https://medium.com/subhrajit-roy/cracking-the-machine-learning-interview-1d8c5bb752d8

<https://www.glassdoor.com/Interview/machine-learning-interview-questions-SRCH_KO0,16_IP2.htm>

<https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/>

### Technologies We Like:

* + Python and specifically Numpy, SciPy, Pandas, scikit-learn
  + Neural network packages like TensorFlow and Torch
  + ML packages like LightGBM and XGBoost

<https://www.analyticsvidhya.com/blog/2017/05/questions-python-for-data-science/>

Phone Interview for 30 minutes, then the first data challenge, where you have one hour to solve it, and another one hour to discuss it.

Followed by a 4-hour data challenge.

How is the bias and variance in Random Forest and Gradient Boosting Tree?

----------------------

No Offer

Negative Experience

Average Interview

Application

I applied online. I interviewed at Kensho Technologies.

Interview

1. phone screen with data scientist

2. 30 minute assignment followed by call with data scientist

3. 2 hour assignment followed by call with data scientist

There were two timed assignments. They were of reasonable difficulty, but very stressful since they're timed and you have to take time out of your workday. It would be much better if they gave you a day to work on them after hours,

Show More

Interview Questions

Fit a time series model, multi-class classification

- = -------------------------------------- ------------

Interview

Background check first. Then give you some idea what the company is doing and what the team is working on. Ask 1-3 tech questions based on your experience. In the end, you can ask some questions to the interviewer. The total process takes around 30 mins.

Interview Questions

What kind of machine learning tool do you like to use most?

-------------------------------------------------- ---------------------------

Interview

Phone screen with HR, asking the typical questions (really just reviewing your resume). ~ 30 minutes

Phone screen with a data scientist, asking technical questions. Generally machine learning focused (explain this algorithm, explain this metric, etc.). Asked deep enough questions to make sure.

Sent a take home assignment. You get one hour to complete and return it, then follow with a one-hour conversation with a data scientist ...

------------------------- --- -----------------

Interview

Asked for a time to setup for coding challenge. Communicatively,

but never gave feedback to the interviewer.

I did not even get feedback for. No respect for one's time.

https://github.com/ShuaiW/ml-interview#linear-regression

<http://scikit-learn.org/stable/tutorial/machine_learning_map/> qc??

Interview Questions

Timeseries, multi-class classification

-------------------------------------------- --------------------

Machine Learning Interview Questions

---------------------------------------------- -----

https://www.dezyre.com/article/top-machine-learning-interview-questions-and-answers-for-2018/357

https://www.glassdoor.com/Interview/machine-learning-interview-questions-SRCH\_KO0,16.htm

https://www.springboard.com/blog/machine-learning-interview-questions/

TimeSeries Interview Questions

-------------------------------------

https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/

https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/

https://www.datacamp.com/community/tutorials/time-series-analysis-tutorial

https://www.kaggle.com/txtrouble/time-series-analysis-with-python

> Seasonality and removing first order dependency (seasonality, acf, pacf

http://www.statsoft.com/Textbook/Time-Series-Analysis#analysis

slide - 25

1. White Noise statistical tests

2. ACF Vs. PACF

Since autocorrelation is the linear correlation of a signal with itself at two different points in time, ACF (autocorrelation function) is just such correlation as a function of the lag h between two points of time, like acf(h)=corr(xt,xx+h)

PACF (partial autocorrelation function) is essentially the autocorrelation of a signal with itself at different points in time, with linear dependency with that signal at shorter lags removed, as a function of lag between points of time. Informally, the partial correlation between xt and xt+h is the autocorrelation between xt and xt+h without the contribution of xt+1,xt+2,....,xt+h−1.

Contrary to its ACF, which cuts off after lag 1, the PACF of an MA(1) model decays exponentially.

