there are a range of different matrix decomposition techniques.
    - Simple and widely used matrix decomposition methods are the LU matrix decomposition, QR matrix decomposition and Cholesky Decomposition.
    - **LU decomposition, the QR decomposition** is often used to solve systems of linear equations, although is not limited to square matrices.
    - The **Cholesky decomposition** is used for solving linear least squares for linear regression, as well as simulation and optimization methods.
    - Matrix factorization can be used in various domain such as image recognition, Recommendation. Matrix used in this type of problem are generally sparse because there is chance that one user might rate only some movies. There are various applications for matrix factorization such as Dimensionality reduction, latent value decomposition

38.
39. **Which method/cost function is used to find the best fit line for data in Linear Regression?**
40. Why is mean square error a bad measure of model performance? What would you suggest instead?
41. Compare logistic regression w. decision trees, neural networks. How have these technologies been vastly improved over the last 15 years?
42. Is it better to have too many false positives, or too many false negatives?
43. **Why not absolute error cost function in linear regression?**
44. **Is linear regression sensitive to outliers?**
45. How does outliers effect are models?
46. Why Gradient descent if linear regression can be solved using matrix multiplications/factorization?
47. What is a concave and convex function? Why we prefer convex objective functions?
48. **What do you expect will happen with bias and variance as you increase the size of training data?**
49. **Lasso and Ridge Regression and which one is used for variable selection?**
50. **Which method/cost function is used to find the best fit classification in logistic Regression?**
51. **Evaluation metric used in Logistic regression: AUC/ROC, LogLoss**
52. What's the trade-off between bias and variance?
53. What is the difference between supervised and unsupervised machine learning?
54. Explain how a ROC curve works.
55. **Define precision and recall.**

56. **Curse of dimensionality?**
57. What's the difference between a generative and discriminative model
58. What is a confidence interval?

59. What are the problems with K means clustering?
60. How do you decide optimal number of clusters?
61. How do you evaluate goodness of clusters? (basic intuition is quantify intra and inter clusters distance so that intra cluster is minimum and inter cluster is maximum
62. How do we interpret components of PCA? What do they represent?
63. What is Non Negative matrix Factorization?
64. How do you visualize high dimensional data?
65. When to use PCA or Non Negative Matrix factorization?
66.
67. **What's the F1 score? How would you use it?**
68. What is a Confusion Matrix?
69. What are collinearity and multicollinearity?
70. Explain false negative, false positive, true negative and true positive with a simple example.
71. **How is KNN different from k-means clustering?**
72. **Which of the machine learning algorithm can be used for imputing missing values of both categorical and continuous variables? KNN**
73. **Different kind of Distance Metric used?**
74. When you increase the k in KNN what will happen to bias and variance? bias will increase while less variance
75. **When you find noise in data which of the following option would you consider in k-NN?**
76. **In k-NN it is very likely to overfit due to the curse of dimensionality. Which of the following option would you consider to handle such problem?** either dimensionality reduction algorithm or the feature selection algorithm
77. **What is Bayes' Theorem? How is it useful in a machine learning context?**
78. **Why is "Naive" Bayes naive?**
79. **What is parametric and Non Parametric Algorithms?**
80. **Explain the difference between L1 and L2 regularization.**
81. How will you define the number of clusters in a clustering algorithm
82. **What's the difference between Type I and Type II error?**
    Answer: "A type I error occurs when the null hypothesis is true, but is rejected. A type II error occurs when the null hypothesis is false, but erroneously fails to be rejected."
83. What is selection bias?
84. What is sampling bias?
85. **What is Anomaly detection ?**
    Answer: Anomaly detection is the identification of data points, items, observations or events that do not conform to the expected pattern of a given group. These anomalies occur very infrequently but may signify a large and significant threat such as cyber intrusions or fraud. **Outliers are a kind of anamoly.**

86. What do you understand by the term Normal Distribution? Explain
87. How can you deploy ML model in production?
    Answer: We have 2 libraries for model deployment in python pickle and joblib.
    You can easily deploy your model using flask api.
88. How can you improve model performance ?
    Answer:
    You can improve the model performance by using following methods

    - Add more data
    - Treat missing and Outlier values
    - Feature Engineering
    - Feature Selection
    - Multiple algorithms
    - Algorithm Tuning
    - Ensemble methods
    - Cross Validation
89. **What are best dimensionality reduction algorithms ?**

    Answer:

    - Missing Value Ratio
    - Low Variance Filter
    - High Correlation Filter
    - Random Forest
    - Backward Feature Elimination
    - Forward Feature Selection
    - Factor Analysis
    - Principal Component Analysis
    - Independent Component Analysis
    - t-Distributed Stochastic Neighbor Embedding (t-SNE)
    - UMAP
90. **What's a Fourier transform?**
91. What is the difference between L1 and L2 regularization? How does it solve the problem of overfitting? Which regularizer to use and when?
    Answer: Regularization helps to solve over fitting problem in machine learning. Simple model will be a very poor generalization of data. At the same time, complex model may not perform well in test data due to over fitting. We need to choose the right model in between simple and complex model. Regularization helps to choose preferred model complexity, so that model is better at predicting. Regularization is nothing but adding a penalty term to the objective function and control the model complexity using that penalty term.
    The ridge regression uses $L2$ norm for regularization and adding $\lambda(\sum_{j=1}^{2}\beta^{2}_{j}$

The Lasso regression uses $L1$ norm for regularization and adding $\lambda(\sum_{j=1}^{2}|\beta_j|$
Typically ridge or ***ℓ2*** penalties are much better for minimizing prediction error rather than ***ℓ1*** penalties. The reason for this is that when two predictors are highly correlated, ***ℓ1*** regularizer will simply pick one of the two predictors. In contrast, the ***ℓ2*** regularizer will keep both of them and jointly shrink the corresponding coefficients a little bit. Thus, while the ***ℓ1*** penalty can certainly reduce overfitting, you may also experience a loss in predictive power.

