Parag reference

Tuesday, September 4, 2018 1:38 PM

Interview Questions

Monday, August 20, 2018

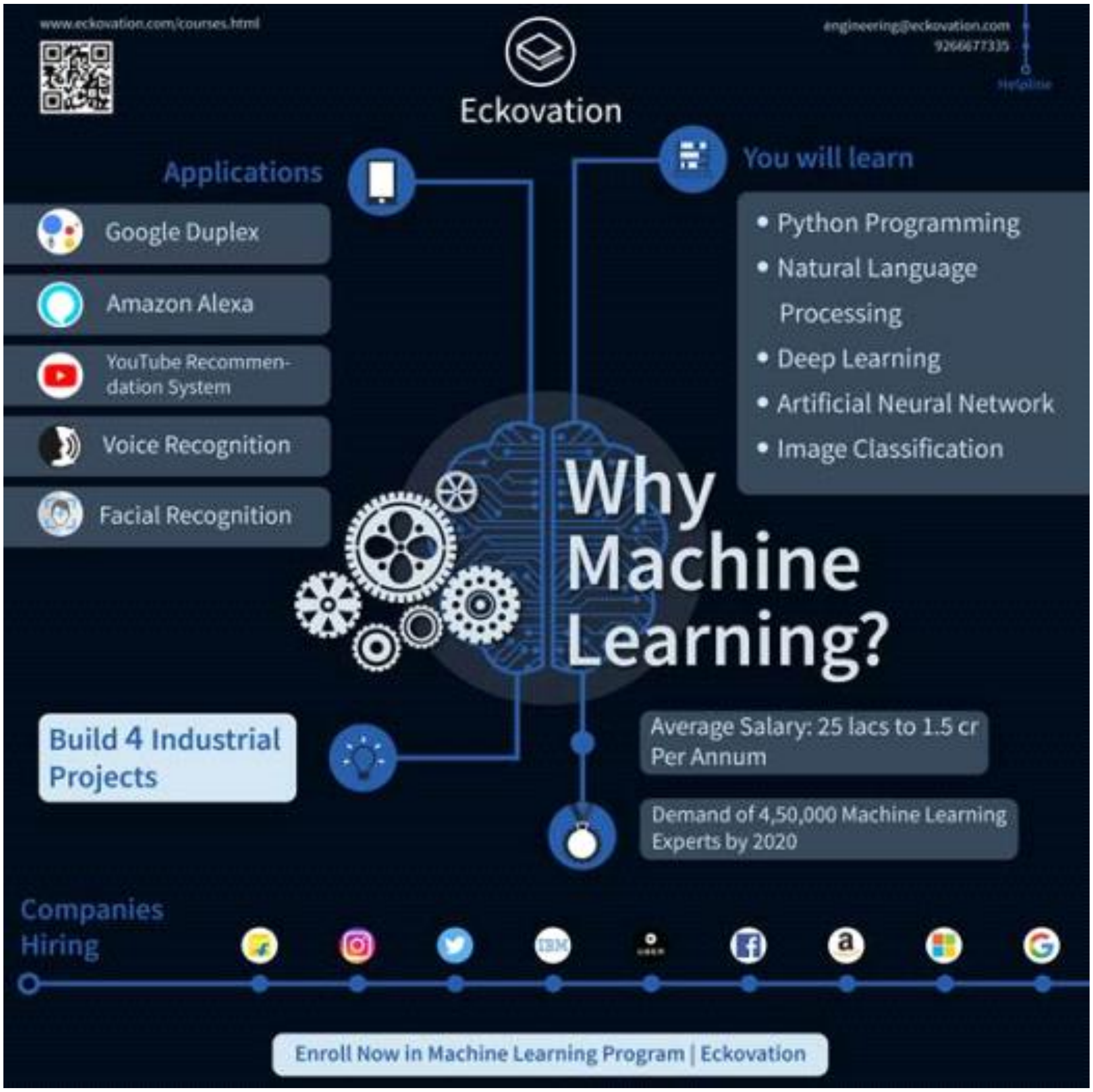
2:12 PM

51 Job Interview Related Questions of Machine Learning (ML) and Artificial Intelligence (AI)

June 28, 2018Engineering Guru

Machine learning and Artificial Intelligence are being looked as the drivers of the next industrial revolution happening in the world today. This also means that there are numerous exciting startups looking for data scientists. What could be a better start for your aspiring career!

Scope of Machine Learning and Artificial Intelligence



www.eckovation.com/courses.html

engineering@eckovation.com
9266677335
Helpline

Eckovation

Applications
- Google Duplex
- Amazon Alexa
- YouTube Recommendation System
- Voice Recognition
- Facial Recognition

You will learn
- Python Programming
- Natural Language Processing
- Deep Learning
- Artificial Neural Network
- Image Classification

Why Machine Learning?