For a general MA(q) process, the ACF “cuts down” to zero after lag q and the PACF will have exponential

behavior depending on the characteristic roots of Θ(B) = (1 + θ1B + θ2B2 . . . + θqBq) = 0

Recognize types af AR and MA process charts from:

notes:

PACF tapers and ACF has significant corr for some lag and then shuts-off => MA process

ACF tapers and PACF has significant partial corr for some lag and then shuts-off => AR process

https://onlinecourses.science.psu.edu/stat510/node/60/

https://onlinecourses.science.psu.edu/stat510/node/62/

http://www.math.unm.edu/~ghuerta/tseries/week6\_1.pdf (good info - plots)

3. If an MA process is invertible, you can always find an Autoregressive representation.

ex: for MA(1) theta = 0.5 and 2; ACF and PACF are not distunguishable

|theta| < 1 => MA process is invertible

if invertible MA process can be transformed to AR process

The MA(1) Process: yt = εt + θεt−1 = (1 + θL)εt; εt ∼ WN(0, σ2)

If invertible:

yt = εt + θ\*yt−1 − θ^2\*yt−2 + θ^3\*yt−3 − ...

4. MA higher theta => higher variance

AR higher theta => lower variance (higher theta => slow decay in ACF)

5. A necessary and sufficient condition for an AR(1) process Y t = c +ϕ\*Y t-1 + ε t to be covariance stationary is that |ϕ|<1.

6. A useful Property for an AR(2) Process: The necessary conditions for an AR(2) process to be covariance stationary are that -1 < ϕ2 <1 and -2 <ϕ1 <2, and the sufficient conditions are that

ϕ1 + ϕ2 <1 and ϕ2 - ϕ 1 <1.

Github:

https://github.com/ikding/pycon\_time\_series

https://www.youtube.com/watch?v=zmfe2RaX-14

https://github.com/AileenNielsen/TimeSeriesAnalysisWithPython

**ICE Resume Part:**

1. Parametric vs Nonparametric Models

(http://mlss.tuebingen.mpg.de/2015/slides/ghahramani/gp-neural-nets15.pdf)

https://stats.stackexchange.com/questions/147587/are-random-forest-and-boosting-parametric-or-non-parametric

->Parametric models: assume some finite set of parameters 'theta'. Given the parameters, future predictions, x, are independent of the observed data, D:

P(x|'theta', D) = P(x|'theta')

therefore 'theta' capture everything there is to know about the data.

So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.

-> Parametric models summarizes the data with finite parameters.

Non-parametric models: assume that the data distribution cannot be defined in terms of such a finite set of parameters. But they can often be defined by

assuming an infinite dimensional 'theta'. Usually we think of 'theta' as a function

Read question 5 below as well.

Examples:

**Parametric Non-parametric Application**

polynomial regression Gaussian processes function approx.

logistic regression Gaussian process classifiers classification

mixture models, k-means Dirichlet process mixtures clustering

hidden Markov models infinite HMMs time series

factor analysis / pPCA / PMF infinite latent factor models feature discovery

Simple Nueral Network SVM

-> https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/

Indeed simple multilayer perceptron neural nets are parametric models. Non-parametric models do not need to keep the whole dataset around, but one example of a non-parametric algorithm is kNN that does keep the whole dataset. Instead, non-parametric models can vary the number of parameters, like the number of nodes in a decision tree or the number of support vectors, etc.

1. Why KNN is considered a non-parametric model?

Here 'k' is hyper-parameter.

Generally in a parametric model, once u learn the parameters we can thow away the training data. But KNN requires it to have the training data.

1. Parametric and Non-parametric k-means qc??

Understand k-means algorithm

1. What is the output of a GMM model? qc??
2. Random Forest is parametric or non-parametric?

<https://stats.stackexchange.com/questions/147587/are-random-forest-and-boosting->parametric-or-non-parametric

The depth of the tree can change if more sample data is used.

Parametrical models have parameters (inferring them)or assumptions regarding the data distribution, whereas RF ,neural nets or boosting trees have parameters related with the algorithm itself, but they don't need assumptions about your data distribution or classify your data into a theoretical distribution. In fact almost all algorithms have parameters such as iterations or margin values related with optimization.