It is recommended to go for Elastic Nets instead. Granted this will only be a practical option if you are doing linear/logistic regression. But, in that case, Elastic Nets have proved to be (in theory and in practice) better than L1/Lasso. Elastic Nets combine L1 and L2 regularization at the "only" cost of introducing another hyperparameter to tune

92. **What's the difference between probability and likelihood?**
93. **What is deep learning, and how does it contrast with other machine learning algorithms?**
94. **What's the difference between a generative and discriminative model?**
95. **What cross-validation technique would you use on a time series dataset?**
96. **How is a decision tree pruned?**
97. What is Tensorflow?
98. **Which is more important to you– model accuracy, or model performance?**
99. **How would you handle an imbalanced dataset?**
100. **When should you use classification over regression?**
101. **Name an example where ensemble techniques might be useful.**
102. **How do you ensure you're not overfitting with a model?**
103. **How to handle high bias and high variance and how to ensure correct tradeoff for higher accuracy of model?**
104. How do you deal with sparsity?
105. **How would you evaluate a logistic regression model?**
106. **What's the "kernel trick" and how is it useful?**
107. What is: collaborative filtering, n-grams, map reduce, cosine distance?
108. **Describe Entropy gain and Gini Index measures for splitting decision tree**
109. **How do you handle missing or corrupted data in a dataset?**
110. **Which data visualization libraries do you use? What are your thoughts on the best data visualization tools?**
111. Correlation vs. co-variance
112. Is Pearson coefficient sensitive to outliers?
113. Explain random sampling, stratified sampling, and cluster sampling.
114. Is **Standard deviation is robust to outliers?**
115. Probability Distribution function
116. What's the best way to visualize this data and how would you do that using Python/R?
117. What's the difference between structured and unstructured data?
118. If you have more than one trained model, how do you assess which is best?
119. What is the Central Limit Theorem and why is it important?

120. Z score
121. What exploratory data analysis? example
122. What is sampling? How many sampling methods do you know?
123. Difference between Interpolation and Extrapolation
124. What do you understand by Eigenvectors and Eigenvalues?
125. **To test linear relationship of y(dependent) and x(independent) continuous variables, which of the following plot best suited? Scatter Plot**
126. t-test
127. chi-square test
128. Poisson distributions
129. time series analysis

Neural Networks:

1. Gradient vanishing and Gradient Explode
    - Give example and scenario
    - Why it happens?
    - How to detect?
    - How to rectify?
    - Best practice to avoid??
2. Different activation functions in DL?

NLP:

Some Common Steps for NLP Problems:

- Sentence Segmentation: break the text apart into separate sentences
- Tokenization: split Sentence to words
- Stemming: process of reducing words to their word stem for example thinking→ think
- Lemmatizing: for example worse→ bad
- POS tags: Predicting Parts of Speech for Each Token
- Identifying Stop Words: like “and”, “the”
- Name entity recognition: detect nouns with the real world concepts.
- Text classification
- Chunking
- Coreference resolution

Applications of NLP in The Real World:

- Personal assistant applications
- Fighting spam
- Chatbots
- Managing the Advertisement

- Sentiment analysis
- Text classification
- Text summarization
- Toxicity Classification
- Name entity recognization
- Part of speech tagging
- Language model building
- Machine translation
- Spell checking
- Speech recognition
- Character recognition

Python Library for NLP:

- NLTK
- spaCy
- Gensim : is a python library specifically for Topic Modelling.
- Pattern
- Stanford CoreNLP
- Polyglot
- TextBlob
- re: python library for regular expression
- WordCloud
- allennlp: an open-source NLP research library, built on PyTorch

Word Embedding Libraries:

- Word2vec
- Glove
- Fasttext
- Genism

1. What is TF/IDF vectorization?
2.

Recommender Systems:

- What are Recommender Systems

Programming Questions:

1. **What are some differences between a linked list and an array?**
2. **Describe a hash table.**
3. What are hash table collisions?

Python Questions:

1. What is the main difference between a Pandas series and a single-column DataFrame in Python?
2. What is Lambada function in python?
3. **How** can you handle duplicate values in a dataset for a variable in Python?
4. In Python, how is memory managed?
   In Python, memory is managed in a private heap space. This means that all the objects and data structures will be located in a private heap. However, the programmer won't be allowed to access this heap. Instead, the Python interpreter will handle it. At the same time, the core API will enable access to some Python tools for the programmer to start coding. The memory manager will allocate the heap space for the Python objects while the inbuilt garbage collector will recycle all the memory that's not being used to boost available heap space
5. How do you find percentile? Write the code for it.
6. Create a function that checks if a word is a palindrome.
7. Find max sum subsequence from a sequence of values

SQL Questions:

1. **What is the purpose of the group functions in SQL? Give some examples of group functions.**
2. Tell me the difference between an inner join, left join/right join, and union.
3. If a table contains duplicate rows, does a query result display the duplicate values by default? How can you eliminate duplicate rows from a query result?
4. How would you sort a table in sql?
5. ***Write a query that returns the name of each department and a count of the number of employees in each.***

EMPLOYEES containing: Emp_ID (Primary key) and Emp_Name
EMPLOYEE_DEPT containing: Emp_ID (Foreign key) and Dept_ID (Foreign key)
DEPTS containing: Dept_ID (Primary key) and Dept_Name

```
Select Dept_Name, count(1)
from DEPTS a right join EMPLOYEE_DEPT b on a.Dept_id = b.Dept_id
```

Group By Dept_Name

Domain Specific Questions:

6. **How would you implement a recommendation system for our company's users?**
7. **How can we use your machine learning skills to generate revenue?**
8. **Describe few industry problems in your industry where you can find ML as potential problem solver.**
9. **What do you think of our current data process?**

Project Questions:

1. Why did you choose to do a project about this?
2. What does this project mean to you?
3. What was your favorite thing about working on this project?
4. What was your least favorite thing about working on this project?
5. What technical challenges did you face during this project and how did you overcome them?
6. Where did you get this data set, and what techniques did you use to clean the data?
7. Why did you choose to use the statistical techniques you used for this project?
8. Why did you choose to use the programming techniques you used for this project?
9. Could you explain how this algorithm/statistical technique/section of code works?
10. What libraries, packages, or other tools did you use for this project?
11. How long did it take you to put this project together?
12. If asked to, how might you expand on this project?
13. If you had to do it again, what might you change about this project?
14. How will the skills you used on this project be valuable to our business?
15. (If group project) What was your job on this project?
16. (If group project) How was this project organized and version-controlled?
17. (If group project) Can you talk about a conflict or disagreement you had with teammates during this project and you overcame it?

If you went for real time job interviews, you will get scenario based project questions on machine learning.

18. What is Problem statement?
19. How you plan to handle it?
20. How you collected data from different sources?

21. How you done preprocessing?
22. How you have chosen machine leaching model?
23. How you have optimized that particular model?
24. How you have deployed that model to production?
25.

Culture Fit:

1. What do you think makes a good data scientist?
2. How did you become interested in data science?
3. Give a few examples of “best practices” in data science.
4. What is the latest data science book / article you read? What is the latest data mining conference / webinar / class / workshop / training you attended?
5. What’s a project you would want to work on at our company?
6. What unique skills do you think you’d bring to the team?
7. What data would you love to acquire if there were no limitations?
8. Have you ever thought about creating your own startup? Around which idea / concept?

