Brandon's Grand Unifying Data Science Map of Everything

Graph Data Science
Unsupervised Graph Algorithms
Supervised Machine Learning
Which parts of my graph are connected to each other?
Which nodes are most similar?
Where will connections form next?
What's the label for this node?
Clustering
- Community Detection

Association
- Similarity
- Pathfinding

Link Prediction
Node Classification
How important is each node?
What's the missing property value?
Dimension Reduction (generalization)
- Centrality
- Embeddings

Property Regression

What can we do with a mapped network?

Graph Algorithms

Pathfinding & Search
- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality & Importance
- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)

Community Detection
- Triangle Count
- K-Means
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation

Supervised Machine Learning
- Node Classification
- Link Prediction
- Node Regression

Heuristic Link Prediction
- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity
- Node Similarity
- K-Nearest Neighbors (KNN)
- Jaccard Similarity
- Cosine Similarity
- Pearson Similarity
- Euclidean Distance
- Approximate Nearest Neighbors (ANN)

Graph Embeddings
- Node2Vec
- FastRP
- GraphSAGE

... and more!
- Synthetic Graph Generation
- Scale Properties
- Collapse Paths
- One Hot Encoding
- Split Relationships
- Graph Export
- Pregel API (write your own algos)

Neo4j, Inc. All rights reserved 2022

```python
import geopandas as gpd
from sklearn.cluster import KMeans

# Load spatial data (GeoDataFrame)
gdf = gpd.read_file('path_to_your_data.geojson')

# Extract spatial coordinates
X = gdf[['longitude', 'latitude']].values

# Apply Spatial K-Means clustering
kmeans = KMeans(n_clusters=3)
labels = kmeans.fit_predict(X)

# Add cluster labels to the GeoDataFrame
gdf['cluster'] = labels

# Visualize spatial clusters
gdf.plot(column='cluster', cmap='viridis', legend=True)
```

Follow same steps for clustering
Construct spatial weights

Local Moran's I
Getis-Ord Gi* Hot Spot
Determine Model Inputs
Run Statistical Tool
Analyze results

Anselin Local Moran's I: Measures spatial autocorrelation to identify clusters of similar or dissimilar values.

Spatial Autoregressive Model (SAR)
Spatial Error Model (SEM)
Spatial Lag Mode
Estimate Model Params PySAL package
Evaluate Fit (R-squared, AIC, BIC)
Check for Spatial autocorrelation in Residuals
Predict
Spatial data mining
Space-Time Analysis
Point
LineString
Polygon
Multi-point
Multi-linestring
Multi-polygon
Geometry Collection

Contains Outliers and Anomolies?
Spline interpolation models
Filling continuous value? (price)
Kriging, Radial Basis Functions (RBF)
Fill discrete value? (low vs high income)
Inverse Distance Weighting (IDW) model
Fit Model
Validate - Mean Absolute Error (MAE) or Root Mean Square Error (RMSE)
Conduct Sensitivity analysis to check each approach if necessary

Network Science
Fantastic book that is open source at: http://networksciencebook.com/ by Albert Laszlo Barabasi (the GOAT)
Define Network/ Graph Shape
Directed (link initiated by one node)
Undirected (link is not initiated by one node)
Weighted (link has some value - e.g. number of text messages shared)
Unweighted (link is simple - e.g. brandon knows tom)
Degree (number of links a node has to another node - represented as "K"
Distribution of K
How to track connections?
Analyses
Adjacency matrix
Reduce dimensionality through Matrix Factorization
Connectedness
A *network is connected* if all pairs of nodes in the network are connected. A *network* is *disconnected* if there is at least one pair with $d_{ij} = \infty$
Shortest Path (between two nodes in the network)
Clustering Coefficient
captures the degree to which the neighbors of a given node link to each other. For a node *i* with degree k_i the *local clustering coefficient* is defined as
$C_i = \frac{2L_i}{k_i(k_i-1)}$
Average Path Length
How do we capture randomness? How to model reality?
Random Network Model
Betweenness
Centrality
Closeness
Eigenvector Centrality
Katz Centrality
PageRank
The Largest Hub
Evolving or Static?
Attractiveness measures for growing networks
Barabasi-Albert Model
Bianconi-Barabasi Model
Measuring Spread
Degree Correlation

Phenomena	Agent	Network
Venereal Disease	Pathogens	Sexual Network
Rumor Spreading	Information, Memes	Communication Network
Diffusion of Innovations	Ideas, Knowledge	Communication Network

I think the average student is 64 inches tall.
Uo = 64 ← this is the null
Ua > Uo ← this is the alternative

1) The Large-Sample z Test

What is It? A test of a distribution mean μ. The distribution itself is unspecified. (If the distribution is normal, use a t test. See below.)

What is Assumed? The test utilizes the Central Limit Theorem, so the sample size is $n \geq 30$. And all data are sampled from the same distribution, i.e., all data are *independent and identically distributed*, or *iid*.

What is the Test Statistic? The standardized quantity

$$Z = \frac{\bar{X} - \mu_o}{S/\sqrt{n}}.$$

2) The One-Sample t Test

What is It? A test of a *normal* distribution mean μ

What is Assumed? All data are iid samples from a normal distribution of unknown mean and standard deviation.