1. z-value in Logistic Regression

z-value = (regression coeficient - 0)/ Std.error; mu = 0 cause that is our null hypothesis that regression soeficient is Zero.

|z| vlaue high implies Significant but we look at p-value Pr(Z > |z|), since we want |z| to be high => Pr(Z > |z|) should be low (p-value should be low)

if |z| is 4 and Z value corresponding to 95% confidence is 1.96 => coeficient is significant

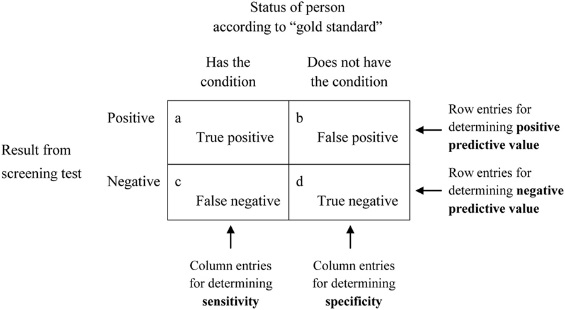
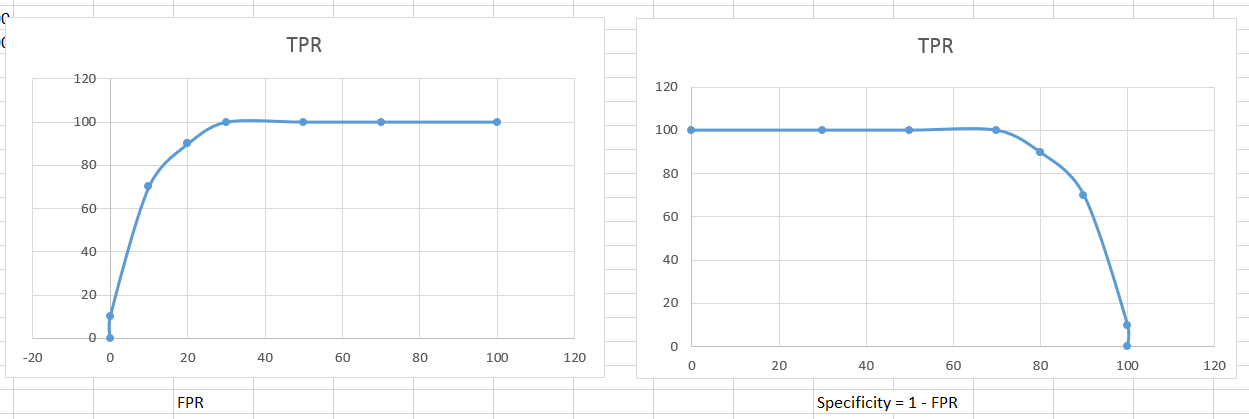
The Z value of 1.96 => we are assuming normal distribution for statistics or coeficients from different samples of data?

http://logisticregressionanalysis.com/1577-what-are-z-values-in-logistic-regression/

myquestion in stackover: https://stats.stackexchange.com/questions/363322/logistic-regressin-z-value

1. Assumptions of Logistic regression (check Udacity Gatech - not sure what's in there?) qc??
2. Check in DVA homeworks, how is logistic preidcted probability decided? Did I set up a cut-off of 0.5? qc??

http://www.statisticssolutions.com/assumptions-of-logistic-regression/

1. ROC Curve
   1. <https://www.youtube.com/watch?v=OAl6eAyP-yo>
   2. Refer pic in Drop box “ROC Curve”
      1. The histogram (predict prob Vs count as per true label) of 0 and 1 in the image depends on model.
   3. We want to quantify the performance of a classifier using ROC curve. (AUC does the job)
      1. AUC is the area under the ROC Curve.
      2. Random guessing AUC = 0.5
      3. Max AUC is 1
   4. ROC curve is calculated the exact same way even if the dataset is unbalanced.
   5. ROC can be used for multi-classification using one-vs-all approach. For 3 labels, we should have 3 ROC curves for all combinations.
   6. Choosing a classification threshold is more of a business decision or domain or problem specific, It depends on whether we want to maximize TPR or minimize FPR.
   7. 
      1. Sensitivity=[a/(a+c)]×100
      2. Specificity=[d/(b+d)]×100
      3. Positive predictive value(PPV)=[a/(a+b)]×100
      4. Negative predictive value(NPV)=[d/(c+d)]×100
      5. TPR = Sensitivity, FPR = 1 - Specificity
   8. 
2. How to evaluate a classifier in scikit-learn
   1. <https://www.youtube.com/watch?v=85dtiMz9tSo>
      1. ROC, AUC, Confusion matrix
      2. Logistic
      3. Adjusting threshold
3. Multi-class Vs Multi-Label
   1. An instance of data has only one class but could have more than one label
4. Multi-class Classification using Logistic Regression