Build 4 Industrial Projects

Average Salary: 25 lacs to 1.5 cr Per Annum

Demand of 4,50,000 Machine Learning Experts by 2020

Companies Hiring

Enroll Now in Machine Learning Program | Eckovation

However, still, getting into these roles is not easy. You obviously need to get excited about the idea, team and the vision of the company. You might also find some real difficult techincal questions on your way. The set of questions asked depend on what does the company do. Do they provide consulting? Do they build ML products ? You should always find this out prior to beginning your interview preparation.

Job in Machine Learning and Artificial Intelligence

If you want to land a job in AI and ML, you'll need to pass a rigorous and competitive interview process. In fact, most top companies will have at least 3 rounds of interviews. **During the process, you'll be tested for a variety of skills, including:**

- Your technical and programming skills
- Your ability to structure solutions to open-ended problems
- Your ability to apply machine learning effectively
- Your ability to analyze data with a range of methods
- Your communication skills, cultural fit, etc.
- And your mastery of key concepts in data science and machine learning

51 Interview Questions of Machine Learning and Artificial Intelligence

To help you prepare for your next interview, We have prepared a list of 51 plausible & tricky questions which are likely to come across your way in interviews. If you can answer and understand these question, rest assured, you will give a tough fight in your job interview.

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:

- 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.
- 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.
- 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.
- Also, we can use PCA and pick the components which can explain the maximum variance in the data set.
- Using online learning algorithms like Vowpal Wabbit (available in Python) is a possible option.
- Building a linear model using Stochastic Gradient Descent is also helpful.
- 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.

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:

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

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.

Join Eckovation's Machine Learning and Artificial Intelligence Program by Industrial Experts: **Click Here**

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:

- There exist a pattern.
- You cannot solve it mathematically (even by writing exponential equations).
- 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:

- Use regularization technique, where higher model coefficients get penalized, hence lowering model complexity.
- 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.

Join Machine Learning and Artificial Intelligence Program by Industrial Experts: **Click Here**

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).

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 (ymean), 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.

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.

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.

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:

- Remove the correlated variables prior to selecting important variables
- Use linear regression and select variables based on p values
- Use Forward Selection, Backward Selection, Stepwise Selection
- Use Random Forest, Xgboost and plot variable importance chart
- Use Lasso Regression
- 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.

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.

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:

- Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^

2+q^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.

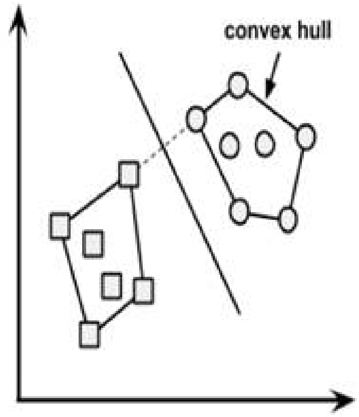
Q23. You've built a random forest model with 10000 trees. You got delighted after getting training error as 0.00. But, the validation error is 34.23. What is going on? Haven't you trained your model perfectly?

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:

- Assign a unique category to missing values, who knows the missing values might decipher some trend
- We can remove them blatantly.
- 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.

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 R², adjusted R² 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 R² as opposed to R² to evaluate model fit because R² increases irrespective of improvement in prediction accuracy as we add more variables. But, adjusted R² 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 R² because it varies between data sets. For example: a gene mutation data set might result in lower adjusted R² and still provide fairly good predictions, as compared to a stock market data where lower adjusted R² 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 R² or F value. How would you evaluate a logistic regression model?
Answer: We can use the following methods:

- Since logistic regression is used to predict probabilities, we can use AUC-ROC curve along with confusion matrix to determine its performance.
- Also, the analogous metric of adjusted R² 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.
- 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.

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 :

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

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

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.
41. What are parametric models? Give an example.
Parametric models are those with a finite number of parameters. To predict new data, you only need to know the parameters of the model. Examples include linear regression, logistic regression, and linear SVMs.

Non-parametric models are those with an unbounded number of parameters, allowing for more flexibility. To predict new data, you need to know the parameters of the model and the state of the data that has been observed. Examples include decision trees, k-nearest neighbors, and topic models using latent dirichlet analysis.

42. What is the “Curse of Dimensionality?”

The difficulty of searching through a solution space becomes much harder as you have more features (dimensions).

Consider the analogy of looking for a penny in a line vs. a field vs. a building. The more dimensions you have, the higher volume of data you’ll need.

Join Machine Learning and Artificial Intelligence Program by Industrial Experts: **Click Here**

43. What is the difference between stochastic gradient descent (SGD) and gradient descent (GD)?

Both algorithms are methods for finding a set of parameters that minimize a loss function by evaluating parameters against data and then making adjustments.

In standard gradient descent, you’ll evaluate all training samples for each set of parameters. This is akin to taking big, slow steps toward the solution.

In stochastic gradient descent, you’ll evaluate only 1 training sample for the set of parameters before updating them. This is akin to taking small, quick steps toward the solution.

44. What is the Box-Cox transformation used for?

The Box-Cox transformation is a generalized “power transformation” that transforms data to make the distribution more normal.

For example, when its lambda parameter is 0, it’s equivalent to the log-transformation.