What is the Test Statistic? The standardized quantity

$$T = \frac{\bar{X} - \mu_o}{S/\sqrt{n}}.$$

3) The Two-Sample t Test

What is It? A test of whether the data in two separate samples were drawn from normal distributions with the same mean μ.

What is Assumed? The data in each sample are normal and are independent and identically distributed. Note that the data may be paired...there may be a one-to-one correspondance between each datum in one sample and each in the other. (This would necessitate a paired t test.)

What is the Test Statistic? Now things start getting a bit complicated. For a classic unpaired two-sample t test, see the formulae on, e.g., this web page.

4) One-Way Analysis of Variance (ANOVA)

What is It? A test of whether the data in k separate samples were drawn from normal distributions with the same mean μ. (It is essentially a k-sample t test.)

What is Assumed? The data in each sample are normal and are independent and identically distributed.

What is the Test Statistic? See this web page.

What is the p-Value? It is output by the `anova()` function in R. See today's lab for more details.

(Wait...is this really useful? It can tell me that one of the means is different...but which one?)

We are glad you asked. You are correct. A standard analysis "protocol" is first to apply ANOVA to see if there is at least one mean that is statistically significantly different from the others (i.e., if $p < \alpha$), then to apply a *post-hoc comparisons test* (like the so-called Tukey test) to see which pairs of means are different. As life is short (as already stated), we will not cover post-hoc tests here.

Name	Type	Parameters	Domain (i.e., allowed x values)
Binomial	discrete	trials n, probability of success p	integers 0 to n
Poisson	discrete	expected number of counts λ	all integers ≥ 0
Normal	continuous	mean μ, standard deviation σ	$(-\infty, \infty)$
Standard Normal	continuous	none ($\mu = 0$ and $\sigma = 1$)	$(-\infty, \infty)$
t	continuous	number of degrees of freedom ν	$(-\infty, \infty)$
Exponential	continuous	event rate β	$(0, \infty)$
Chi-Square	continuous	number of degrees of freedom ν	$(0, \infty)$

Distribution Shape

Mean (Average):
- The arithmetic mean, calculated as the sum of all data points divided by the number of data points.
- Formula: Mean $= \frac{1}{n}\sum_{i=1}^{n} x_i$

Median:
- The middle value in a dataset when it is ordered.
- For an odd number of data points, it's the middle value; for an even number, it's the average of the two middle values.

Mode:
- The value(s) that appear most frequently in the dataset. A dataset may have one mode (unimodal), more than one mode (multimodal), or no mode.

Range:
- The difference between the maximum and minimum values in the dataset, indicating the spread of data.
- Formula: Range = Max − Min

Variance:
- A measure of the spread or dispersion of data points around the mean.
- Formula for population variance: $\sigma^2 = \ldots$
- Formula for sample variance: $s^2 = \frac{1}{n-1}\ldots$

Standard Deviation:
- The square root of the variance, representing the average distance of data points from the mean.
- Formula: Standard Deviation $= \sqrt{\ldots}$

Interquartile Range (IQR):
- The range between the first quartile (25th percentile) and the third quartile (75th percentile). It measures the spread of the middle 50% of data points and is resistant to outliers.

Skewness:
- A measure of the asymmetry of the data distribution. Positive skewness indicates a longer tail on the right, while negative skewness indicates a longer tail on the left.

Kurtosis:
- A measure of the "tailedness" of the data distribution. It quantifies whether the data distribution has heavier or lighter tails than a normal distribution.

Correlation Coefficient (Pearson's r):
- A measure of the linear relationship between two variables. It ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear correlation.

$$WSS = \sum_{i=1}^{n}\sum_{j=1}^{k} ||x_i - c_j||^2$$

Optimal number of clusters
Elbow method
Total Within Sum of Square
Number of clusters k

Optimal number of clusters
We're looking for the highest value. That is 2.

Generally, you should use odd values of k to avoid ties. Ties can occur when the same number of neighbors from each class are present, which can complicate the decision.
Odds vs Evens
Evaluate the performance of the model with different values of k.
K Fold Cross Validation
Perform a search over a range of k values and evaluate the model's performance for each value. Choose the k that gives the best results on the validation set
Grid Search
Apply K-means over range of k-values from k=1 to k=max. For each k, calculate the Sum of Squares (distance between each data point and assigned cluster - see WSS). Plot (left image) and use your best judgement
Elbow Method
Pass an argument for the gap statistic and look for the highest value amongst k
Gap Statistic
What is the right number of K?
K- Means Cluster
How do I measure distance?
Euclidean: Simple, good for continuous, bad for categorical. Bad w/ outliers and scale
Manhattan: good for categorical, ignores diagonal movements
Mahalanobis: consides covariance, good for data on diff scales and correlations, intensive
Clustering
- Hierarchical Clustering
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- Agglomerative Clustering
- Mean Shift Clustering
- Affinity Propagation
- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
-OPTICS (Ordering Points To Identify the Clustering Structure)
- Kernel K-Means

Text Mining / NLP
Clean text: remove symbols, special characters, punctuation, capital letters, and stop words
Market basket analysis - which items are purchased together
Is this a good predictor?
Support: Indicates the frequency of occurrence of an itemset in the dataset.
Confidence: Measures the likelihood that an associated item (or items) will be bought when a certain item is bought.
Lift: Compares the likelihood of a rule being true to the likelihood of it being true by chance.

