# KNN

Distance-based

Find neighbors based on distance → majority voting, average…

K = 1, high variance, low bias → overfit

K = 100, low variance, high bias, smoother

Find best k, distance → try different k, plot test error vs. k

Good for small p. Large p → curse of dimensionality

How to choose k:

* Simplest approach:
  + K = sqrt(# data points)
* Validation error vs. k (plot)
  + More robust: CV error =

Why is high-dimension problematic for KNN?

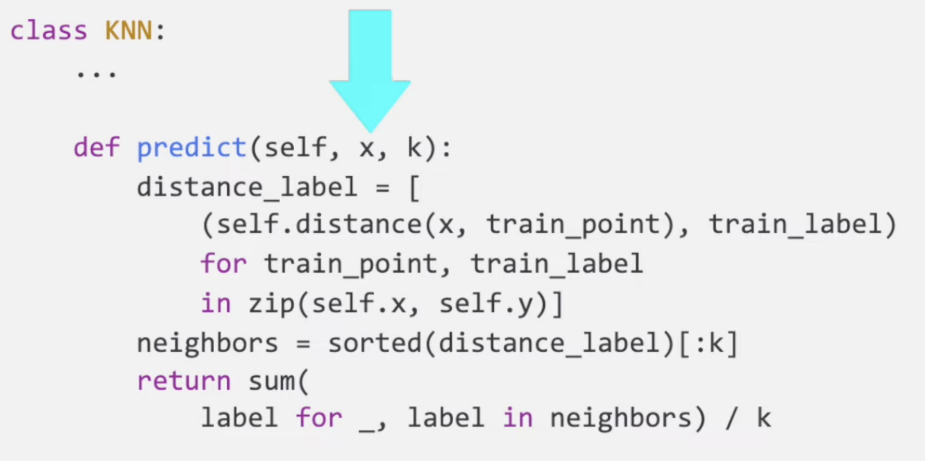
* Knn requires points to be close in every single dimension.
* As dimension increases, it is harder for two specific points to be close to each other (probability of at least in one dimension that two point is not close increases) → no points close to each other

总结：used when small dataset with small number of features, and you don’t know the shape of your data and the way input and output are related.

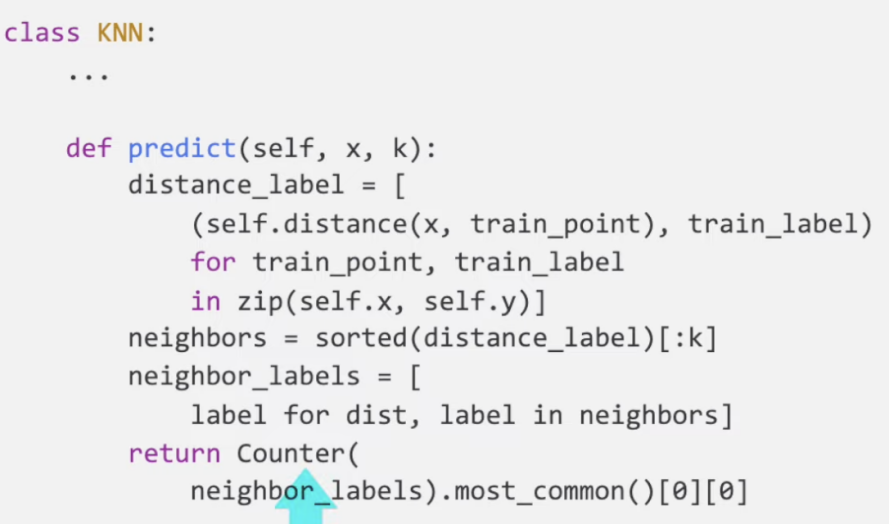
|  |  |
| --- | --- |
| pro | con |
| Simple to implement | Computationally expensive (需要compute distance from new observation to all known samples) → slow for large dataset |
| No assumptions about data  (Linear: assume linear relationship; Naive B: assume features are independent) | Sensitive to imbalance datasets (poor results for minority classes) |
| Few tuning parameters (k, distance metric) | Sensitive to irrelevant features (make distance less meaningful for identifying similar neighbors) |
| Multi-class can be solved | Scaling of data is a must |
|  | Cannot work well with outlier and missing values |

Implementation:

Regressor:



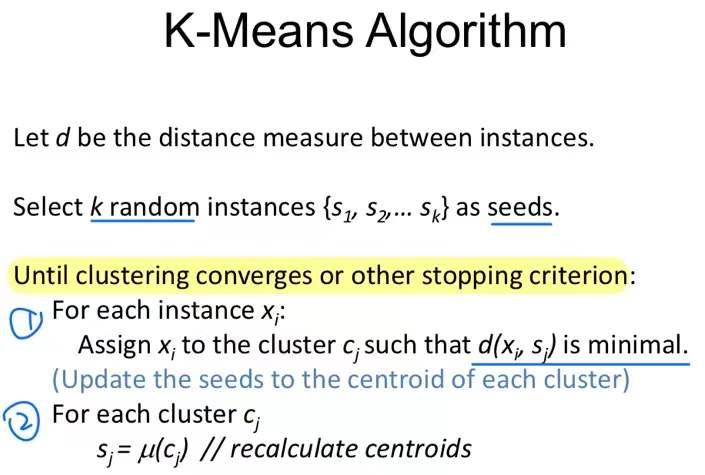
Classification:



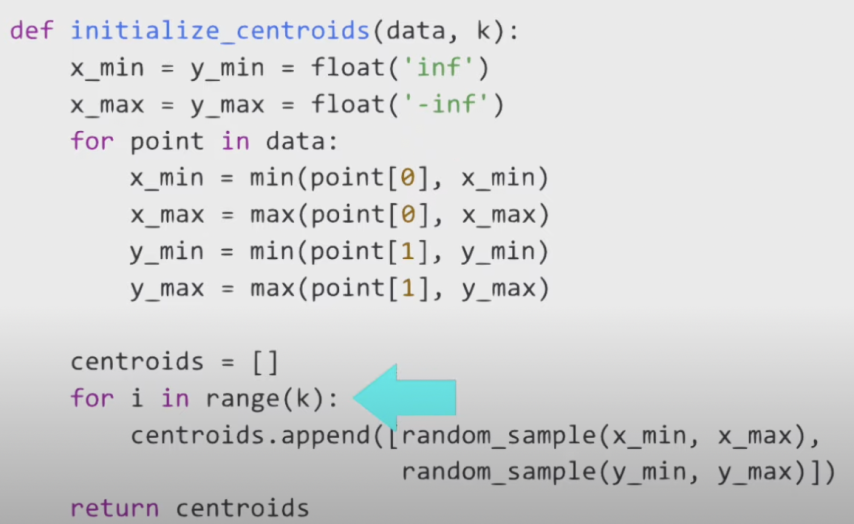
# K-MEANS

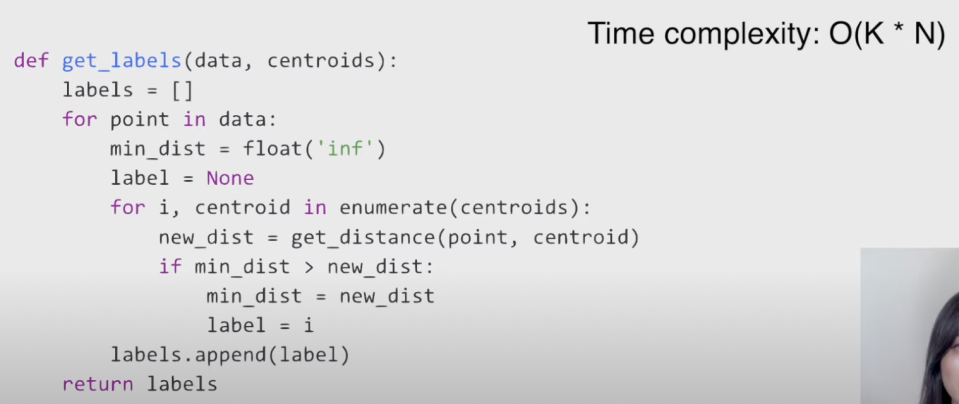
Algorithm:

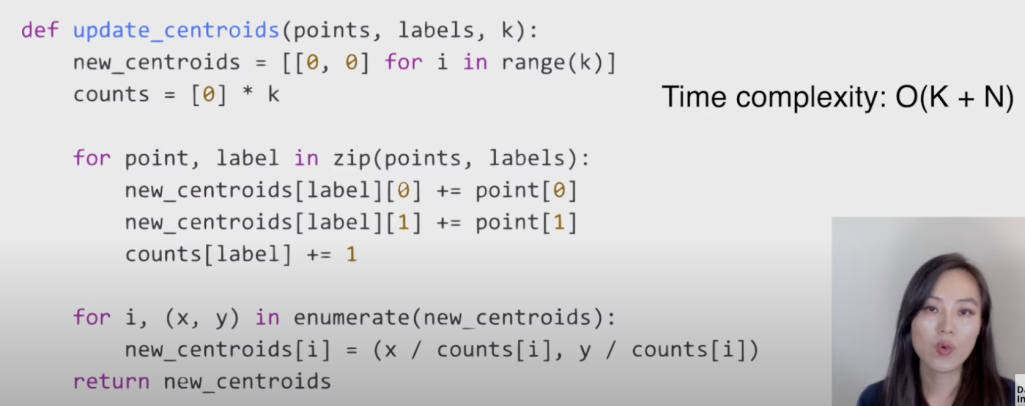
* Randomly select k centers → k clusters
* Calculate the new centers → k clusters
* Keep doing until they converge to local minimum

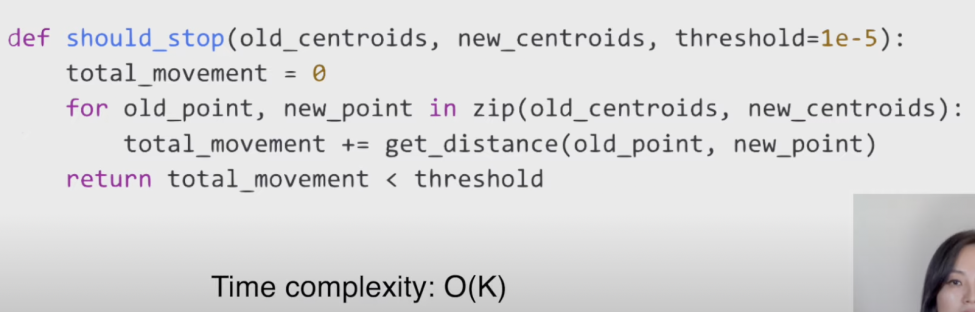
****

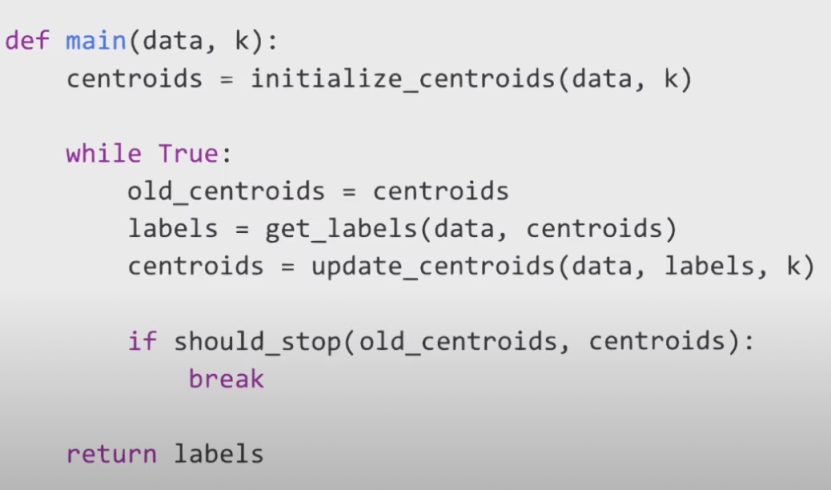
**Implementation**

****

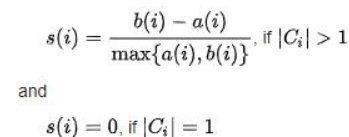
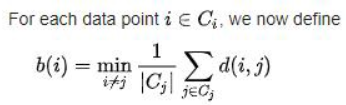
****

****

****

****

Best K

1. The Elbow Method
   1. Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS first starts to diminish. In the plot of WSS-versus-k, this is visible as an elbow.
   2. WSS = square of the distance of the point from its predicted cluster center.
2. The Silhouette Method (more computational intensive, so not suitable for big dataset)
   1. The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).
      1. ( -1,1). A high value is desirable and indicates that the point is placed in the correct cluster.
      2. If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters.
   2. 
   3. a(i): similarity of the point i to its own cluster
      1. average distance of i from other points in the cluster.
   4. b(i): dissimilarity of i from points in other clusters.
      1. 

|  |  |
| --- | --- |
| Pros | Cons |
| Simple to implement | Choosing k manually |
| Scales to large data sets | Being dependent on initial values |
| Guarantee converge | Clustering outliers |
| Generalizes to clusters of different shapes and sizes, such as elliptical clusters | performed in numerical data only |
| Computational cost low |  |

<https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages>

Alternatives

* Hierarchical clustering: dendrogram
  + Doesn’t require a specific number of clusters
  + A more interpretable and informative output
* Density clustering DBSCAN

# Time Series

<https://neptune.ai/blog/arima-vs-prophet-vs-lstm>

Interpolation: fill missing value

<https://www.analyticsvidhya.com/blog/2021/06/power-of-interpolation-in-python-to-fill-missing-values/#:~:text=Interpolation%20is%20mostly%20used%20while,the%20mean%20of%20the%20month>.

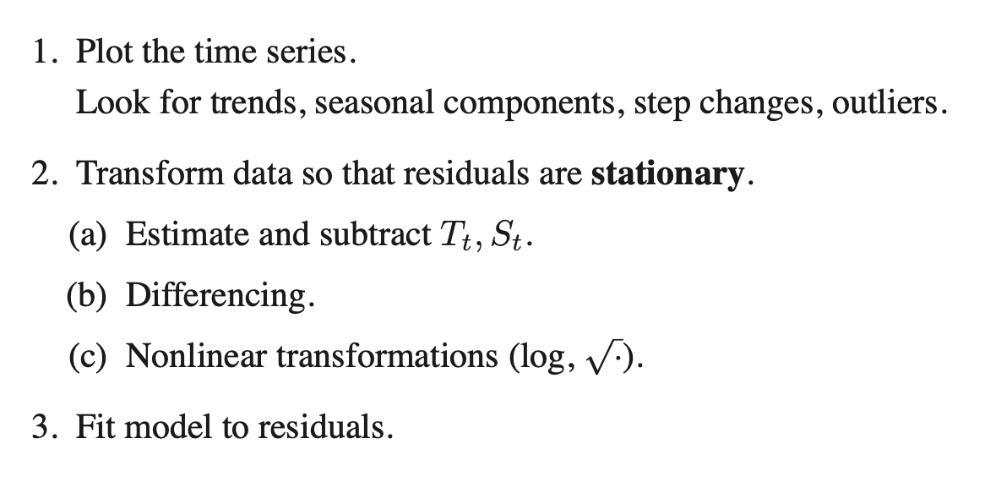
## What is Time Series Analysis

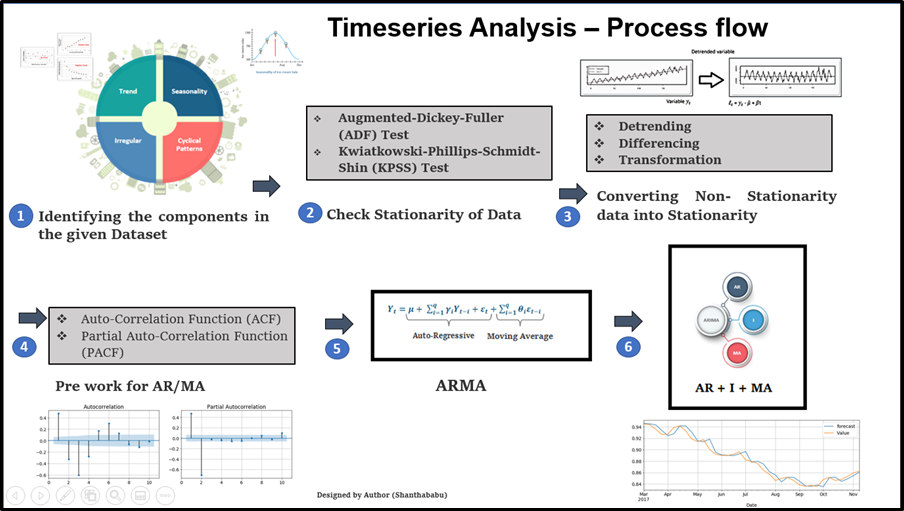
A time series is nothing but a sequence of various data points that occurred in a successive order for a given period of time