http://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

* 1. from sklearn import linear\_model
  2. # Train multinomial logistic regression
  3. mul\_lr = linear\_model.LogisticRegression(multi\_class='multinomial', solver='newton-cg').fit(train\_x, train\_y)
  4. sklearn.linear\_model.LogisticRegression(penalty=’l2’, dual=False, tol=0.0001, C=1.0, fit\_intercept = True, intercept\_scaling=1, class\_weight=None, random\_state=None, solver=’liblinear’, max\_iter=100, multi\_class=’ovr’, verbose=0, warm\_start=False, n\_jobs=1)
     1. fit\_intercept => if a constant bias needs to be added
     2. class\_weight => dict or ‘balanced’ (The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data )
     3. solver:
        1. For small datasets, ‘liblinear’ is a good choice, whereas ‘sag’ and ‘saga’ are faster for large ones
        2. multiclass problems, only ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’ (‘newton-cg’, ‘lbfgs’ and ‘sag’ only handle L2 penalty, whereas ‘liblinear’ and ‘saga’ handle L1 penalty)
     4. multi\_class: ‘ovr’ (binary) or ‘multinomal’ (multi-class)
     5. n\_jobs: number of cores to use
  5. sklearn.linear\_model.SGDClassifier(loss=’hinge’, …………………….)
     1. loss = ‘log’ gives logistic regression
     2. **coef\_** : array, shape (1, n\_imports) if n\_classes == 2 else (n\_classes, n\_features)
     3. In the case of multi-class classification coef\_ is a two-dimensionally array of shape=[n\_classes, n\_features] and intercept\_ is a one dimensional array of shape=[n\_classes]. The i-th row of coef\_ holds the weight vector of the OVA classifier for the i-th class; classes are indexed in ascending order (see attribute classes\_). Note that, in principle, since they allow to create a probability model, loss="log" and loss="modified\_huber" are more suitable for one-vs-all classification.
     4. http://scikit-learn.org/stable/modules/sgd.html