General Questions:

1. **What are the last machine learning papers you’ve read?**
2. **Do you have research experience in machine learning?**
3. **What are your favorite use cases of machine learning models?**
4. **How would you approach the “Netflix Prize” competition?**
5. Give a few examples of "best practices" in data science.
6. Describe complete machine learning Pipeline from raw data to final model?
   Check out Quandl for economic and financial data, and Kaggle’s Datasets collection for another great list.

7. **How do you think Google is training data for self-driving cars?**
8. How to compute an inverse matrix faster by playing around with some computational tricks?(Gaussian elimination method)
9. In your opinion, is having more data a good thing?
10. How will you reduce long model training time?
11. What are the basic steps taken when analyzing a project?
12. Describe a project in which you encountered an obstacle and explain how you handled it.
13. What is better: good data or good models? And how do you define "good"? Is there a universal good model? Are there any models that are definitely not so good?

14. You have a data set containing 100,000 rows and 100 columns, with one of those columns being our dependent variable for a problem we'd like to solve. How can we quickly identify which columns will be helpful in predicting the dependent variable. Identify two techniques and explain them to me as though I were 5 years old.
15. How would you detect bogus reviews, or bogus Facebook accounts used for bad purposes?
16. How would you perform clustering on a million unique keywords, assuming you have 10 million data points—each one consisting of two keywords, and a metric measuring how similar these two keywords are? How would you create this 10 million data points table in the first place?
17. Is it better to spend 5 days developing a 90% accurate solution, or 10 days for 100% accuracy?
18. Do you think 50 small decision trees are better than a large one? Why?
19. What is the biggest data set that you processed, and how did you process it, what were the results?
20. Tell me two success stories about your analytic or computer science projects? How was lift (or success) measured?
21. What are your favorite data visualization techniques?
22. How would you effectively represent data with 5 dimensions?
23. K-means clustering
    - what is the loss function? when does it converge?
    - know the algorithm iteration steps
    - how do you choose the right number of clusters?
    - is the optimization convex?
24. You are given a data set consisting of variables having more than 30% missing values? Let's say, out of 50 variables, 8 variables have missing values higher than 30%. How will you deal with them?
25. **A company has build a kNN classifier that gets 100% accuracy on training data. When they deployed this model on client side it has been found that the model is not at all accurate. Which of the following thing might gone wrong? Overfittting**
26. **It is possible to construct a 2-NN classifier by using the 1-NN classifier?** You can implement a 2-NN classifier by ensembling 1-NN classifiers
27. You have trained a k-NN model and now you want to get the prediction on test data. Before getting the prediction suppose you want to calculate the time taken by k-NN for predicting the class for test data.(N= Number of records and D= dimension) ND
28. How would you predict who will renew their subscription next month Netflix? What data would you need to solve this? What analysis would you do? Would you build predictive models? If so, which algorithms?
29. **Suppose we fit "Lasso Regression" to a data set, which has 100 features (X1,X2…X100). Now, we rescale one of these feature by multiplying with 10 (say that feature is X1), and then refit Lasso regression with the same regularization parameter.** It is more likely for X1 to be included in the model
30. **Which of the following is true about "Ridge" or "Lasso" regression methods in case of feature selection?** "Ridge regression" will use all predictors in final model whereas "Lasso regression" can be used for feature selection because coefficient values can be zero.

31. **A Pearson correlation between two variables is zero but, still their values can still be related to each other.** $Y=X^2$ Note that, they are not only associated, but one is a function of the other and Pearson correlation between them is 0.
32. Imagine, you are solving a classification problem with highly imbalanced class. The majority class is observed 99% of times in the training data. Your model has 99% accuracy after taking the predictions on test data. **? What will you use** Accuracy or Precision recall for accuracy?
33. **N-grams are defined as the combination of N keywords together. How many bi-grams can be generated from given sentence:**
    "IT Bodhi is a great place to learn data science" Total 9
    IT Bodhi, Bodhi is, is a, a great, great place, place to, To learn, learn data, data science
34. You are given a cancer detection data set. Let's suppose when you build a classification model you achieved an accuracy of 96%. Why shouldn't you be happy with your model performance? What can you do about it?

You can do the following:

- Add more data
- Treat missing outlier values
- Feature Engineering
- Feature Selection
- Multiple Algorithms
- Algorithm Tuning
- Ensemble Method
- Cross-Validation

35. Suppose you found that your model is suffering from low bias and high variance. Which algorithm you think could tackle this situation and Why?

Type 1: How to tackle high variance?

- Low bias occurs when the model's predicted values are near to actual values.
- In this case, we can use the bagging algorithm (eg: Random Forest) to tackle high variance problem.
- Bagging algorithm will divide the data set into its subsets with repeated randomized sampling.
- Once divided, these samples can be used to generate a set of models using a single learning algorithm. Later, the model predictions are combined using voting (classification) or averaging (regression).

Type 2: How to tackle high variance?

- Lower the model complexity by using regularization technique, where higher model coefficients get penalized.

- You can also use top n features from variable importance chart. It might be possible that with all the variable in the data set, the algorithm is facing difficulty in finding the meaningful signal.

36. You're asked to build a random forest model with 10000 trees. During its training, you got training error as 0.00. But, on testing the validation error was 34.23. What is going on? Haven't you trained your model perfectly?

- The model is overfitting the data.
- Training error of 0.00 means that the classifier has mimicked the training data patterns to an extent.
- But when this classifier runs on the unseen sample, it was not able to find those patterns and returned the predictions with more number of errors.
- In Random Forest, it usually happens when we use a larger number of trees than necessary. Hence, to avoid such situations, we should tune the number of trees using cross-validation.

37. **Recommendations on Amazon are based on which algorithm?** collaborative filtering.

Q1. You are given a train data set having 1000 columns and 1 million rows. The data set is based on a classification problem. Your manager has asked you to reduce the dimension of this data so that model computation time can be reduced. Your machine has memory constraints. What would you do? (You are free to make practical assumptions.)

Answer: Processing a high dimensional data on a limited memory machine is a strenuous task, your interviewer would be fully aware of that. Following are the methods you can use to tackle such situation:

1. Since we have lower RAM, we should close all other applications in our machine, including the web browser, so that most of the memory can be put to use.
2. We can randomly sample the data set. This means, we can create a smaller data set, let's say, having 1000 variables and 300000 rows and do the computations.

3. To reduce dimensionality, we can separate the numerical and categorical variables and remove the correlated variables. For numerical variables, we'll use correlation. For categorical variables, we'll use chi-square test.
4. Also, we can use PCA and pick the components which can explain the maximum variance in the data set.
5. Using online learning algorithms like Vowpal Wabbit (available in Python) is a possible option.
6. Building a linear model using Stochastic Gradient Descent is also helpful.
7. We can also apply our business understanding to estimate which all predictors can impact the response variable. But, this is an intuitive approach, failing to identify useful predictors might result in significant loss of information.