Objectives:

* To understand how time series works, what factors are affecting a certain variable(s) at different points of time.
* provide the consequences and insights of features of the given dataset that changes over time
* Supporting to derive the predicting the future values of the time series variable.
* Assumptions: There is one and the only assumption that is “stationary”, which means that the origin of time does not affect the properties of the process under the statistical factor.

## TSA Framework

* Collecting the data and cleaning it
* Preparing Visualization with respect to time vs key feature
  + Plot time series
    - Look for trends, seasonal components, step changes, outliers
* Observing the stationarity of the series
  + Transform the data so that residuals are stationary
  + 
* Developing charts to understand its nature
* Model building – AR, MA, ARMA and ARIMA
  + (fit model to residuals)
* Extracting insights from prediction



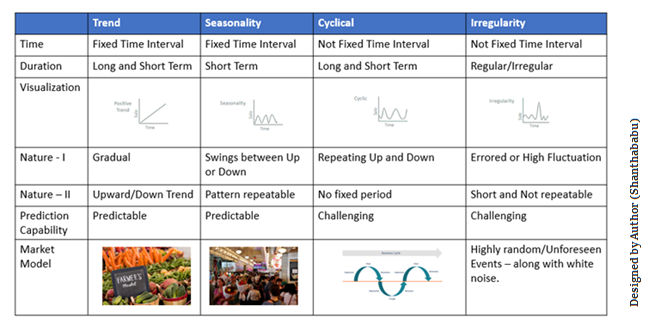
## Significance and its types

TSA is the backbone for prediction and forecasting analysis, specific to the time-based problem statements.

* Analyzing the historical dataset and its patterns
* Understanding and matching the current situation with patterns derived from the previous stage.
* Understanding the factor or factors influencing certain variable(s) in different periods.

## Components of TSA

* Trend
  + there is no fixed interval
  + no divergence within the given dataset is a continuous timeline
  + would be Negative or Positive or no Trend
* Seasonality
  + regular or fixed interval shifts within the dataset in a continuous timeline.
  + Would be bell curve or saw tooth
* Cyclical
  + no fixed interval, uncertainty in movement and its pattern
* Irregularity
  + Unexpected situations/events/scenarios and spikes in a short time span.



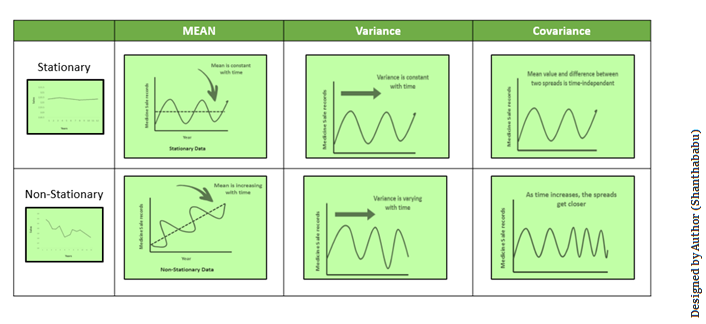
## Limitations

* Similar to other models, the missing values are not supported by TSA
* The data points must be linear in their relationship.
* Data transformations are mandatory, so a little expensive.
* Models mostly work on Uni-variate data.

## Data Types of Time Series

<https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322>

Let’s discuss the time series’ data types and their influence. While discussing TS data-types, there are two major types.

* Stationary: without having Trend, Seasonality, Cyclical, and Irregularity component of time series
  + stationarity means that the statistical properties of a process generating a time series do not change over time. It does not mean that the series does not change over time, just that the way it changes does not itself change over time.
    - The MEAN value is constant
    - The VARIANCE is constant with respect to the time-frame
    - The COVARIANCE measures the relationship between two variables.
      * Mean value and difference between two spreads is time-independent
* Non- Stationary: This is just the opposite of Stationary.
* 

## Methods to check Stationarity

<https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/#:~:text=Two%20tests%20for%20checking%20the,check%20stationarity%20in%20Time%20Series>.

1. Augmented Dickey-Fuller (ADF) Test / Unit Root Test:

most popular statistical test with the following assumptions

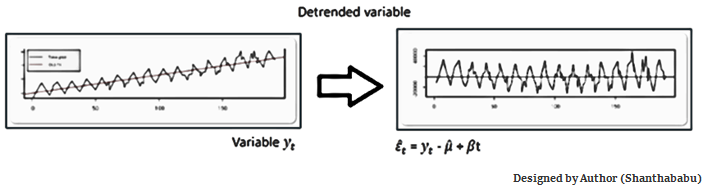
* 1. H0: Series is non-stationary
  2. Ha: Series is stationary
     1. p-value <= 0.05 Accept (H1)

1. Kwiatkowski–Phillips–Schmidt–Shin (KPSS):
   1. These tests are used for testing a NULL Hypothesis (H0), that will perceive the time-series as stationary around a deterministic trend against the alternative of a unit root.

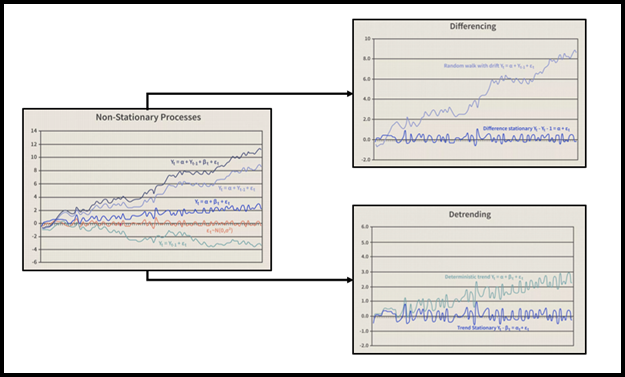
<https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/>

## Converting Non-stationary into stationary

* Detrending
  + It involves removing the trend effects
    - showing only the differences in values from the trend. it always allows the cyclical patterns to be identified.