1. KNN
   1. <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
      1. Parameters:
         1. Weights
         2. Algorithm: ‘auto’, ‘ball\_tree’, ‘kd\_tree’ e.t.c
            1. ‘auto’ will attempt to find the best algorithm based on input data
         3. P, metric: p = 2 and metric = ‘minkowski’ => these are defaults and this specific case is Standard Euclidean distance.
   2. Pros and Cons of KNN
      1. <https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>
   3. <https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>
2. K-Fold Cross Validation
   1. **from** **sklearn.model\_selection** **import** KFold
3. Parallel Job for Visualization tool - R
4. Multi-class coding (also look in to Ensemble methods)
   1. <https://datascienceplus.com/multi-class-text-classification-with-scikit-learn/>
   2. <https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/>
   3. <https://www.codementor.io/agarrahul01/multiclass-classification-using-random-forest-on-scikit-learn-library-hkk4lwawu>
5. Random Forest
   1. Can be used in Feature Importance
      1. I have used it kaggle\_pr1 project (home Credit Default Risk Prediction)
      2. Read ‘Varaiable Importance’ in https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/
   2. <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>
   3. Algo:
      1. Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but with replacement. This sample will be the training set for growing the tree.
      2. If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.
      3. Each tree is grown to the largest extent possible and there is no pruning
      4. Predict new data by aggregating the predictions of the n trees (i.e., majority votes for classification, average for regression).
   4. How Random Forest deals with missing data? Qc??
      1. It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing
   5. How Random Forest balances imbalanced data sets? Qc??
      1. It has methods for balancing errors in data sets where classes are imbalanced
6. Bootstrap Aggregation (Bagging) (key words: random forest, decision trees)
   1. Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method.
   2. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.
   3. Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithm that have high variance. An algorithm that has high variance are decision trees, like classification and regression trees (CART).
   4. https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/
7. Metrics: qc??
   1. ROC/AUC
   2. Confusion Matrix
   3. Log Loss
   4. MAE, MSE, R^2
8. Classification and Regression Trees (CART) qc??
   1. <https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/>
9. Introduction to Clustering Methods:
   1. <https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/>
   2. <https://www.analyticsvidhya.com/blog/2013/11/getting-clustering-right/> qc??
      1. Types of Clustering:
         1. Connectivity, Centroid, Distribution models and Density Models
            1. Distribution: Ex: Expectation-Maximization
      2. Difference between K Means and Hierarchical clustering
   3. It should be kept in mind while performing distance based methods we must attempt to scale the data, so that the feature with lesser significance might not end up dominating the objective function due to its larger range. In addition, features having different unit should also be scaled thus providing each feature equal initial weightage and at the end we will have a better prediction model.
      1. "Normalizing" a vector most often means dividing by a norm of the vector, for example, to make the Euclidean length of the vector equal to one.
      2. "Standardizing" a vector most often means subtracting a measure of location and dividing by a measure of scale. For example, if the vector contains random values with a Gaussian distribution, you might subtract the mean and divide by the standard deviation, thereby obtaining a "standard normal" random variable with mean 0 and standard deviation 1.
      3. <https://www.analyticsvidhya.com/blog/2016/07/practical-guide-data-preprocessing-python-scikit-learn/>
      4. Feature Standardization:
         1. In the link above, scaling increases accuracy for KNN whereas doesn’t improve for Logistic. Standardization improves accuracy of Logistic Regression. In logistic regression, each feature is assigned a weight or coefficient (Wi). If there is a feature with relatively large range and it is insignificant in the objective function then logistic regression will itself assign a very low value to its co-efficient, thus neutralizing the dominant effect of that particular feature, whereas distance based method such as kNN does not have this inbuilt strategy, thus it requires scaling.
            1. For Logistic we have to Standardize rather than Scaling
         2. Elements such as l1 ,l2 regularizer in linear models (logistic comes under this category) and RBF kernel in SVM in objective function of learners assumes that all the features are centered around zero and have variance in the same order.
         3. Features having larger order of variance would dominate on the objective function as it happened in the previous section with the feature having large range (KNN).
         4. Other learners like kNN with euclidean distance measure, k-means, SVM, perceptron, neural networks, linear discriminant analysis, principal component analysis may perform better with standardized data.
         5. **Note :**Choosing between scaling and standardizing is a confusing choice, you have to dive deeper in your data and learner that you are going to use to reach the decision. For starters, you can try both the methods and check cross validation score for making a choice.
      5. https://sebastianraschka.com/Articles/2014\_about\_feature\_scaling.html
         1. Tree-based classifier are probably the only classifiers where feature scaling doesn’t make a difference.
         2. As a rule of thumb I’d say: When in doubt, just standardize the data, it shouldn’t hurt.
         3. Intuitively, we can think of gradient descent as a prominent example (an optimization algorithm often used in logistic regression, SVMs, perceptrons, neural networks etc.); with features being on different scales, certain weights may update faster than others since the feature values xjxj play a role in the weight updates

Δwj = −η∂J/∂wj = η∑ (t(i)−o(i))\*x(i)j

* + - 1. Some examples of algorithms where feature scaling matters are: in the above link
      2. The cost of having this bounded range(Min-Max scaling) - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers.
    1. Normalization(scaling) retains the input data distribution. Standardization doesn’t unless the input data is Gaussian
    2. Standardizing can make training faster and reduce the chance of getting stuck in local optima qc?? (Neural Nets)
  1. <https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/>
     1. Normalization is useful when your data has varying scales and the algorithm you are using does not make assumptions about the distribution of your data, such as k-nearest neighbors and artificial neural networks.
     2. Standardization is useful when your data has varying scales and the algorithm you are using does make assumptions about your data having a Gaussian distribution, such as linear regression, logistic regression and linear discriminant analysis.