It’s used to stabilize the variance (eliminate heteroskedasticity) and normalize the distribution.

45. What are 3 data preprocessing techniques to handle outliers?

- Winsorize (cap at threshold).
- Transform to reduce skew (using Box-Cox or similar).
- Remove outliers if you’re certain they are anomalies or measurement errors.

46. How much data should you allocate for your training, validation, and test sets?

You have to find a balance, and there’s no right answer for every problem.

If your test set is too small, you’ll have an unreliable estimation of model performance (performance statistic will have high variance). If your training set is too small, your actual model parameters will have high variance.

A good rule of thumb is to use an 80/20 train/test split. Then, your train set can be further split into train/validation or into partitions for cross-validation.

47. What are the advantages and disadvantages of decision trees?

Advantages: Decision trees are easy to interpret, nonparametric (which means they are robust to outliers), and there are relatively few parameters to tune.

Disadvantages: Decision trees are prone to be overfit. However, this can be addressed by ensemble methods like random forests or boosted trees.

48. What are the advantages and disadvantages of neural networks?

Advantages: Neural networks (specifically deep NNs) have led to performance breakthroughs for unstructured datasets such as images, audio, and video. Their incredible flexibility allows them to learn patterns that no other ML algorithm can learn.

Disadvantages: However, they require a large amount of training data to converge. It’s also difficult to pick the right architecture, and the internal “hidden” layers are incomprehensible.

49. Explain Latent Dirichlet Allocation (LDA).

Latent Dirichlet Allocation (LDA) is a common method of topic modeling, or classifying documents by subject matter.

LDA is a generative model that represents documents as a mixture of topics that each have their own probability distribution of possible words.

The "Dirichlet" distribution is simply a distribution of distributions. In LDA, documents are distributions of topics that are distributions of words.

50. What are some key business metrics for (S-a-a-S startup | Retail bank | e-Commerce site)?

Thinking about key business metrics, often shortened as KPI's (Key Performance Indicators), is an essential part of a data scientist's job. Here are a few examples, but you should practice brainstorming your own.

Tip: When in doubt, start with the easier question of "how does this business make money?"

- S-a-a-S startup: Customer lifetime value, new accounts, account lifetime, churn rate, usage rate, social share rate
- Retail bank: Offline leads, online leads, new accounts (segmented by account type), risk factors, product affinities
- e-Commerce: Product sales, average cart value, cart abandonment rate, email leads, conversion rate

51. How can you help our marketing team be more efficient?

The answer will depend on the type of company. Here are some examples.

- Clustering algorithms to build custom customer segments for each type of marketing campaign.
- Natural language processing for headlines to predict performance before running ad spend.
- Predict conversion probability based on a user's website behavior in order to create better re-targeting campaigns.

From <https://engineering.eckovation.com/51-job-interview-related-questions-machine-learning-ml-artificial-intelligence-ai/>

ML Algorithms

01 April 2018

19:40

Decision Tree -

Pros -

1. Creates branches for arriving at leaves (Values/predictions).
2. Can work on both Numeric as well as Non Numeric data.
3. Can be used for both classification and regression. (Supervised learning method).
4. Require little effort in terms of data preparation.
5. Decision criteria is determined in a different way for classification and regression.

Cons -

1. Can easily overfit data. This will not generalize well.
2. Can become unstable for even a small change in data. This is Variance issue. This needs to be controlled using bagging and boosting.
3. Greedy algorithms can not gurantee to return globally optimal deicion tree.

Pruning is required for preventing overfitting.

Following are most commonly used algorithms in decision trees-

1. GIRI Index
2. CHI Square
3. Information gain
4. Reduction in Variance

Random Forest -

Pros -

a. We generate multiple trees.
b. Can be used for both classification and regression. (Supervised learning method).
c. Will handle missing values and maintain accuracy when a large proportion of data is missing.
d. Wont over fit the model.
e. Can handle large dataset with higher dimensionality.
f. Ensemble machine learning algorithms.

Cons -

a. RF algorithm is quite suitable for classification but not as good for regression.
b. You have very little control over what the model does.

Applications -

a. Banking sector for identifying loyal and non-loyal customers.
b. Medical sector for identifying diagnosis of customers.
c. Stock market for identifying profit or loss from a given stock.
d. In retail for recommendation engine for recommending products.
e. Image classification in computer vision. MS used for XBOX.

Bagging - Is a machine learning ensemble meta algorithm designed to improve the stability, reduce variance and accuracy.
Boosting - Is a machine learning ensemble meta algorithm for reducing Bias and variance in supervised learning.

XGBoost -

1. XGBoost is the leading model for working with standard tabular data (the type of data you store in Pandas DataFrames, as opposed to more Exotic types of data like images and videos).
2. XGBoost models dominate many Kaggle competitions.
3. To reach peak accuracy, XGBoost models require more knowledge and model tuning than techniques like Random Forest.
4. XGBoost is an implementation of the Gradient Boosted Decision Trees algorithm.
Naive Model -> Calculate Erros -> Build Model, Predictions Errors -> Add last model to ensemble.