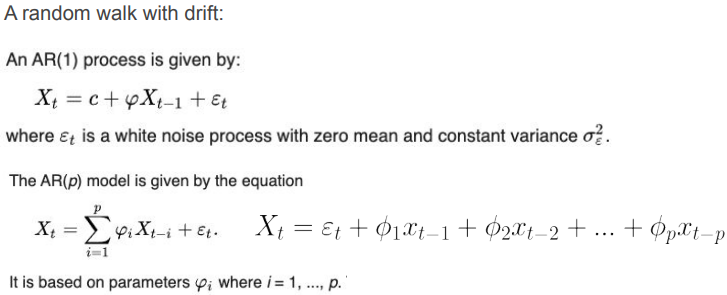


* Differencing
  + remove the series dependence on time and stabilize the mean of the time series, so trend and seasonality are reduced during this transformation.
  + Yt= Yt – Yt-1
  + Yt=Value with time



* Transformation
  + three different methods:
    - Power Transform, Square Root, and Log Transfer (most commonly used)

## Auto Regressive AR

* A representation of a type of random process
* specifies that the output variable depends linearly on its own previous values and on a stochastic term; thus the model is in the form of a stochastic difference equation.
* 
* Yt =C+b1 Yt-1+ b2 Yt-2+……+ bp Yt-p+ Ert

## Moving Average MA

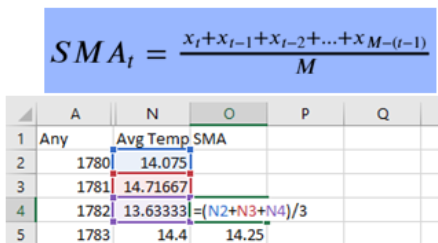
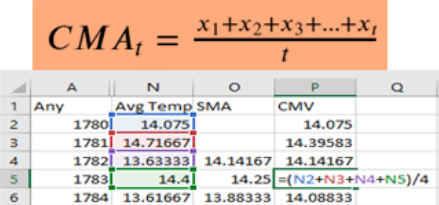
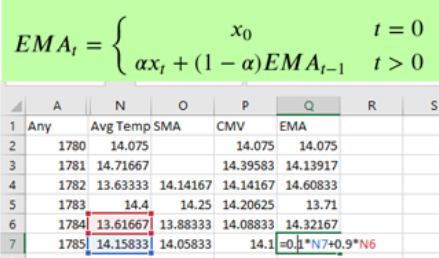
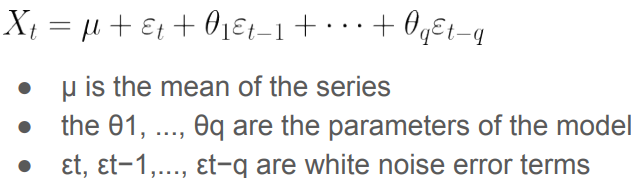
This method is slick with random short-term variations. (这种方法很灵活，具有随机的短期变化)

Relatively associated with the components of time series.

The Moving Average (MA) (Or) Rolling Mean:

* In which MA has been calculated by taking averaging data of the time-series, within k periods.

Types of MA:

* Simple Moving Average (SMA)
  + unweighted mean of the previous M or N points
  + The selection of sliding window data points depends on the amount of smoothing
    - increase the value of M or N → improves the smoothing at the expense of accuracy.
  + 
* Cumulative Moving Average (CMA)
  + unweighted mean of past values, till the current time (之前所有)
  + 
* Exponential Moving Average (EMA)
  + mainly used to identify trends and to filter out noise.
  + The weight of elements is decreased gradually over time: gives weight to recent data points, not historical ones. Compared with SMA, the EMA is faster to change and more sensitive.
  + α –>Smoothing Factor
    - [0,1]
    - represents the weighting applied to the very recent period
  + 
* A moving-average model is conceptually a linear regression of the current value of the series against current and previous (observed) white noise error terms or random shocks.
  + 

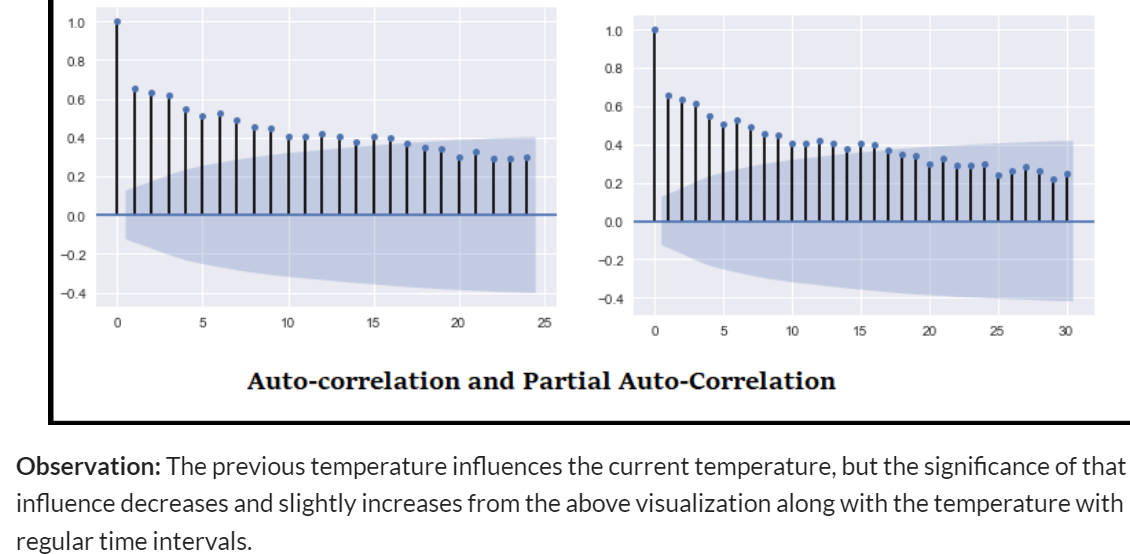
## ACF and PACF

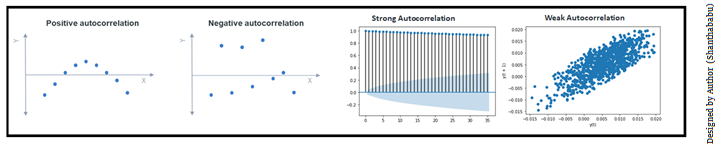
Auto-Correlation Function (ACF):

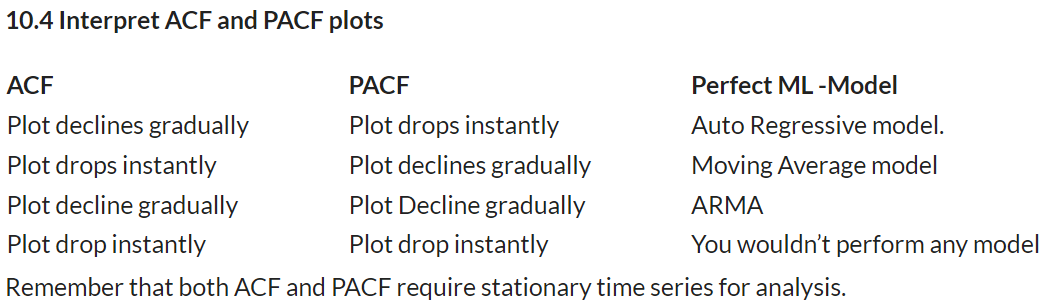
* measures the degree of the similarity between a given time series and the lagged version of that time series at different intervals that we observed. (how similar a value and previous value)
* used to identify trends and the influence of former observed values on the currently observed values.

Partial Auto-Correlation (PACF):

* It always shows the correlation of the sequence with itself with some number of time units per sequence order in which only the direct effect has been shown, and all other intermediary effects are removed from the given time series.

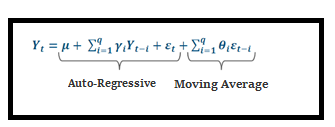






## ARMA and ARIMA

ARMA

* a combination of the Auto-Regressive and Moving Average model for forecasting
* 
* ARMA is best for predicting stationary series. So ARIMA came in since it supports stationary as well as non-stationary.