Performing PCA on un-normalized variables will lead to insanely large loadings for variables with high variance. In turn, this will lead to dependence of a principal component on the variable with high variance. This is undesirable.

TEST the NN via Scaling and Standardization and Compare Performance

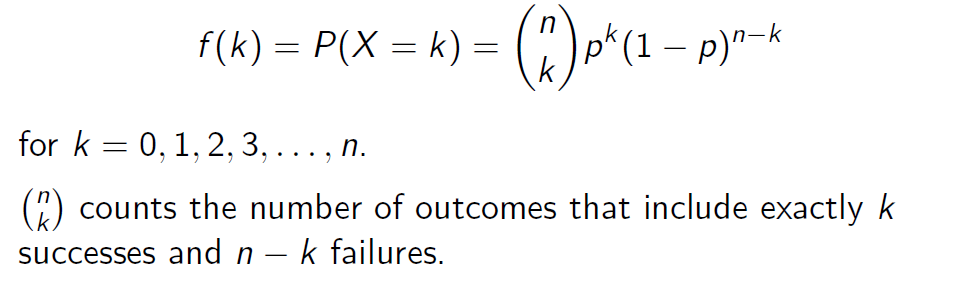
Min-Max scaler gave an accuracy of 63.5% very much close to a guess (63% -1 classified data), whereas Standardization gave an accuracy of 74%. (self test)

from sklearn.preprocessing import scale, normalize, MinMaxScaler

normalize() does Standardization, same as using scale with default settings

<http://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html>

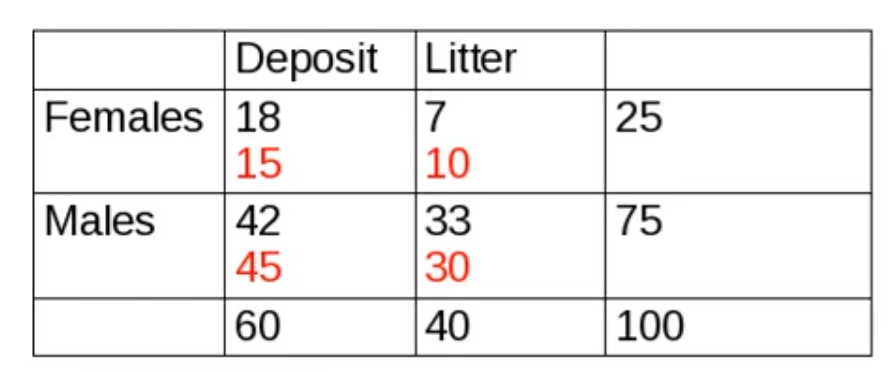
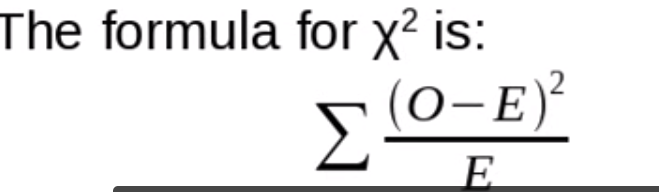
Should we center sparse data qc?? What is scipy.sparse CSC matrix ?? context: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html