5. XGBoost has a few parameters that can dramatically affect your model's accuracy and training speed.
n_estimators - specifies how many times to go through the modeling cycle. [100-1000]
early_stopping_rounds - Early stopping causes the model to stop iterating when the validation score stops improving, even if we aren't at the hard stop for n_estimators. [5-10]
6. In general, a small learning rate (and large number of estimators) will yield more accurate XGBoost models, though it will also take the model longer to train since it does more iterations through the cycle.

ML Model Evaluation Metrics -

31 March 2018
18:03

1. Regression model (continuous output) -

- **Mean Absolute Error**

1. The Mean Absolute Error (or MAE) is the sum of the absolute differences between predictions and actual values.
2. It gives an idea of how wrong the predictions were.

3. The measure gives an idea of the magnitude of the error, but no idea of the direction.
4. A value of 0 indicates no error or perfect predictions.

- **Mean Squared Error -**

  The Mean Squared Error (or MSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of error.

- **Root Mean Squared Error (RMSE) -**

  1. RMSE is the most popular evaluation metric used in regression problems.
  2. It follows an assumption that error are unbiased and follow a normal distribution. Here are the key points to consider on RMSE:
  3. The power of 'square root' empowers this metric to show large number deviations.
  4. The 'squared' nature of this metric helps to deliver more robust results which prevents cancelling the 5. positive and negative error values. In other words, this metric aptly displays the plausible magnitude of error term.
  6. It avoids the use of absolute error values which is highly undesirable in mathematical calculations.
  When we have more samples, reconstructing the error distribution using RMSE is considered to be more reliable.
  7. As compared to mean absolute error, RMSE gives higher weightage and punishes large errors.
  8. Limitations - RMSE is highly affected by outlier values. Hence, make sure you've removed outliers from your data set prior to using this metric.

- **R^2 Metric**

  a. The R^2 (or R Squared) metric provides an indication of the goodness of fit of a set of predictions to the actual values. In statistical literature, this measure is called the coefficient of determination.
  b. This is a value between 0 and 1 for no-fit and perfect fit respectively.
  c. The higher the value, the better.
  d. 0 to 0.5 values of R^2 is considered poor.
  e. R2 = 1- SSE/TSS

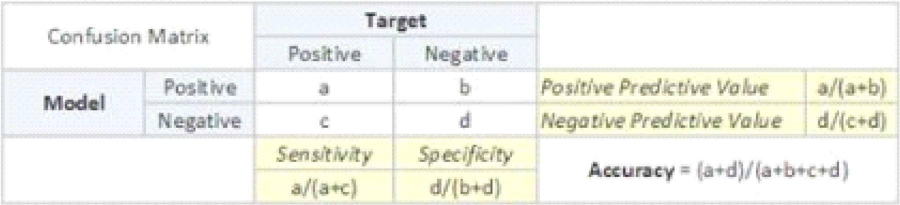
- **Explained variance score -**

  a. explained variance score = 1 - var(y_hat - y_true) / var(y_true), where the var is biased variance, i.e. var(y_hat - y_true) = sum(error^2 - mean(error))/n. Compared with R^2, the only difference is from the mean(error). if mean(error)=0, then R2 = explained variance score.

2. Classification model (nominal or binary output) -
   - Class output : Algorithms like SVM and KNN create a class output.
   - Probability output : Algorithms like Logistic Regression, Random Forest, Gradient Boosting, Adaboost etc. give probability outputs.

- **A confusion matrix** is an N X N matrix, where N is the number of classes being predicted. For the problem in hand, we have N=2, and hence we get a 2 X 2 matrix. Here are a few definitions, you need to remember for a confusion matrix :

  Accuracy : the proportion of the total number of predictions that were correct.
  Positive Predictive Value or Precision : the proportion of positive cases that were correctly identified.
  Negative Predictive Value : the proportion of negative cases that were correctly identified.
  Sensitivity or Recall : the proportion of actual positive cases which are correctly identified.
  Specificity : the proportion of actual negative cases which are correctly identified.



Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

- **Gain and Lift charts -**

  Any model with lift @ decile above 100% till minimum 3rd decile and maximum 7th decile is a good model.
  Else you might consider over sampling first.
  Lift / Gain charts are widely used in campaign targeting problems.

- **K-S or Kolmogorov-Smirnov chart** measures performance of classification models.

  The K-S would be 0. In most classification models the K-S will fall between 0 and 100, and that the higher the value the better the model is at separating the positive from negative cases.

- **ROC (Receiver operating characteristic) curve.**

  The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders.
  The ROC curve is the plot between sensitivity and (1- specificity).
  i.e. Curve between a/(a+c) and b/(b+d).
  (1- specificity) is also known as false positive rate and sensitivity is also known as True Positive rate. Following is the ROC curve for the case in hand.
  0.9-1 = excellent (A)
  0.8-.9 = good (B)
  0.7-.8 = fair (C)
  0.6-.7 = poor (D)
  0.5-.6 = fail (F)

- **Gini Coefficient -**
  a. Gini coefficient can be straigh away derived from the AUC ROC number. Gini is nothing but ratio between area between the ROC curve and the diagnol line & the area of the above triangle. Following is the formulae used :
  b. Gini = 2*AUC – 1
  c. Gini above 60% is a good model. For the case in hand we get Gini as 92.7%.