Note: For point 4 & 5, make sure you read about online learning algorithms & Stochastic Gradient Descent. These are advanced methods.

1.Missing Values Ratio
Data columns with too many missing values are unlikely to carry much useful information. Thus data columns with number of missing values greater than a given threshold can be removed. The higher the threshold, the more aggressive the reduction.

2.Low Variance Filter
Similarly to the previous technique, data columns with little changes in the data carry little information. Thus all data columns with variance lower than a given threshold are removed. A word of caution: variance is range dependent; therefore normalization is required before applying this technique.

3.High Correlation Filter.
Data columns with very similar trends are also likely to carry very similar information. In this case, only one of them will suffice to feed the machine learning model. Here we calculate the correlation coefficient between numerical columns and between nominal columns as the Pearson's Product Moment Coefficient and the Pearson's chi square value respectively. Pairs of columns with correlation coefficient higher than a threshold are reduced to only one. A word of caution: correlation is scale sensitive; therefore column normalization is required for a meaningful correlation comparison.

4.Random Forests / Ensemble Trees

Decision Tree Ensembles, also referred to as random forests, are useful for feature selection in addition to being effective classifiers. One approach to dimensionality reduction is to generate a large and carefully constructed set of trees against a target attribute and then use each attribute's usage statistics to find the most informative subset of features. Specifically, we can generate a large set (2000) of very shallow trees (2 levels), with each tree being trained on a small fraction (3) of the total number of attributes. If an attribute is often selected as best split, it is most likely an informative feature to retain. A score calculated on the attribute usage

statistics in the random forest tells us – relative to the other attributes – which are the most predictive attributes.

5.Backward Feature Elimination

In this technique, at a given iteration, the selected classification algorithm is trained on n input features. Then we remove one input feature at a time and train the same model on n-1 input features n times. The input feature whose removal has produced the smallest increase in the error rate is removed, leaving us with n-1 input features. The classification is then repeated using n-2 features, and so on. Each iteration k produces a model trained on n-k features and an error rate e(k). Selecting the maximum tolerable error rate, we define the smallest number of features necessary to reach that classification performance with the selected machine learning algorithm.

6.Forward Feature Construction.

This is the inverse process to the Backward Feature Elimination. We start with 1 feature only, progressively adding 1 feature at a time, i.e. the feature that produces the highest increase in performance. Both algorithms, Backward Feature Elimination and Forward Feature Construction, are quite time and computationally expensive. They are practically only applicable to a data set with an already relatively low number of input columns.

Q2. Is rotation necessary in PCA? If yes, Why? What will happen if you don't rotate the components?

Answer: Yes, rotation (orthogonal) is necessary because it maximizes the difference between variance captured by the component. This makes the components easier to interpret. Not to forget, that's the motive of doing PCA where, we aim to select fewer components (than features) which can explain the maximum variance in the data set. By doing rotation, the relative location of the components doesn't change, it only changes the actual coordinates of the points.

If we don't rotate the components, the effect of PCA will diminish and we'll have to select more number of components to explain variance in the data set.

Q3. You are given a data set. The data set has missing values which spread along 1 standard deviation from the median. What percentage of data would remain unaffected? Why?

Answer: This question has enough hints for you to start thinking! Since, the data is spread across median, let's assume it's a normal distribution. We know, in a normal distribution, ~68% of the data lies in 1 standard deviation from mean (or mode, median), which leaves ~32% of the data unaffected. Therefore, ~32% of the data would remain unaffected by missing values.

Q4. You are given a data set on cancer detection. You've build a classification model and achieved an accuracy of 96%. Why shouldn't you be happy with your model performance? What can you do about it?

Answer: If you have worked on enough data sets, you should deduce that cancer detection results in imbalanced data. In an imbalanced data set, accuracy should not be used as a measure of performance because 96% (as given) might only be predicting majority class correctly, but our class of interest is minority class (4%) which is the people who actually got diagnosed with cancer. Hence, in order to evaluate model performance, we should use Sensitivity (True Positive Rate), Specificity (True Negative Rate), F measure to determine class wise performance of the classifier. If the minority class performance is found to to be poor, we can undertake the following steps:

1. We can use undersampling, oversampling or SMOTE to make the data balanced.
2. We can alter the prediction threshold value by doing probability caliberation and finding a optimal threshold using AUC-ROC curve.
3. We can assign weight to classes such that the minority classes gets larger weight.
4. We can also use anomaly detection.

Know more: Imbalanced Classification

Q5. Why is naive Bayes so 'naive' ?

Answer: naive Bayes is so 'naive' because it assumes that all of the features in a data set are equally important and independent. As we know, these assumption are rarely true in real world scenario.

Q6. Explain prior probability, likelihood and marginal likelihood in context of naiveBayes algorithm?

Answer: Prior probability is nothing but, the proportion of dependent (binary) variable in the data set. It is the closest guess you can make about a class, without any further information. For example: In a data set, the dependent variable is binary (1 and 0). The proportion of 1 (spam) is

70% and 0 (not spam) is 30%. Hence, we can estimate that there are 70% chances that any new email would be classified as spam.

Likelihood is the probability of classifying a given observation as 1 in presence of some other variable. For example: The probability that the word 'FREE' is used in previous spam message is likelihood. Marginal likelihood is, the probability that the word 'FREE' is used in any message.

Q7. You are working on a time series data set. You manager has asked you to build a high accuracy model. You start with the decision tree algorithm, since you know it works fairly well on all kinds of data. Later, you tried a time series regression model and got higher accuracy than decision tree model. Can this happen? Why?

Answer: Time series data is known to posses linearity. On the other hand, a decision tree algorithm is known to work best to detect non – linear interactions. The reason why decision tree failed to provide robust predictions because it couldn't map the linear relationship as good as a regression model did. Therefore, we learned that, a linear regression model can provide robust prediction given the data set satisfies its linearity assumptions.

Q8. You are assigned a new project which involves helping a food delivery company save more money. The problem is, company's delivery team aren't able to deliver food on time. As a result, their customers get unhappy. And, to keep them happy, they end up delivering food for free. Which machine learning algorithm can save them?

Answer: You might have started hopping through the list of ML algorithms in your mind. But, wait! Such questions are asked to test your machine learning fundamentals.

This is not a machine learning problem. This is a route optimization problem. A machine learning problem consist of three things:

1. There exist a pattern.
2. You cannot solve it mathematically (even by writing exponential equations).
3. You have data on it.

Always look for these three factors to decide if machine learning is a tool to solve a particular problem.