1. Principal Component Analysis: qc??
   1. <http://sebastianraschka.com/Articles/2014_pca_step_by_step.html>
   2. <https://sebastianraschka.com/Articles/2014_about_feature_scaling.html>
   3. Even though PCA is used when we have high dimensional data possibly correlated features. Having correlated features can make PCA overemphasize more on these correlated features.
   4. **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?**
      1. 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.
      2. https://www.analyticsvidhya.com/blog/2016/09/40-interview-questions-asked-at-startups-in-machine-learning-data-science/
2. Metrics: Precision, Recall, F1-Score, Confusion Matrix qc??
3. VIF:  variance inflation factor is the ratio of variance in a model with multiple terms, divided by the variance of a model with one term alone.[[1]](https://en.wikipedia.org/wiki/Variance_inflation_factor#cite_note-1) It quantifies the severity of [multicollinearity](https://en.wikipedia.org/wiki/Multicollinearity) in an [ordinary least squares](https://en.wikipedia.org/wiki/Ordinary_least_squares) [regression](https://en.wikipedia.org/wiki/Linear_regression) analysis.
4. Bernoulli Distribution: only two outcomes possible for a Bernoulli event
   1. P(X=1) = p, p(X=0) = 1-p
   2. E[X] = p, Var[X] = p\*(1-p)
5. Binomial Distribution: ‘n’ repeated Bernoulli independent trials (Independent trials is the distribution assumption)
   1. ‘k’ - no. of success in ‘n’ trials
   2. 
   3. Since independent trials: E[X] = np and Var[X] = np(1-p)
6. Regression can be enhanced by using along with DTrees (Trick To Enhance Power Of Regression Model using Decision Trees.pdf) in /BD/ folder (<https://www.analyticsvidhya.com/blog/2013/10/trick-enhance-power-regression-model-2/>)
7. Chi-Squared Calculation: <https://www.youtube.com/watch?v=VskmMgXmkMQ>
   1. Chi-Square = (Observed - Expected)^2/Expected

Male

Female

Node (100 pop)

* 1. Null Hypothesis: No Significance (Male or Female doesn’t have a significance whether Deposit/Litter)
     1. 
     2. The highlight red values are expected based on Hypothesis:
        1. Since no significance; % of females that deposit or litter = %females in Total Pop
        2. Therefore Expected = 25% \* Total Deposit (60) = 15
     3. 
     4. Look for Chi-square in the table with same degrees of freedom
        1. Here DOF is = 1: After calculating value of 15, the rest of expected values are obvious i.e it has 1 DOF
        2. DOF formula: (num. of rows - 1) \* (num. of columns or num. of classes -1 )

1. Entropy & Information Gain: The formula for entropy is -Sigma(p(x)) \* log p(x); p(x) is proportion of classifications.
   1. <https://stackoverflow.com/questions/1859554/what-is-entropy-and-information-gain>
   2. Entropy is measure of impurity (uncertainty) and Information gain is measure of purity.
      1. Entropy maximum if all classes are equal in proportion
   3. Information Gain = Entropy\_beforeSplit - (Entropy\_left + Entropy\_right)
      1. i.e. split should decrease the Entropy
2. Split in Regression Trees:
   1. Reduction In Variance method
   2. RSS (ISLR, 7th Ed Page 307)
3. Overfitting in Decision Trees:
   1. Setting constraints on tree size (Greedy, best step at the moment)
      1. Read about the parameters used to set constraints in “A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python).pdf”
   2. Tree Pruning (looks ahead) - hence better than Setting Constraints
      1. Details of Pruning - sequence of sub trees qc??
4. TF-IDF: (Term Frequency - Inverse Document Frequency)
   1. Tf \* idf
   2. Idf = log(N/df)
   3. df = document frequency, number of docs the word appeared in, N - number of docs
   4. Decreases the importance of common words not relevant subject matter.
5. In regression, it is often recommended to **center** the variables so that the predictors have mean 00. This makes it so the intercept term is interpreted as **the expected value of**YiYi**when the predictor values are set to their means**. Otherwise, the intercept is interpreted as the expected value of YiYiwhen the predictors are set to 0, which may not be a realistic or interpretable situation.
6. Doubt: The main goal of standardizing features is to help convergence of the technique used for optimization. For example, if you use Newton-Raphson to maximize the likelihood, standardizing the features makes the convergence faster. Otherwise, you can run your logistic regression without any standardization treatment on the features. Qc?? Wait for reply from NLP