- **Concordant – Discordant ratio -**

  This is one of the most important metric for any classification predictions problem.
  Concordant ratio of more than 60% is considered to be a good model.
  This metric generally is not used when deciding how many customer to target etc.
  It is primarily used to access the model's predictive power. For decisions like how many to target are again taken by KS / Lift charts.

- **Logarithmic Loss -**

  1. Logarithmic loss (or log-loss) is a performance metric for evaluating the predictions of probabilities of membership to a given class.
  2. The scalar probability between 0 and 1 can be seen as a measure of confidence for a prediction by an algorithm. Predictions that are correct or incorrect are rewarded or punished proportionally to the confidence of the prediction.

- **Cross Validation and Learning Curves:**

  1. If you plot cross-validation (cv) error and training set error rates versus training set size, you can learn a lot. If the two curves approach each other with low error rate, then you are doing well.
  2. If it looks like the curves are starting to approach each other and both heading/staying low, then you need more data.

3. If the cv curve remains high, but the training set curve remains low, then you have a high-variance situation. You can either get more data, or use regularization to improve generalization.
4. If the cv stays high and the training set curve comes up to meet it, then you have high bias. In this case, you want to add detail to your model.

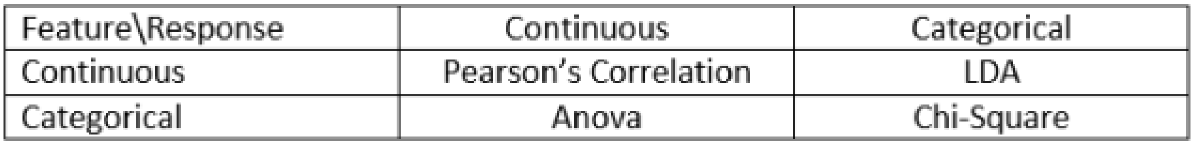
ML Feature Engineering Methods-

31 March 2018
13:22

https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/
https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/

Methods -

1. Filter Methods -
    - They are normally used as preprocessing step.
    - selection of features is independent of any machine learning algorithms
    - features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable.
    - filter methods do not remove multi-collinearity.



Feature\Response	Continuous	Categorical
Continuous	Pearson's Correlation	LDA
Categorical	Anova	Chi-Square

- **Pearson's Correlation:** It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson's correlation is given as:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- **LDA:** Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.
- **ANOVA:** ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.
- **Chi-Square:** It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.

    One thing that should be kept in mind is that filter methods do not remove multicollinearity. So, you must deal with multicollinearity of features as well before training models for your data.

2. Wrapper Methods -
    - In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset.
    - These methods are usually computationally very expensive.
    - **Forward Selection :** Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.
    - **Backward Elimination :** In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

- **Recursive Feature elimination** : It is a greedy optimization algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.

3. Embedded Methods -
   - Embedded methods combine the qualities' of filter and wrapper methods.
   - It's implemented by algorithms that have their own built-in feature selection methods.
   - Eg. LASSO and RIDGE regression, Regularized trees, Memetic algorithm, Random multinomial logit.

   The main differences between the filter and wrapper methods for feature selection are:

- Filter methods measure the relevance of features by their correlation with dependent variable while wrapper methods measure the usefulness of a subset of feature by actually training a model on it.
- Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.
- Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.
- Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features.
- Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

ML Statistical and visualization techniques

Monday, August 20, 2018
11:40 AM

https://www.analyticsvidhya.com/blog/2015/05/data-visualization-python/

Visualization techniques - Types of Graphs you can draw.

Statistical methods -

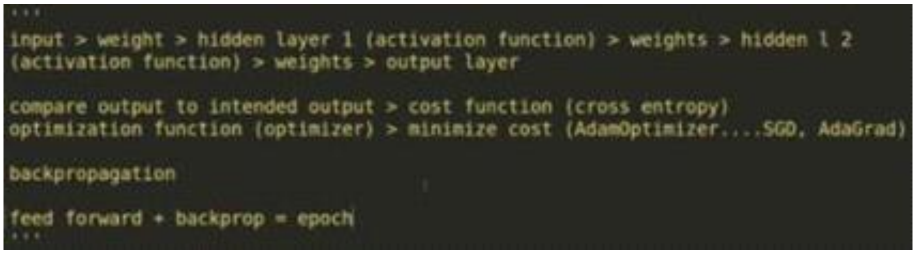
Ml Gradient Decent Algorithms -

Monday, August 20, 2018
11:42 AM

1. Gradient descent - Uses full data set for each iteration.
2. Stochastic gradient descent optimizer - Updates are made for each training example.
3. Mini Batch Gradient Descent - Combination of Gradient descent and Stochastic gradient descent optimizer. Instead of using full data, uses mini batches.
4. Momentum - Amplifies the movement in right direction. It also dampens oscillations. This results in faster convergence and reduced oscillations. This has a con though. When it reaches minimum, this may overshoot due to movement.
5. Adagrad - Adaptive gradient descent. Uses a different learning rate for every parameter. Makes bigger updates for infrequent parameters and smaller one for frequent. Learning rate becomes very slow. This is called learning rate decay.
6. AdaDelta - Instead of using only last steps learning gradient, all previous step gradient is used.
7. Adam - Adaptive moment estimation. Since we are learning rate for each parameter separately Adam stores momentum changes each of them separately.
8. ~~RMSprop - It is recommended to leave the parameters of this optimizer at their default values~~