ARIMA = AR+I+MA

* AR ==> Uses the past values to predict the future
* MA ==> Uses the past error terms in the given series to predict the future
* I==> uses the differencing of observation and makes the stationary data

Understand the Signature of ARIMA

* p:
  + log order => # of AR lag observations
* d:
  + degree of differencing => # times that the raw observations are differenced.
* q:
  + order of MA => the size of the moving average window

# Neural Network

## **Perceptron**

Take in various input (at input layer)

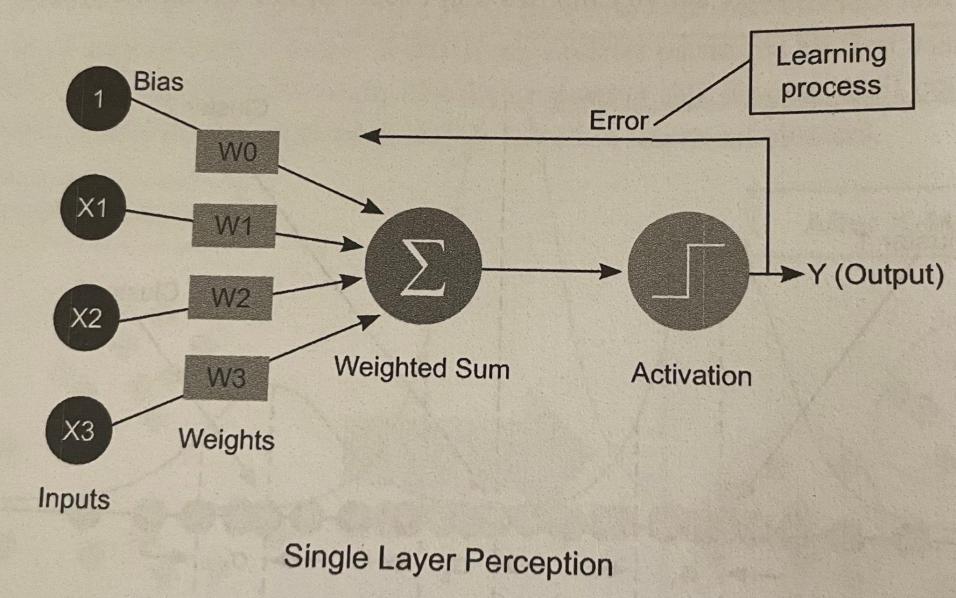
→ weight these inputs

→ combine weighted inputs through a linear combination

→ if combined weighted output past a threshold (set by activation function)

→ send output to other layers.

* Base unit is perceptron → combine perceptrons to form NN (also called multi-layer perceptrons MLPs)



Activation function:

* Usually nonlinear
* 常用的：sigmoid, ReLU, softplus…
* 在ML-intensive的面试中可能问到

Hidden Layers:

* Layers that are neither input layer nor output layer
* Allows for specific transformations of the data
* 每层可以有particular output
  + 例子： 导航，其中一层识别stop sign，另外一层识别traffic lights…

## **Backpropagation**

Learning process of NN 用来更新weight的

* Modifies the weights of NN iteratively, through calculations of deltas between predicted and expected outputs

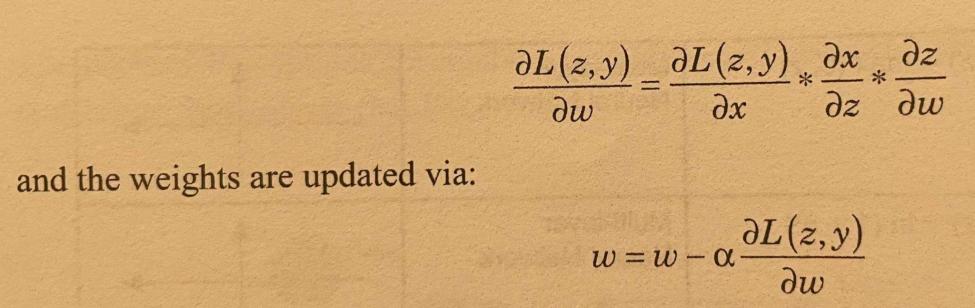
—> then, the weights are updated backward through earlier layers visa stochastic gradient descent

—> process continues and the weights that minimize the loss function are found. (直到loss function最小化）

Loss function:

* Regression 常用MSE
* Classification 常用 cross-entropy
* Given L, update weights w through the chain role of the following form:

z = model’s output, = learning rate



面试：ML-heavy role 会要求explain the technical details behind basic backpropagation, for basic methods 例如linear or logistic regression

Hyperparameters:

= learning rate

* too small, optimization process may freeze
* too large, optimization might converge prematurely at a suboptimal solution

NN 里其他hyperparameters:

* # hidden layers, activation function used, batch size…
* 面试里可以知道how each hyperparameter affects a NN’s training time and model performance

## **Training NN**

General Framework:

* Vanishing gradients (梯度消失)

Training Optimization techniques

Transfer Learning

Avoid Overfitting:

* Add more training data
* Standardize features
* Batch normalization
* Dropout

## **Types of NN**

CNN

RNNs

LSTMs

# Reinforcement Learning

# 题目

Eigenvalues & eigenvector

* Eigenvalues are the directions along which a particular linear transformation acts by flipping, compressing, or stretching.
* Eigenvectors are for understanding linear transformations. In data analysis, we usually calculate the eigenvectors for a correlation or covariance matrix.

## **Easy**

**1. Robinhood: how to build a binary classifier for an** unbalanced dataset

**2. Easy. Square: a model minimizes** MSE VS. **the model minimizes** MAE, **what are differences and in which case would each error metric be appropriate?**

Both are measures of distances between vectors.

Error:

* The main difference is that, in MSE, errors are squared before being averaged, meaning there is a relatively high weight given to large errors. → MSE is useful when large errors are trying to be avoided.

Outlier:

* Outliers disproportionately affect MSE more than MAE → MAE is more robust to outliers

Computation-wise:

* MSE is easier to use, since the gradient calculation is more straightforward than that of MAE, which requires some linear programming to compute the gradient 看不懂.

So, if the model needs to be computationally easier to train or needs to be robust to outliers, MSE should be used. MSE corresponds to maximizing the likelihood of Gaussian random variables, and MAE not. MSE is minimized by conditional mean, whereas MAE ..conditional median 没懂.

**3. Facebook: How to choose K when K-means clustering**

elbow method 算ESS

**4. Easy. Salesforce: How to make models more robust to** outliers

Investigating outliers is the first step. 知道why outliers occurred.

Possible methods to handle:

* Add regularization: reduce variance. Ex. L1, L2
* Different models: can use a model more robust to outliers. Ex. tree-based (random forest, gradient boosting)
* Winsorize data (缩尾处理): cap the data at various arbitrary thresholds. Ex. 90% winsorization, set top and bottom 5% values = 95th and 5th percentile of values
* Transform data:Ex. Log transformation (when the response variable follows an exponential distribution or is right -skewed).
* Change the error metric that is more robust to outliers: Ex. MSE → MAE
* Remove outliers: true anomalies not worth incorporating into the model.

**5. Easy. AQR: Run a multiple linear regression →** predictors are correlated**. Regression的结果会受到什么影响 & 怎么处理**

影响:

1. Coefficient estimates and signs will vary dramatically.
   1. Ex. certain coefficients may have a confidence level that includes 0, meaning it’s difficult to tell whether an increase in X is associated with an ↑ or ↓ in Y, and hence the result will not be statistically significant.
2. Resulting p-value will be misleading.
   1. Ex. an important variable may have high p-value, 然后就被认为是statistically insignificant

处理：

1. Remove: 需要知道cause of correlation, 例如include extraneous predictors such as X and 2X…
2. Combine predictors:
   1. Possible to add interaction terms (X1 \* X2)
   2. Center the data
   3. Obtain a larger size of sample → narrower confidence interview
   4. Regularization

**6. Motivation behind** random forest**. Two ways in which they improve upon individual decision trees.**

Motivation:

* Individual decision trees are usually prone to overfitting → random forest
* Can utilize multiple decision trees and average their decisions
* Used for both classification and regression

Ways of improving upon individual trees:

* 和其他ensemble model一样，using a large set of trees created in a resample of the data (bootstrap aggregation) will lead to a model yielding more consistent result. 更具体一点: 和decision tree比, it leads to diversity in training data for each tree and so contributes to better results in terms of bias-variance trade-off (尤其是variance)
* Using only features at each split helps to de-correlate the decision trees, 避免了很重要的feature永远在第一层 (avoiding the having the very important features always appearing at the first splits)
* Fairly easy to implement & fast to run
* Interpretable feature-importance values → improve model understandability and feature selection