Q9. You came to know that your model is suffering from low bias and high variance. Which algorithm should you use to tackle it? Why?

Answer: Low bias occurs when the model's predicted values are near to actual values. In other words, the model becomes flexible enough to mimic the training data distribution. While it sounds like great achievement, but not to forget, a flexible model has no generalization capabilities. It means, when this model is tested on an unseen data, it gives disappointing results.

In such situations, we can use bagging algorithm (like random forest) to tackle high variance problem. Bagging algorithms divides a data set into subsets made with repeated randomized sampling. Then, these samples are used to generate a set of models using a single learning algorithm. Later, the model predictions are combined using voting (classification) or averaging (regression).

Also, to combat high variance, we can:

1. Use regularization technique, where higher model coefficients get penalized, hence lowering model complexity.
2. Use top n features from variable importance chart. May be, with all the variable in the data set, the algorithm is having difficulty in finding the meaningful signal.

Q10. You are given a data set. The data set contains many variables, some of which are highly correlated and you know about it. Your manager has asked you to run PCA. Would you remove correlated variables first? Why?

Answer: Chances are, you might be tempted to say No, but that would be incorrect. Discarding correlated variables have a substantial effect on PCA because, in presence of correlated variables, the variance explained by a particular component gets inflated.

For example: You have 3 variables in a data set, of which 2 are correlated. If you run PCA on this data set, the first principal component would exhibit twice the variance than it would exhibit with uncorrelated variables. Also, adding correlated variables lets PCA put more importance on those variable, which is misleading.

Q11. After spending several hours, you are now anxious to build a high accuracy model. As a result, you build 5 GBM models, thinking a boosting algorithm would do the magic. Unfortunately, neither of models could perform better than benchmark score. Finally, you decided to combine those models. Though, ensembled models are known to return high accuracy, but you are unfortunate. Where did you miss?

Answer: As we know, ensemble learners are based on the idea of combining weak learners to create strong learners. But, these learners provide superior result when the combined models

are uncorrelated. Since, we have used 5 GBM models and got no accuracy improvement, suggests that the models are correlated. The problem with correlated models is, all the models provide same information.

For example: If model 1 has classified User1122 as 1, there are high chances model 2 and model 3 would have done the same, even if its actual value is 0. Therefore, ensemble learners are built on the premise of combining weak uncorrelated models to obtain better predictions.

Q12. How is kNN different from kmeans clustering?

Answer: Don't get mislead by 'k' in their names. You should know that the fundamental difference between both these algorithms is, kmeans is unsupervised in nature and kNN is supervised in nature. kmeans is a clustering algorithm. kNN is a classification (or regression) algorithm.

kmeans algorithm partitions a data set into clusters such that a cluster formed is homogeneous and the points in each cluster are close to each other. The algorithm tries to maintain enough separability between these clusters. Due to unsupervised nature, the clusters have no labels.

kNN algorithm tries to classify an unlabeled observation based on its k (can be any number) surrounding neighbors. It is also known as lazy learner because it involves minimal training of model. Hence, it doesn't use training data to make generalization on unseen data set.

Q13. How is True Positive Rate and Recall related? Write the equation.

Answer: True Positive Rate = Recall. Yes, they are equal having the formula (TP/TP + FN).

Know more: Evaluation Metrics

Q14. You have built a multiple regression model. Your model R^2 isn't as good as you wanted. For improvement, your remove the intercept term, your model R^2 becomes 0.8 from 0.3. Is it possible? How?

Answer: Yes, it is possible. We need to understand the significance of intercept term in a regression model. The intercept term shows model prediction without any independent variable i.e. mean prediction. The formula of $R^2 = 1 - \sum(y - y')^2/\sum(y - ymean)^2$ where y' is predicted value.

When intercept term is present, R^2 value evaluates your model wrt. to the mean model. In absence of intercept term (y_{mean}), the model can make no such evaluation, with large denominator, $\sum(y - y')^2/\sum(y)^2$ equation's value becomes smaller than actual, resulting in higher R^2.

Q15. After analyzing the model, your manager has informed that your regression model is suffering from multicollinearity. How would you check if he's true? Without losing any information, can you still build a better model?

Answer: To check multicollinearity, we can create a correlation matrix to identify & remove variables having correlation above 75% (deciding a threshold is subjective). In addition, we can use calculate VIF (variance inflation factor) to check the presence of multicollinearity. VIF value <= 4 suggests no multicollinearity whereas a value of >= 10 implies serious multicollinearity. Also, we can use tolerance as an indicator of multicollinearity.

But, removing correlated variables might lead to loss of information. In order to retain those variables, we can use penalized regression models like ridge or lasso regression. Also, we can add some random noise in correlated variable so that the variables become different from each other. But, adding noise might affect the prediction accuracy, hence this approach should be carefully used.

Know more: Regression

Q16. When is Ridge regression favorable over Lasso regression?

Answer: You can quote ISLR's authors Hastie, Tibshirani who asserted that, in presence of few variables with medium / large sized effect, use lasso regression. In presence of many variables with small / medium sized effect, use ridge regression.

Conceptually, we can say, lasso regression (L1) does both variable selection and parameter shrinkage, whereas Ridge regression only does parameter shrinkage and end up including all the coefficients in the model. In presence of correlated variables, ridge regression might be the preferred choice. Also, ridge regression works best in situations where the least square estimates have higher variance. Therefore, it depends on our model objective.

Know more: Ridge and Lasso Regression

Q17. Rise in global average temperature led to decrease in number of pirates around the world. Does that mean that decrease in number of pirates caused the climate change?

Answer: After reading this question, you should have understood that this is a classic case of "causation and correlation". No, we can't conclude that decrease in number of pirates caused the climate change because there might be other factors (lurking or confounding variables) influencing this phenomenon.

Therefore, there might be a correlation between global average temperature and number of pirates, but based on this information we can't say that pirated died because of rise in global average temperature.

Know more: Causation and Correlation

Q18. While working on a data set, how do you select important variables? Explain your methods.

Answer: Following are the methods of variable selection you can use:

1. Remove the correlated variables prior to selecting important variables
2. Use linear regression and select variables based on p values
3. Use Forward Selection, Backward Selection, Stepwise Selection
4. Use Random Forest, Xgboost and plot variable importance chart
5. Use Lasso Regression
6. Measure information gain for the available set of features and select top n features accordingly.

Q19. What is the difference between covariance and correlation?