~~(except the learning rate, which can be freely tuned).~~

9. ~~This optimizer is usually a good choice for recurrent neural networks.~~
10. ~~Adamax -~~
11. ~~TFOptimizer - Wrapper class for native TensorFlow optimizers~~



```
'''
input > weight > hidden layer 1 (activation function) > weights > hidden l 2
(activation function) > weights > output layer

compare output to intended output > cost function (cross entropy)
optimization function (optimizer) > minimize cost (AdamOptimizer....SGD, AdaGrad)

backpropagation

feed forward + backprop = epoch
'''
```

ML Using Categorical Data as features-

02 April 2018
17:17

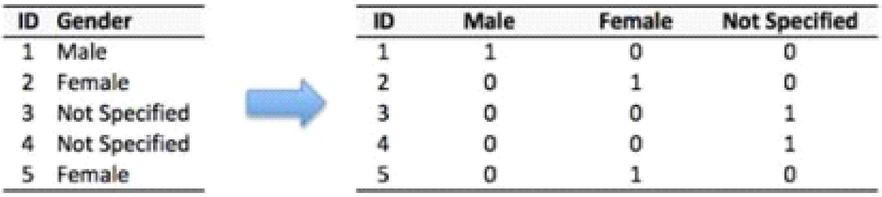
sklearn.preprocessing import OneHotEncoder
pandas.get_dummies

How to convert categorical data in numeric for machine learning algorithm to consume -

Method 1: Encoding to ordinal variables

Original Data City	Encoding to ordinal variables
New York	1
New Jersey	2
Tehran	3
New York	1

Method 2: One hot encoding (or dummy variables) :



ID	Gender
1	Male
2	Female
3	Not Specified
4	Not Specified
5	Female

ID	Male	Female	Not Specified
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	1
5	0	1	0

OHE has some significant shortcomings:
1. OHE representation produces very high dimensionality, this causes an increase in the model's training and serving time and memory consumption.
2. OHE can easily cause a model to overfit the data.
3. OHE can't handle categories that weren't in the training data (like new URLs, new device types etc), this can be problematic in domains that change all the time.

Method 3: Feature hashing (a.k.a the hashing trick)
In feature hashing we apply a hashing function to the category and then represent it by its indices. for example, if we choose a dimension of 5 to represent "New York" we will calculate H(New York) mod 5 = 3 (for example) so New York representation will be (0,0,1,0,0).

Feature hashing has some major pros: It is low dimensional thus it is very efficient in processing time and memory, it can be computed with online learning because as opposed to one hot encoding we don't need to go over all the data and build a dictionary of all possible categories and their mapping and it is not affected by new kinds of categories.

But it also have some cons. As we know, hashing functions sometimes have collision so if H(New York) = H(Tehran) the model can't know what city were in the data. There are some sophisticated hashing function that try to reduce the number of collision but anyway, studies have shown that collisions usually doesn't affect significantly on the models performance. Second shortcoming is that hashed features are not interpretable so doing things like feature importance and model debugging is very hard.

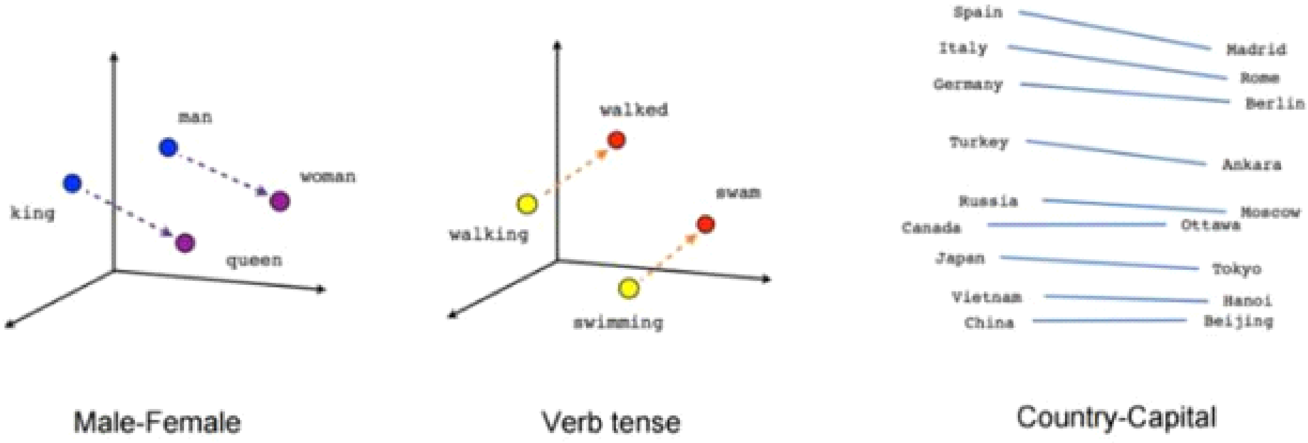
Method 4: Encoding categories with dataset statistics

we will try to give our models a numeric representation for every category with a small number of columns but with an encoding that will put similar categories close to each other.