**7. Predict the likelihood of a given transaction being fraudulent, 但很多missing values, 怎么处理**

**8.** Logistic regression**, but find the results to be unsatisfactory. 怎么**Improve**, 或者可以用什么别的model？**

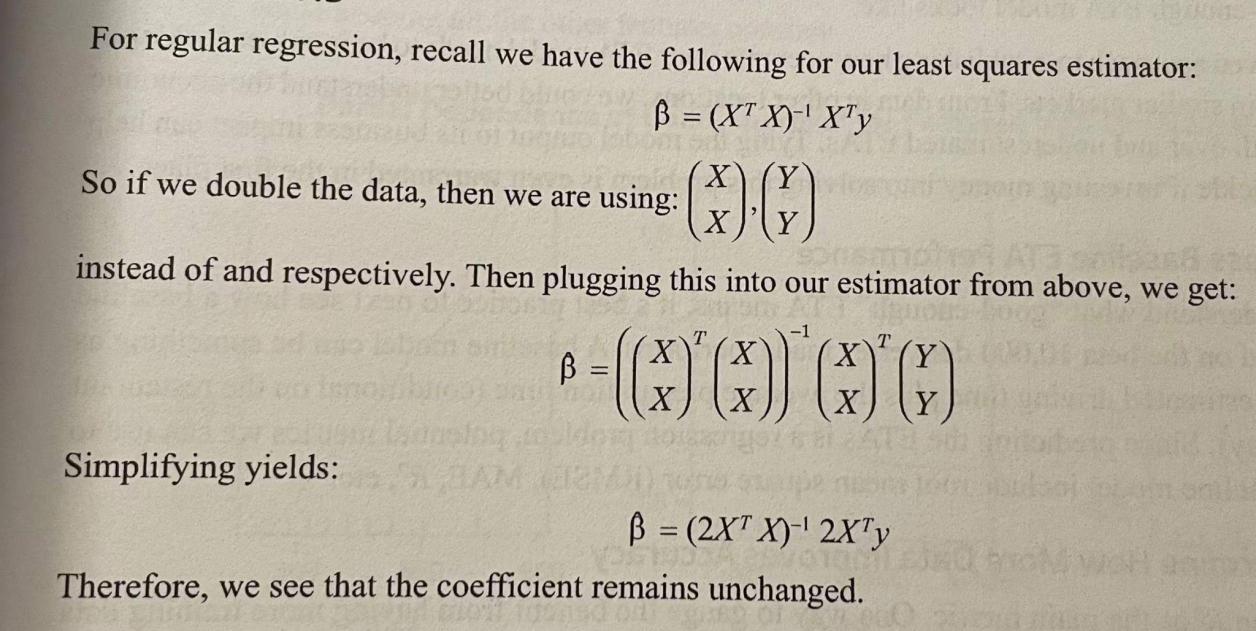
Ways to improve the performance of logistic regression:

* Normalize features: so that particular weights do not dominate within the model
* Add additional features: it may simply be the case that there aren’t enough useful features. 一般如果high bias, 可以加
* Address outliers: identify and decide whether to retain or remove them
* Select variables: 看有无features have introduced too much noise into the process
* Cross Validation & hyperparameter tuning (HT): k-fold along with HT (ex. Introduce a penalty term for regularization purposes should help)

**9. Easy. Two Sigma: running a** linear regression **for a dataset, but 意外地**duplicated every data points**,**  coefficient**会怎样？**

Intuitively，不变

从公式来看：



**10.** Gradient boosting (GB) vs. Random Forest (RF)

Similarity:

* An ensemble of decision trees are used
* Flexible and don’t need much data preprocessing

Difference:

* In GB, trees are built one at a time, such that successive weak learners learn from the mistakes of preceding weak learners. In RF, trees are built independently at the same time.
* Output: GB combines the results of weak learners with each successive iteration; in RF, trees are combined at the end (average or majority)
* In real life: GB excels when used on unbalanced datasets (ex. Fraud detection); RF excels at multi-class object detection with noisy data (ex. Computer vision)
* GB is more prone to overfitting due to their focus on mistakes over training iterations and the lack of independence in tree building.
* GB hyperparameters are harder to tune
* GB may take longer to train because the trees of the latter are built sequentially.

**11. Easy. Doordash: Doordash is launching in Singapore. 为这个新市场 predict the estimated time of arrival (ETA) (time of “order has been placed in app → reach customer”). From an earlier beta test, there were 10,000 deliveries made. 问do you have enough training data to create an accurate ETA model?**

→ “Accurate enough”比较主观，所以最好ask clarifying questions before addressing the lack of training data. 还可以在最后主动提及ways to source more training data.

**Step 1: Clarify what “Good” ETA means**

To determine how accurate the ETA model needs to be

* 首先问what ETA prediction will be used for
  + 例如order-driver matching algorithm需要的ETA accuracy 比 display to customer in app 要高
* Consider if ETA estimates under-promises and over-delivers.
  + 收到的比预期快customer有可能高兴，但也有可能customer觉得送货时间太长就order了
* Data-driven approach: 可以看ETA in similar markets (better understand the economic impact of both over and underestimate ETAs)

**Step 2: Assess Baseline ETA Performance**

See how a baseline model, trained on 10,000 deliveries made, performs.

* 比如baseline model可以是 estimated driving time + avg preparation time (conditional on the restaurant and time of day)
* Regression problem → metric可以是RMSE, MAE, R^2, …

**Step 3: Determine how more data improves accuracy**

假设用R^2

Build learning curves: how accuracy changes when we train a model on a progressively larger percentage of data.

* 如果R^2 improvement 在用50%data和75%data中明显减少了，可能就意味着should reevaluating features rather than simply adding more training data.

**Step 4: In case performance isn’t “Good Enough”**

如果learning curve说明data确实不够，讨论就会转移到dealing with lack of data。

然后也可以提出以下点，如果你认为自己可以。。

* Too few/many features?
* Different models?
* Possible to acquire data in a cost-effective way?

…

## **Medium**

**12. Affirm: a binary classification loan model, rejected applicants must be supplied with a reason why they were rejected. Without digging into the weights of features, how would you supply these reasons?**

**13. Google: Say you are given a very large corpus of words. How would you identify synonyms?**

1. Word2vec can produce vectors for words
2. Measure distance between these vectors, Euclidean distance, cosine similarity
3. Run k-means clustering / k-nearest neighbor

**14. Facebook: What is the bias-variance trade-off? How is it expressed using an equation?**

Total model error = Bias + Variance + Irreducible error

Flexible models low bias, high variance

Rigid models high bias, low variance

High bias: model too simple

High Variance: error occurs when a model overfits data, too much noise, complex neural network

What we want: low bias and low variance. Prevent overfitting and retain sufficient accuracy

**15. Uber: Define the cross-validation process. What is the motivation behind using it?**

Process:

1. Randomly shuffle data into k equally-sized blocks (folds)
2. For each i in fold 1…k, train the model on all the data except for fold i, and evaluate the validation error using block i
3. Average the k validation errors from step 2 to get an estimate of the true error

Motivation:

1. Avoiding training and testing on the same subsets of data points, which leads to overfitting
2. Avoiding using a dedicated validation set, with which no training can be done.

CV works well for smaller datasets

**16. Salesforce: How would you build a lead scoring algorithm to predict whether a prospective company is likely to convert into being an enterprise customer?**

**Step 1: Clarify lead scoring requirements**

* Are we building this for our own company’s sales leads? Or, are we building an extensible version as part of the Salesforce product?
* Are there any business requirements behind the lead scoring( does it need to be easy to explain internally and/or externally?
* Are we running this algorithm only on companies in our sales database, or looking at a larger landscape of all companies?

Assume internally use internal sales data to predict whether prospective company will purchase a Salesforce product

**Step 2: Explain the features you’d use**

* Firmographic Data: What type of company is this? Industry? Amount of revenue? Employee count?
* Marketing Activity: Have they interacted with marketing materials, like clicking on links within email marketing campaigns? Have employees from that company downloaded whitepapers, read case studies, or clicked on ads? If so, how much activity has there been recently?
* Sales Activity: Has the prospective company interacted with sales? How many sales meetings took place, and how recently did the last one take place?
* Deal Details: What products are being bought? Some might be harder to close than others. How many seats(licenses) are being bought? What’s the size of the deal? What’s contract length?