Answer:

- Correlation is the standardized form of covariance.
- Covariances are difficult to compare. For example: if we calculate the covariances of salary ($) and age (years), we'll get different covariances which can't be compared because of having unequal scales. To combat such situation, we calculate correlation to get a value between -1 and 1, irrespective of their respective scale.
- "Covariance" indicates the direction of the linear relationship between variables. "Correlation" on the other hand measures both the strength and direction of the linear relationship between two variables.
- Correlation is a function of the covariance
- **What sets them apart is the fact that correlation values are standardized whereas, covariance values are not.** You can obtain the correlation coefficient of two variables by dividing the

covariance of these variables by the product of the standard deviations of the same values and it essentially scales the value down to a limited range of **-1 to +1**.

- Vital tool for feature selection and multivariate analysis in data preprocessing and exploration. Correlation helps us investigate and establish relationships between variables. This is employed in feature selection before any kind of statistical modelling or data analysis.
- PCA or Principal Component Analysis is one significant application of the same. So how do we decide what to use? Correlation matrix or the covariance matrix? In simple words, you are advised to **use the covariance matrix when the variable are on similar scales and the correlation matrix when the scales of the variables differ.**

Q20. Is it possible capture the correlation between continuous and categorical variable? If yes, how?

Answer: Yes, we can use ANCOVA (analysis of covariance) technique to capture association between continuous and categorical variables.

Q21. Both being tree based algorithm, how is random forest different from Gradient boosting algorithm (GBM)?

Answer: The fundamental difference is, random forest uses bagging technique to make predictions. GBM uses boosting techniques to make predictions.

In bagging technique, a data set is divided into n samples using randomized sampling. Then, using a single learning algorithm a model is build on all samples. Later, the resultant predictions are combined using voting or averaging. Bagging is done is parallel. In boosting, after the first round of predictions, the algorithm weighs misclassified predictions higher, such that they can be corrected in the succeeding round. This sequential process of giving higher weights to misclassified predictions continue until a stopping criterion is reached.

Random forest improves model accuracy by reducing variance (mainly). The trees grown are uncorrelated to maximize the decrease in variance. On the other hand, GBM improves accuracy my reducing both bias and variance in a model.

Know more: Tree based modeling

Q22. Running a binary classification tree algorithm is the easy part. Do you know how does a tree splitting takes place i.e. how does the tree decide which variable to split at the root node and succeeding nodes?

Answer: A classification trees makes decision based on Gini Index and Node Entropy. In simple words, the tree algorithm find the best possible feature which can divide the data set into purest possible children nodes.

Gini index says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure. We can calculate Gini as following:

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2).
2. Calculate Gini for split using weighted Gini score of each node of that split

Entropy is the measure of impurity as given by (for binary class):

$$\text{Entropy} = -p \log_2 p - q \log_2 q$$

Here p and q is probability of success and failure respectively in that node. Entropy is zero when a node is homogeneous. It is maximum when a both the classes are present in a node at 50% – 50%. Lower entropy is desirable.

Q23. You've built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven't you trained your model perfectly?

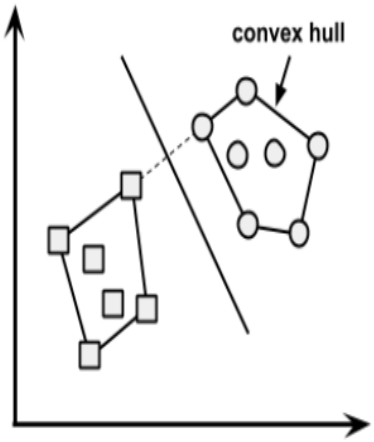
Answer: The model has overfitted. Training error 0.00 means the classifier has mimiced the training data patterns to an extent, that they are not available in the unseen data. Hence, when this classifier was run on unseen sample, it couldn't find those patterns and returned prediction with higher error. In random forest, it happens when we use larger number of trees than necessary. Hence, to avoid these situation, we should tune number of trees using cross validation.

Q24. You've got a data set to work having p (no. of variable) > n (no. of observation). Why is OLS as bad option to work with? Which techniques would be best to use? Why?

Answer: In such high dimensional data sets, we can't use classical regression techniques, since their assumptions tend to fail. When p > n, we can no longer calculate a unique least square coefficient estimate, the variances become infinite, so OLS cannot be used at all.

To combat this situation, we can use penalized regression methods like lasso, LARS, ridge which can shrink the coefficients to reduce variance. Precisely, ridge regression works best in situations where the least square estimates have higher variance.

Among other methods include subset regression, forward stepwise regression.



Q25. What is convex hull ? (Hint: Think SVM)

Answer: In case of linearly separable data, convex hull represents the outer boundaries of the two group of data points. Once convex hull is created, we get maximum margin hyperplane (MMH) as a perpendicular bisector between two convex hulls. MMH is the line which attempts to create greatest separation between two groups.

Q26. We know that one hot encoding increasing the dimensionality of a data set. But, label encoding doesn't. How ?

Answer: Don't get baffled at this question. It's a simple question asking the difference between the two.

Using one hot encoding, the dimensionality (a.k.a features) in a data set get increased because it creates a new variable for each level present in categorical variables. For example: let's say we have a variable 'color'. The variable has 3 levels namely Red, Blue and Green. One hot encoding 'color' variable will generate three new variables as Color.Red, Color.Blue and Color.Green containing 0 and 1 value.

In label encoding, the levels of a categorical variables gets encoded as 0 and 1, so no new variable is created. Label encoding is majorly used for binary variables.

Q27. What cross validation technique would you use on time series data set? Is it k-fold or LOOCV?

Answer: Neither.

In time series problem, k fold can be troublesome because there might be some pattern in year 4 or 5 which is not in year 3. Resampling the data set will separate these trends, and we might end up validation on past years, which is incorrect. Instead, we can use forward chaining strategy with 5 fold as shown below:

- fold 1 : training [1], test [2]
- fold 2 : training [1 2], test [3]
- fold 3 : training [1 2 3], test [4]
- fold 4 : training [1 2 3 4], test [5]
- fold 5 : training [1 2 3 4 5], test [6]

where 1,2,3,4,5,6 represents "year".

Q28. You are given a data set consisting of variables having more than 30% missing values? Let's say, out of 50 variables, 8 variables have missing values higher than 30%. How will you deal with them?

Answer: We can deal with them in the following ways:

1. Assign a unique category to missing values, who knows the missing values might decipher some trend
2. We can remove them blatantly.
3. Or, we can sensibly check their distribution with the target variable, and if found any pattern we'll keep those missing values and assign them a new category while removing others.

29. 'People who bought this, also bought…' recommendations seen on amazon is a result of which algorithm?

Answer: The basic idea for this kind of recommendation engine comes from collaborative filtering.

Collaborative Filtering algorithm considers "User Behavior" for recommending items. They exploit behavior of other users and items in terms of transaction history, ratings, selection and purchase information. Other users behaviour and preferences over the items are used to recommend items to the new users. In this case, features of the items are not known.