The easiest way to do it is replace every category with the number of times that we saw it in the dataset. This way if New York and New Jersey are both big cities, they will probably both appear many times in our dataset and the model will know that they are similar.

Method 5: Cat2Vec

Problems in NLP usually have the same issues that categorical data has and Bag of words (A very common text representation) is pretty much the same as one hot encoding. NLP researchers has developed a very cool method called Word2Vec (or word embedding) in order to deal with the problems of bag of words. In Word2Vec we represent each word as a vector in a way that terms that are semantically close will be close in the vector space, Moreover, we can apply linear arithmetic for example in Word2vec space King-Man+Women= Queen.



Male-Female

Verb tense

Country-Capital

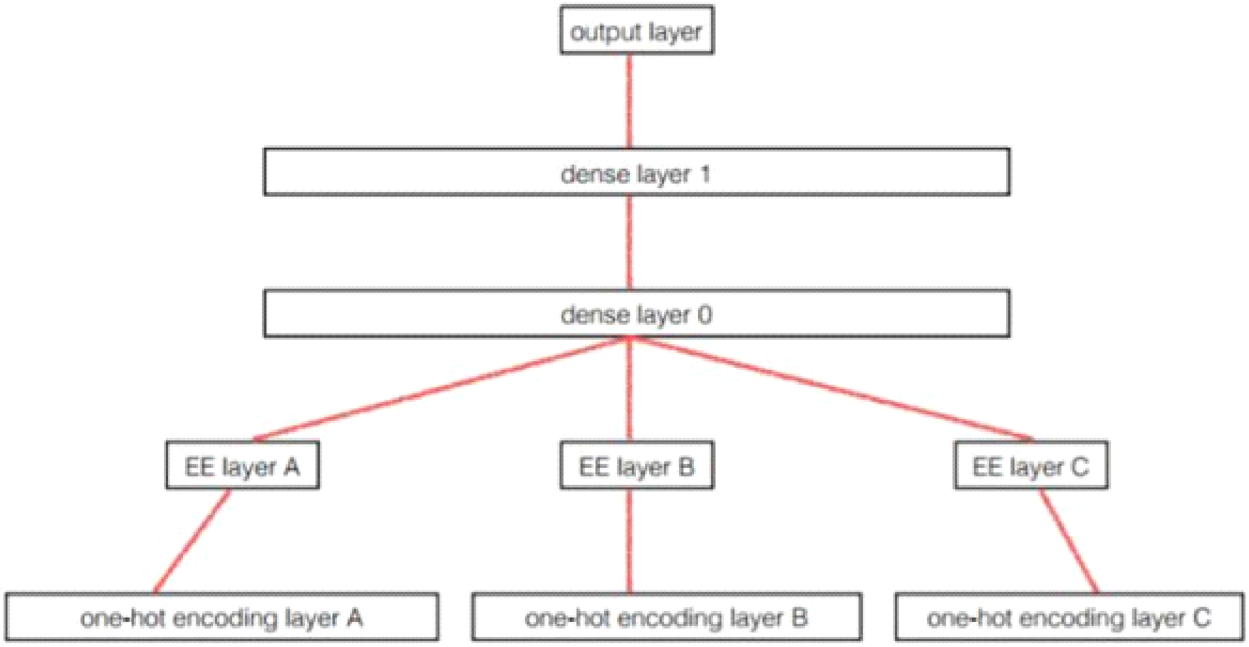
The Cat2vec has some weird behavior, if we plot the categories we will find that for example the most close category to New York is Cnn.com which is weird . This pitfall in our method is caused by the way that we train the Cat2Vec that assumes that all categories are similar like words that are all similar.

Method 6: Category embedding with deep learning

In our last method we will deal with the pitfall of Cat2vec and separately build a vector embedding to every category type. We will do it with an embedding layer in a deep neural network.

Embedding layers are used to convert one hot encoded variables into vector representation but as opposed to Word2vec the layers mission is not creating a semantical embedding but it is creating an embedding that will help our prediction goal.

We will use the DNN (Dense neural network?) described in the paper Entity Embeddings of Categorical Variables. After training the DNN we will use it's category embedding as input for the LR and RF.



Dnn architecture from Entity Embeddings of Categorical Variables. The DNN is composed of a one hot encoded input layer followed by an embedding layer for each categorical feature followed by two fully connected layers and a softmax layer.