**Step 3: Explain models You’d Use**

Logistic Regression:

Pros: A straightforward solution with easily interpretable result

Cons: 1) cannot capture complex interaction effects between variables

2) Numerically unstable

More complex (Neural network / SVM)

Pros: 1) suit for high-dimensional data

2) capturing the complex interaction

Cons: require a large amount of data to perform well

Tree-based models (random forests / XGBoost)

Pros: features that have the highest influence on predictions are readily perceived

**Step 4: Model Deployment Nuance**

Feature shifts, model degradations

**17. Spotify: How would you approach creating a music recommendation algorithm?**

**Collaborative filtering**

A user-song matrix

1. Employ a binary system to count the number of times a song is streamed and store this count

2. Matrix factorization

songs M users N Matrix R

User preference:

Model: alternating least squares, KNN

**18. Amazon: Define what it means for a function to be** convex**. Example of a ML algorithm that is not convex and why.**

Mathematically, a convex function f satisfies the following:

* For any 2 points x and y in the domain of f:

意思是 straight line between any pair of points on the curve of to be above or just meets the graph.

* If is convex, any local minimum = global minimum

Ex. Neural Network

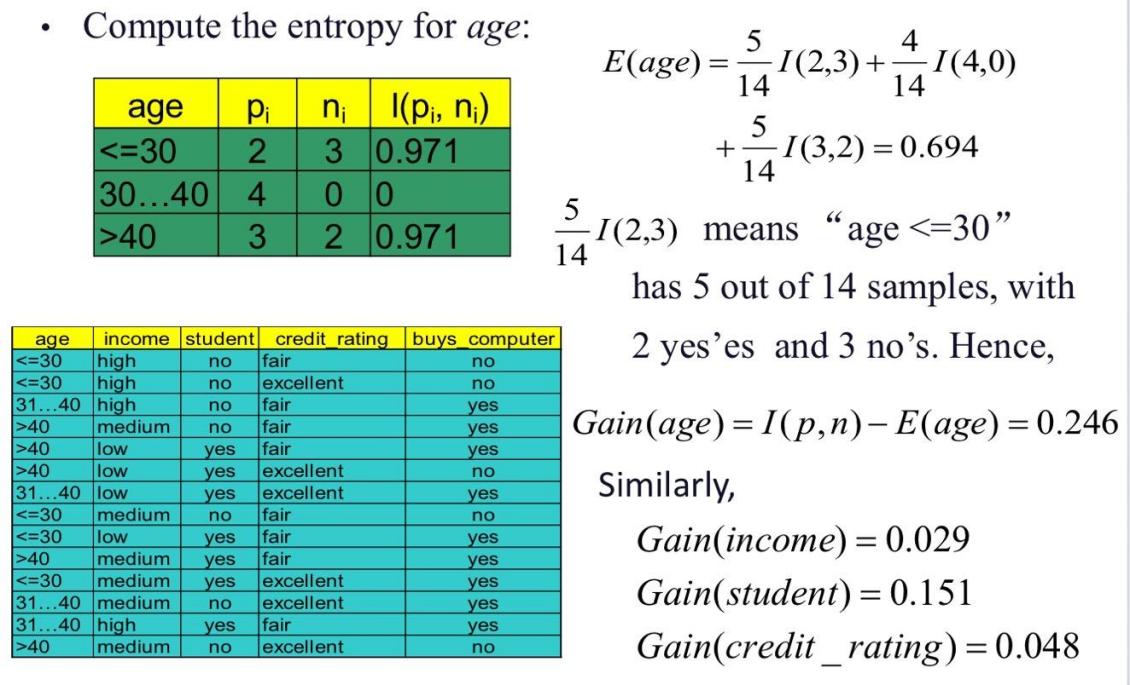
* NN are universal function approximators, meaning that they can approximate any function arbitrarily well. Because not all functions are convex (convex cannot approximate non-convex), they are non-convex.
* Cost function of NN has a number of local minima → no particular global minima

**19. Microsoft: Explain information gain and entropy in the context of a decision tree and walk through a numerical example.**

For a discrete variable Y

High entropy distribution closer to uniform than a skewed one

Information Gain



**20. Uber: L1 and L2, and difference.**

Both are regularization methods that prevent overfitting by coercing the coefficients of a regression model towards 0.

Difference:

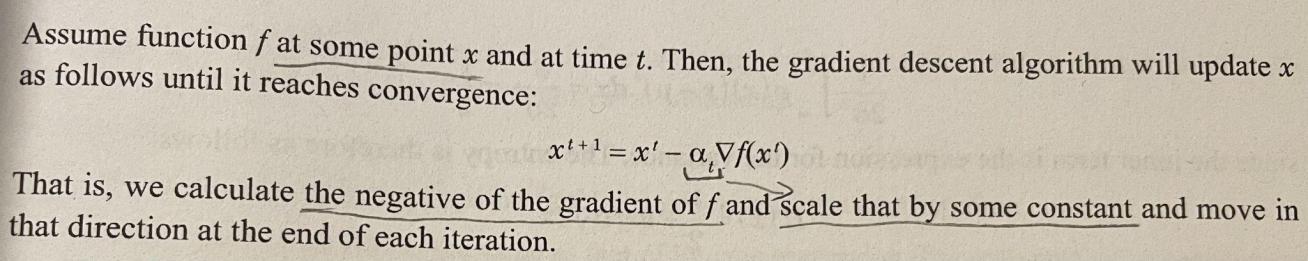
* Form of penalization applied to loss function
* L1 is more likely to “zero” out particular weights → removed certain features → sparser model

**21. Describe gradient descent & motivations behind stochastic gradient descent.**

<https://www.youtube.com/watch?v=sDv4f4s2SB8>

The gradient descent is an optimization algorithm for finding a local minimum of a differentiable function.

* It is simply used in machine learning to find the values of a function's parameters (coefficients) that minimize a cost function.
* It takes small steps in the direction of steepest descent to optimize a particular objective function.
* Step size: 离minimum远的时候大，近的时候小，就是要efficiently找到minimum.
  + Proportional to the negative gradient of the function at the current value of the parameters.



Stochastic gradient descent SGD:

* Done by using only one randomly selected sample at each step to evaluate the derivative of the function
* To be faster and more attractive 当data很多的时候
* Also useful when there is redundancy in data (i.e observations that are similar)

**22. Affirm: a** classifier **predicts probability score [0,1]. 把每个score平方了，**ROC curve怎么变**？如果不变, what functions can change the curve?**

不变，since the relative ordering has been maintained.

* ROC curve plots true positive rate vs. false positive
* If all scores 同时change, none of the actual classifications change (since threshold are adjusted) → same rates → since only the relative ordering of the scores matters.

Any function that is not monotonically increasing would change the ROC curve, 因为relative ordering 会变.