Know more: Recommender System

Q30. What do you understand by Type I vs Type II error ?

Answer: Type I error is committed when the null hypothesis is true and we reject it, also known as a 'False Positive'. Type II error is committed when the null hypothesis is false and we accept it, also known as 'False Negative'.

In the context of confusion matrix, we can say Type I error occurs when we classify a value as positive (1) when it is actually negative (0). Type II error occurs when we classify a value as negative (0) when it is actually positive(1).

Q31. You are working on a classification problem. For validation purposes, you've randomly sampled the training data set into train and validation. You are confident that your model will work incredibly well on unseen data since your validation accuracy is high. However, you get shocked after getting poor test accuracy. What went wrong?

Answer: In case of classification problem, we should always use stratified sampling instead of random sampling. A random sampling doesn't takes into consideration the proportion of target classes. On the contrary, stratified sampling helps to maintain the distribution of target variable in the resultant distributed samples also.

Q32. You have been asked to evaluate a regression model based on R^2, adjusted R^2 and tolerance. What will be your criteria?

Answer: Tolerance (1 / VIF) is used as an indicator of multicollinearity. It is an indicator of percent of variance in a predictor which cannot be accounted by other predictors. Large values of tolerance is desirable.

We will consider adjusted R^2 as opposed to R^2 to evaluate model fit because R^2 increases irrespective of improvement in prediction accuracy as we add more variables. But, adjusted R^2 would only increase if an additional variable improves the accuracy of model, otherwise stays same. It is difficult to commit a general threshold value for adjusted R^2 because it varies between data sets. For example: a gene mutation data set might result in lower adjusted R^2 and still provide fairly good predictions, as compared to a stock market data where lower adjusted R^2 implies that model is not good.

Q33. In k-means or kNN, we use euclidean distance to calculate the distance between nearest neighbors. Why not manhattan distance ?

Answer: We don't use manhattan distance because it calculates distance horizontally or vertically only. It has dimension restrictions. On the other hand, euclidean metric can be used in any space to calculate distance. Since, the data points can be present in any dimension, euclidean distance is a more viable option.

Example: Think of a chess board, the movement made by a bishop or a rook is calculated by manhattan distance because of their respective vertical & horizontal movements.

Q34. Explain machine learning to me like a 5 year old.

Answer: It's simple. It's just like how babies learn to walk. Every time they fall down, they learn (unconsciously) & realize that their legs should be straight and not in a bend position. The next time they fall down, they feel pain. They cry. But, they learn 'not to stand like that again'. In order to avoid that pain, they try harder. To succeed, they even seek support from the door or wall or anything near them, which helps them stand firm.

This is how a machine works & develops intuition from its environment.

Note: The interview is only trying to test if have the ability of explain complex concepts in simple terms.

Q35. I know that a linear regression model is generally evaluated using Adjusted R^2 or F value. How would you evaluate a logistic regression model?

Answer: We can use the following methods:

1. Since logistic regression is used to predict probabilities, we can use AUC-ROC curve along with confusion matrix to determine its performance.
2. Also, the analogous metric of adjusted R^2 in logistic regression is AIC. AIC is the measure of fit which penalizes model for the number of model coefficients. Therefore, we always prefer model with minimum AIC value.
3. Null Deviance indicates the response predicted by a model with nothing but an intercept. Lower the value, better the model. Residual deviance indicates the response predicted by a model on adding independent variables. Lower the value, better the model.

Know more: Logistic Regression

Q36. Considering the long list of machine learning algorithm, given a data set, how do you decide which one to use?

Answer: You should say, the choice of machine learning algorithm solely depends of the type of data. If you are given a data set which is exhibits linearity, then linear regression would be the best algorithm to use. If you given to work on images, audios, then neural network would help you to build a robust model.

If the data comprises of non linear interactions, then a boosting or bagging algorithm should be the choice. If the business requirement is to build a model which can be deployed, then we'll use regression or a decision tree model (easy to interpret and explain) instead of black box algorithms like SVM, GBM etc.

In short, there is no one master algorithm for all situations. We must be scrupulous enough to understand which algorithm to use.

Q37. Do you suggest that treating a categorical variable as continuous variable would result in a better predictive model?

Answer: For better predictions, categorical variable can be considered as a continuous variable only when the variable is ordinal in nature.

Q38. When does regularization becomes necessary in Machine Learning?

Answer: Regularization becomes necessary when the model begins to ovefit / underfit. This technique introduces a cost term for bringing in more features with the objective function. Hence, it tries to push the coefficients for many variables to zero and hence reduce cost term. This helps to reduce model complexity so that the model can become better at predicting (generalizing).

Q39. What do you understand by Bias Variance trade off?

Answer: The error emerging from any model can be broken down into three components mathematically. Following are these component :

$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\hat{f}\,(x) - E[\hat{f}\,(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Bias error is useful to quantify how much on an average are the predicted values different from the actual value. A high bias error means we have a under-performing model which keeps on missing important trends. **Variance** on the other side quantifies how are the prediction made on same observation different from each other. A high variance model will over-fit on your training population and perform badly on any observation beyond training.

Q40. OLS is to linear regression. Maximum likelihood is to logistic regression. Explain the statement.

Answer: OLS and Maximum likelihood are the methods used by the respective regression methods to approximate the unknown parameter (coefficient) value. In simple words,

Ordinary least square(OLS) is a method used in linear regression which approximates the parameters resulting in minimum distance between actual and predicted values. Maximum Likelihood helps in choosing the the values of parameters which maximizes the likelihood that the parameters are most likely to produce observed data.

If you are having 4GB RAM in your machine and you want to train your model on 10GB dataset. How would you go about this problem. Have you ever faced this kind of problem in your machine learning/data science experience so far ?

First of all you have to ask which ML model you want to train.

For Neural networks: Batch size with Numpy array will work.

Steps:

1. Load the whole data in Numpy array. Numpy array has property to create mapping of complete dataset, it doesn't load complete dataset in memory.
2. You can pass index to Numpy array to get required data.
3. Use this data to pass to Neural network.
4. Have small batch size.

For SVM: Partial fit will work

Steps:

1. Divide one big dataset in small size datasets.
2. Use partialfit method of SVM, it requires subset of complete dataset.
3. Repeat step 2 for other subsets

Q41. How do I start if I want to build a model to predict if a tweet will go viral?

Answer: https://www.quora.com/How-do-I-start-if-I-want-to-build-a-model-to-predict-if-a-tweet-will-go-viral

http://www.cs.cornell.edu/home/kleinber/www14-cascades.pdf

Q42. How does LinkedIn's People You May Know work?