ML Hyper Parameter Tuning -

Thursday, August 23, 2018
9:38 AM

Approaches

1. **Grid search**

   The traditional way of performing hyperparameter optimization has been *grid search*, or a *parameter sweep*, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set[3] or evaluation on a held-out validation set.[4]

   Since the parameter space of a machine learner may include real-valued or unbounded value spaces for certain parameters, manually set bounds and discretization may be necessary before applying grid search.

   For example, a typical soft-margin SVM classifier equipped with an RBF kernel has at least two hyperparameters that need to be tuned for good performance on unseen data: a regularization constant *C* and a kernel hyperparameter γ. Both parameters are continuous, so to perform grid search, one selects a finite set of "reasonable" values for each, say

   Grid search then trains an SVM with each pair (*C*, γ) in the Cartesian product of these two sets and evaluates their performance on a held-out validation set (or by internal cross-validation on the training set, in which case multiple SVMs are trained per pair). Finally, the grid search algorithm outputs the settings that achieved the highest score in the validation procedure.

   Grid search suffers from the curse of dimensionality, but is often embarrassingly parallel because typically the hyperparameter settings it evaluates are independent of each other.[2]

2. **Random search**

   Main article: Random search

   Random Search replaces the exhaustive enumeration of all combinations by selecting them randomly. This can be simply applied to the discrete setting described above, but also generalizes to continuous and mixed spaces. It can outperform Grid search, especially when only a small number of hyperparameters affects the final performance of the machine learning algorithm[2]. In this case, the optimization problem is said to have a low intrinsic dimensionality[5]. Random Search is also embarrassingly parallel, and additionally allows to include prior knowledge by specifying the distribution from which to sample.

3. **Bayesian optimization**

   Main article: Bayesian optimization

   Bayesian optimization is a global optimization method for noisy black-box functions. Applied to hyperparameter optimization, Bayesian optimization builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set. By iteratively evaluating a promising hyperparameter configuration based on the current model, and then updating it, Bayesian optimization, aims to gather observations revealing as much information as possible about this function and, in particular, the location of the optimum. It tries to balance exploration (hyperparameters for which the outcome is most uncertain) and exploitation (hyperparameters expected close to the optimum). In practice, Bayesian optimization has been shown[6][7][8][9] to obtain better results in fewer evaluations compared to grid search and random search, due to the ability to reason about the quality of experiments before they are run.

4. **Gradient-based optimization**

   For specific learning algorithms, it is possible to compute the gradient with respect to hyperparameters and then optimize the hyperparameters using gradient descent. The first usage of these techniques was focused on neural networks.[10] Since then, these methods have been extended to other models such as support vector machines[11] or logistic regression.[12]
   A different approach in order to obtain a gradient with respect to hyperparameters consists in differentiating the steps of an iterative optimization algorithm using automatic differentiation.[13][14]

5. **Evolutionary optimization**

   Main article: Evolutionary algorithm

   Evolutionary optimization is a methodology for the global optimization of noisy black-box functions. In hyperparameter optimization, evolutionary optimization uses evolutionary algorithms to search the space of hyperparameters for a given algorithm.[7] Evolutionary hyperparameter optimization follows a process inspired by the biological concept of evolution:

a. Create an initial population of random solutions (i.e., randomly generate tuples of hyperparameters, typically 100+)

b. Evaluate the hyperparameters tuples and acquire their fitness function (e.g., 10-fold cross-validation accuracy of the machine learning algorithm with those hyperparameters)

c. Rank the hyperparameter tuples by their relative fitness

d. Replace the worst-performing hyperparameter tuples with new hyperparameter tuples generated through crossover and mutation

e. Repeat steps 2-4 until satisfactory algorithm performance is reached or algorithm performance is no longer improving

Evolutionary optimization has been used in hyperparameter optimization for statistical machine learning algorithms[7], automated machine learning[15][16], deep neural network architecture search[17][18], as well as training of the weights in deep neural networks[19].

ML Terminologies -

Monday, August 20, 2018
11:50 AM

NLP -

word embedding - **Word embedding** is the collective name for a set of language modeling and feature learning techniques in natural language processing (NLP) where **words** or phrases from the vocabulary are mapped to vectors of real numbers.

ML Processing Nan Values -

Sunday, April 15, 2018
5:33 PM

Imputation and dealing with missing data a broad subject; you should start by researching standard material on this subject. The first question to figure out is Why is some data missing? and What is the process that causes data to be missing? It's important to understand how this happens, because this will affect what solution is appropriate.

Do re-search on missing data -

1. Randomly missing data
2. Non-randomly missing data

Assign them a separate category. All missing values will be treated as a separate category.

Difference between LabelEncoder and OneHotEncoder

LabelEncoder turn text value in a column into numeric values.
OneHotEncoder turn text value in a column into one or more binary columns that only have [0,1]

Thanks & Regards,
Parag Gurjar
Mobile No :+91 7387092108
Email: parag_gurjar@infosys.com
Infosys Limited