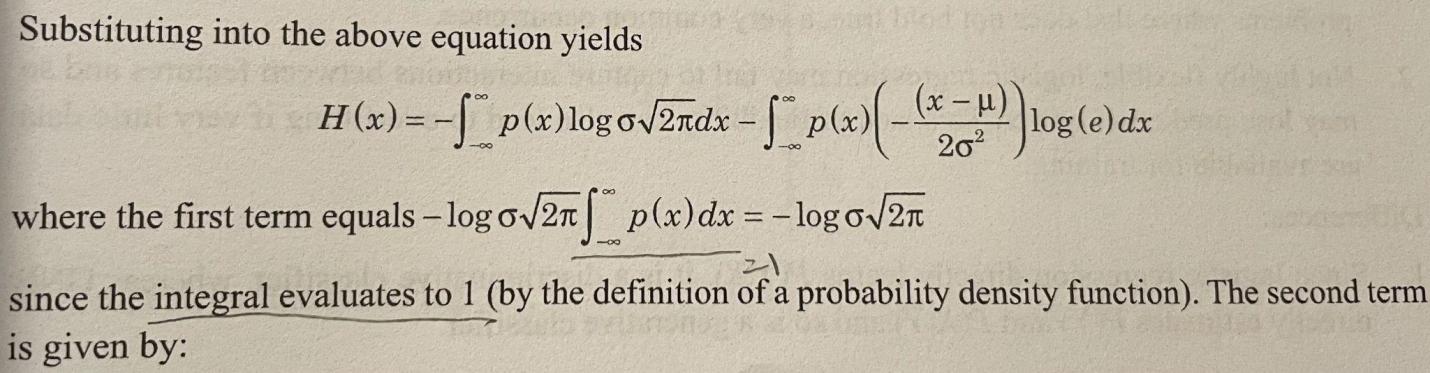
* 例如

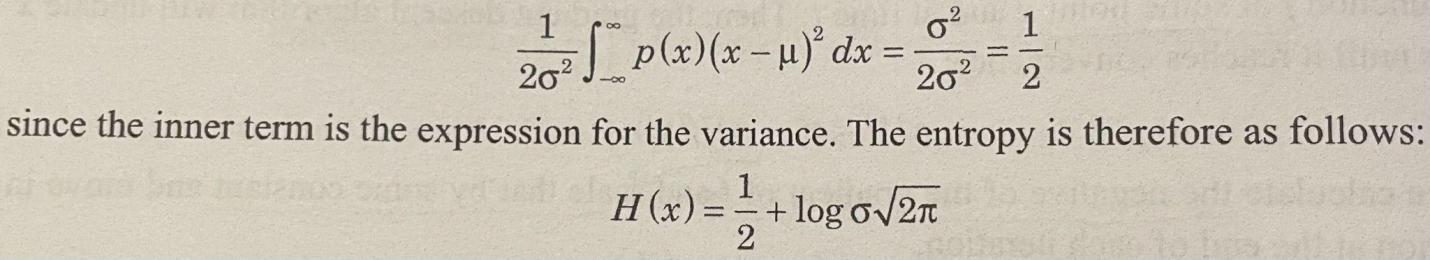
**23. IBM: X is a univariate** Gaussian **random variable.** Entropy of X**.**

Entropy for a continuous random variable is:

For a Gaussian,

把p(x)带入：(这部分计算看不懂了。。）





**24. Stitch Fix: Build a model to calculate a customer’s propensity to buy a particular term. Pros & Cons （简易版case)**

Propensity models are a form of binary classifier.

1. Construct a large dataset
   1. Variable of interest (purchase or not)
   2. Relevant covariates (age, gender, income, etc)
2. Build a model to calculate the probability of purchase of each item

Select model:

* Logistic regression
  + Straightforward solution, easily interpretable result
  + Resulting log-odds is a probability score for purchasing a particular item
  + But, 不能capture complex interaction effects between different variables, 还可能numerically unstable under certain conditions (例如correlated covariates 或 small user base)
* NN or SVM
  + Both are great with high-dimensional data and with capturing the complex interactions
  + 但是不easy to explain, 需要大量data才能perform well
* Tree-based models, Ex. random forest
  + Highly accurate, easily understandable
  + Features which have the high influence on predictions are readily perceived

**25. Gaussian Naive Bayes (GNB) v.s Logistic Regression. When to use one over the other?**

GNB:

* requires a small number of observations to be adequately trained; easy to use; reasonably fast to implement; interpretation of results is highly useful.
* assume features are independent → usually wrongly employed when assumption does not hold true

Logistic regression:

* simple interpretation in terms of class probabilities; can make inferences about features and identify the most relevant one.
* 不能capture interactions between features; lack of flexibility，在data少的时候可能会overfit

Similarities:

* Both are linear decision functions generated from training data.
* GNB’s implied P(Y|X) == that of logistic (but with particular parameters)

Difference:

* Logistic directly learn P(Y|X) → discriminative classifier; GNB directly learns P(Y), P(X|Y) → generative
* Logistic requires an optimization setup (where weights cannot be learned directly through counts), 但GNB不需要

## **Hard**

26. Loss function used in k-means (k clusters, n sample points). Compute the update formula using 1) 2) for the cluster mean for cluster k using a learning rate

1. Batch gradient descent
2. Stochastic gradient descent

27. Kernel trick in SVMs; example. How to decide what kernel to choose?

28. N observations, from some variables (gaussian distribution). Best guesses for the parameters of the distribution.

29. Use GMM for anomaly detection of fraudulent transactions, to classify incoming transactions into K classes.

1. Model setup formulaically
2. Evaluate the posterior probabilities & log likelihood
3. How to determine if a new transaction should be deemed fraudulent

**30. Build a model to predict whether a particular Robinhood user will churn?**

Churn: the process of a platform’s loss of users over time

**Step 1: Clarify what churn is and why important**

understand how Robinhood monetizes

Trading activity: Cancellation membership / long period no activity → churn

User’s’ account balance: negligible account balance → churn users

Churn is important because it’s more expensive to acquire new users than to retain existing ones, high churn means more financial resources to support new customer acquisition.

**Step 2: Modeling Considerations**

Factor 1：更看重probability of the customer’s loss还是whether the customer will be lost or not

Factor 2: model explainability

Interpretable models: logistic regression, decision trees, random forests

Less interpretable models: neural networks, SVM

**Step 3: Features we’d use to model churn**

Raw account balance

* Account balance trend
* Experienced heavy looses
* Recent usage patterns
* User demographics

**Step 4: Deploying the churn model**

* Monitor model performance, adjust features as necessary whenever there is new data/ feedback from customer-facing team (prevent model degradation),
* Error analysis, keep refining model
* A/B test to validate its impact

**31. Two Sigma:** linear regression**, model the** error terms as being normally distributed**. Show maximizing the likelihood of the data is equivalent to minimizing the sum of squared residuals (MLE = MSE)**

**32. PCA, formulation and derivation in matrix form.**

**33. Logistic regression, model formulation. How to maximize the log-likelihood of a given model (two-class case)**

**34. 怎么做music recommendation algorithm for Discover Weekly (a 30-song weekly playlist personalized to an individual user)**

Step 1: Clarify details of Discover Weekly

* Goal of algorithm
  + 推荐新歌，push their musical boundaries?
  + 喜好相近的
  + Generally, how we think about the trade-off of exploration vs. exploitation
* Recommend songs, podcasts?
* Various service-level agreements?
  + 如果user不听，playlist需要每周该吗
* New user要推吗

Step 2: Use what data features

* User-song interactions (strongest signal for whether or not they enjoy a song)
  + 和movie recommendation相似，但有点不同
  + 例如repeated consumption, 听歌会听很多遍
  + Music 有wider variety(i.e niche music)
  + Scale of music catalog >> movies
* Metadata of song
  + 例如artist, album, playlists
  + Audio features in the songs, 例如tempo, speechiness, instruments used
* Demographic info
  + 例如age, gender, location

Step 3: Explain Collaborative Filtering Model Setup

Recommendation一般有2种：content-based filtering & collaborative filtering

Collaborative filtering uses data from feedback users have provided on certain items (songs), in order to make recommendations.

* Dataset: User-song matrix, play count = # times a user streamed the song (类似于movie rating)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Song 1 | Song 2 | … |
| User 1 |  | Play count | … |
| User 2 | Play count | Play count | … |
| … | … | … | … |

* The output is a latent user matrix and a song matrix.
* Use vectors from these matrices, 计算dot product (denotes the relevance of a particular song a particular user)
* 也可以用这些vectors来assess similarity between users and different songs using KNN
* Sort by relevance scores on songs that user has not yet streamed → recommend!

Step 4: Additional considerations

Pros & Cons of collaborative:

* Run it in a scalable manner to find correlations behind user-song interactions.
* “Cold start” problem: an existing base of data is needed for any given user

Scale:

* Spotify has hundreds of millions of users; the Discover Weekly could be updated in batches for various users at different times to help speed up data processing and model training.

Dynamic nature:

* Influx of new users and songs, along with the fast-changing music trends, would necessitate constant retraining

Measure and Track the impact over time:

* Collaborative filtering doesn’t come with clear metrics of performance.
* 可以用A/B test to find users with the improved recommendations had increased engagement on the platform (例如他们的time spent listening)

**35. 推导variance-covariance matrix of the least squares parameter estimates in matrix form.**