Answer: Linkedin People You May Know product is a recommender system (i.e Connection suggestion algorithm) that uses many features to calculate a connection probability among two people.

Some examples are : companies position overlap (How many months/years both individuals work together at same company, similar/related field) , school education overlap, strong unconnected people in linkedin connection graph. (both individual have many common friends) etc.

Here is our list of 12 signs the company you are interviewing with for a data scientist job should be avoided

Red flags on how the data science team runs

1. No data engineering or infrastructure.

Data science requires data to be easily available for analysis. If the company doesn't have a well-maintained data infrastructure, you won't have what you need to do your job. A data engineer is a person who prepares data for analysis, and if your company doesn't have them you'll have to do the work yourself. If you feel qualified to take on the role of a data engineer that may be okay, but otherwise you'll be struggling to deliver anything of value.

Question to ask during the interview: what is your data infrastructure like and who maintains it? What format is the data typically in (Excel, a SQL database, csv)?

2. No peer review between data scientists.

A strong data science team will have ways to ensure mistakes don't slip through the cracks. These can include code reviews, practice presentations, and consistent check-ins with the team. If the team doesn't consistently do these, mistakes won't be found until the work is already delivered, which usually ends with someone getting reprimanded.

Question to ask: what steps does the team take for QA and peer review?

3. No standard set of languages on the team.

Many data science teams take the approach of letting anyone on the team use any language they want. The idea is that if everyone uses their favorite languages work will be completed faster. There is a huge problem with this: when everyone uses separate languages, no one will be able to pass off work to anyone else. Every data science task will have a single person responsible for it, and if they quit, get sick, or just need help no one will be able to do so, creating a very stressful environment. It's fine to use R, Python, or even dare we say SAS, but just have a consistent set of languages amongst the team.

Question to ask: what languages does your team use, and how do you decide whether to adopt a new one?

4. They don't understand the data hierarchy of needs.

Similar to not having a data infrastructure, some companies get really excited about concepts like AI without having the foundation in place. Machine learning and AI require a company to have a high level of data science maturity, including understanding how to build models, their limitations, and how to deploy them. You might get blamed when their unrealistic expectations meet reality.

Question to ask: how does the company balance spending time on complex approaches like AI with foundational work like cleaning data, checking data quality, and adding logging?

5. No version control.

Mature data science teams use git to keep track of changes to analyses and code. Other teams instead use methods like shared network folders, which don't let you see when things changed, why they are changed, or previous versions. Occasionally teams don't share code at all and work just lives on the data scientists individual laptops. Avoid these last groups like the plague. Not having methods of sharing code means the team can't work together.

Question to ask: how to you share code amongst the team? Is all code shared or just some of it?

6. No clear delineation between people who run reports vs do analyses.

The skillsets required to create and maintain reports, to build data science models, and to put machine learning models into production are all different. If the company doesn't have a clear way of determining who does what work, you could start your job and end up doing work totally different than what you expected. You don't want to walk in on your first day expecting to build a time-series forecast and find out your job is to refresh the monthly sales Excel spreadsheet.

Question to ask: how are reporting, analysis, and production-model building tasks split?

Questions You Should Ask

The interview process isn't all about *answering* questions, it's also about *asking* them. Most interviewers will end each interview by giving you an opportunity to ask questions, and you should not pass it up. This is a valuable chance for you to learn more about the company and to further impress the person you're speaking with.

Ask to Impress

Most of the recruiters and hiring managers we spoke with for this guide agreed that their impression of a candidate was influenced by the questions they asked, and that asking the right questions could help a candidate. Among the types of questions they recommend asking:

Detailed questions about doing the job: The key here is *detailed*; don't just ask what you'll be doing day to day. Ask specific questions that show you're already thinking about how you would function most effectively in the position, like questions about specific tools or workflows you'll be using or specific business questions you'll be addressing. For example, you might ask about a company's flexibility towards using a tool or package/library that isn't included in their tech stack, but that you think might be helpful in the role you're applying for.

"I'll get questions like, 'oh have you tried this? Do you work with this type of data? I'd imagine you do this or that,'" says G2 Crowd Data Science and Analytics Manager Michael Hupp. "And we haven't gone down that road, but just the fact that they're thinking about that really stands out."

Detailed questions about the business problems you'll be solving in the position: As above, the idea behind asking this kind of question is to demonstrate that you're already mentally engaged with solving *this* company's business problems.

"The questions that really stick out as good are ones that demonstrate that they have done their homework and are interested in Kitware specifically," says Kitware HR Director Jeff Hall.

Outlier.ai CTO Mike Kim agrees: "The questions I can think of that stand out as, 'Wow, these were impressive questions,' are the ones that really demonstrated that the candidate understood our problem, what they were working on, what we are working on, and how they fit into that."

This is a sentiment that was echoed by many of the other recruiters and hiring managers we spoke with for this guide.

This technique is particularly helpful when interviewing at smaller companies, Edouard Harris says. "Almost always the best question to ask is: 'What's the biggest bottleneck for you at the moment? What is the one thing that's blocking you as a company the most?' What that question does is tell the interviewer: this person, right out of the gate, wants to know how they can be as helpful as possible."

That particular question is also a valuable information-gathering tool, Edouard says. "If the company is 20 people or less, virtually everyone in the company should be able to answer that question. If not, there's some kind of communication problem, which is itself a red flag."

Questions about growth opportunities and training: These questions demonstrate that you're interested in continually improving your skills and learning, which is something most employers want to see. (And of course, it's also valuable information for you to have later when you're assessing offers; a company with a lower salary offer could still be the better choice if it can also offer great training opportunities that'll be better for your career in the long term).

Questions about collaboration with other departments: Data science teams typically have to work in collaboration with a lot of other departments. Questions along these lines show you're interested in that aspect of the position, and the answer will probably give you some idea of what the company's culture is like, and how efficient the collaborative workflow is likely to be.

Questions about long-term plans and projects you'll be working on: "Those are the questions that I look for," says CiBo Technologies Talent Acquisition Manager Jamieson Vazquez, "folks that want to know what the long-term future is, want to know where we are building but want to know how they can really impact those future plans too."

What Not *to Ask*

Asking no questions at all: This demonstrates to an interviewer that you're not engaged at all. You should always have at least a few questions ready.

Asking about compensation, paid time off, etc.: The appropriate time for these kinds of negotiations is at the end of the interview process, after you've received a job offer. If you ask about this before then, especially if you ask about it *repeatedly*, interviewers will get the impression that you're just in it for the paycheck and not genuinely interested in the work.

Asking questions with easy-to-find answers: If you ask about something that's clearly answered on the company's website, for example, it just shows the interviewer that you haven't

bothered to do your research prior to the interview, and *that* suggests you don't really care about the job.