Antecedent: {bread, butter}
Consequent: {milk}
Support: 0.2 (20% of transactions contain both bread, butter, and milk)
Confidence: 0.8 (80% of transactions that contain bread and butter also contain milk)
Lift: 1.33 (Buying bread and butter makes it 1.33 times more likely to also buy milk)

Sentiment Analysis (can be supervised but not always)
Text/Document Classification
Entity Extraction
Entity Relation Modeling
Pre-trained Models
-roBERTa
-HuggingFAce
Summarization

Multiple Possible Categories [Risk: High, Medium, Low]
One vs Rest
ONE V REST - Support Vector Machine
ONE V REST - Logistic Regression
ONE V REST - Naive Bayes
Decision Tree
Random Forest
K-Nearest Neighbor
Gradient Boosting
Logistic Regression
Truly Distinguish

Build a spark session
Spark Streaming
ML Lib (make model)

```python
# Perform an action to collect results
squared_data = squared_rdd.collect()

# Stop SparkContext
sc.stop()

# Print the results
print(squared_data)
```

Where can I even use this?
Deep Learning

Type of Machine Learning

	Unsupervised Learning		Supervised Learning		Reinforcement Learning
Objective	Group and interpret data based only on input data		Develop predictive model based on both input and output data		
Category of Algorithms	Dimensionality Reduction	Clustering	Classification	Regression / Prediction	Prediction
Algorithm	Singular value decomposition	k-means	Support vector machines		Dynamic programming
	Principle Components	Fuzzy C-means	Naïve Bayes		Temporal-difference learning
	Linear Factor Analysis	Hierarchical	Nearest Neighbor		Q-learning
	Independent components	Gaussian mixture	Discriminant analysis		SARSA
	Non-negative matrix factorization	Hidden Markov/Markov Chains			Policy gradient
	Proper orthogonal decomposition	Decision Trees			Evolutionary programming
	Deep (Neural) Networks				
			Linear regression		Genetic programming
			Nonlinear regression		

Decision Tree
Random Forest
Gradient Boosting
Support Vector Machine
K-Nearest Neighbor
Regression? See Inference Section

Regression	Classification	Recommender System
Mean Absolute Error (MAE) Root Mean Squared Error (RMSE) R-Squared and Adjusted R-Squared	Recall Precision F1-Score Accuracy Area Under the Curve (AUC)	Mean Reciprocal Rank Root Mean Squared Error (RMSE)

Youden's J Statistic: attempts to find the threshold where the sum of the correct classifications for both classes are maximized. Sacrifices accuracy in the dominant class to do well in both.

By introducing ROC curves and the concept of AUC, we now have a new metric for model selection. But ultimately, we do need to choose *one* threshold value so as to generate predictions. We can use the ROC curve to do this, while acknowledging that there is no unique means to choose that one value.

An often-used choice is *Youden's J* statistic:

$$J = \text{sensitivity} + \text{specificity} - 1.$$

Basically, we want to determine the threshold value that allows us to best identify NON-CBs and CBs at the same time. For our example:

```r
J = roc.log$sensitivities + roc.log$specificities - 1
w = which.max(J)
cat("Optimum threshold for logistic regression: ",round(roc.log$thresholds[w],3),"\n")
```

```
## Optimum threshold for logistic regression:  0.488
```

The value is close to 0.5, which is not surprising because our data had balanced classes (i.e., just as many CBs as NON-CBs). When the data feature unbalanced classes, optimal thresholds may differ greatly from 0.5.

Problem Statement
Binary Classification (A,B = 0.5)
Class Labels
Probability
Metrics
1. ROC Curve
2. AUC Curve
3. PR Curve
Balanced Dataset
Imbalanced Dataset
Accuracy= $\frac{TP + TN}{TP + FP + FN + TN}$
1. Precision
2. Recall
3. F Beta
Recall = $\frac{TP}{TP + FN}$
Precision= $\frac{TP}{TP + FP}$

Confusion Matrix
Actual Value
Predicted Value
Type I Error / FPR = $\frac{FP}{FP + TN}$
Actual Results
Type II Error
Always have to focus on reducing the Type I Error and Type II Error

Recall (TPR) : Out of the total positive actual value, how many values did we correctly predicted positively. Whenever the FN is important wrt the problem statement, use Recall.

Precision : Out of the total predicted positive results, how many results were actual positive. Whenever the FP is important wrt the problem statement, then use Precision

In some of the problem statements FP and FN both are equally important in an imbalanced dataset.In such cases we have to consider both Recall and Precision. In such cases we have to use F-beta score

F-Beta = $\frac{(1+B^2)\ \text{Precision X Recall}}{B^2 \times (\text{Precision + Recall})}$

1. If we have a problem statement where the FP and FN are both equally important, we select Beta = 1
2. In scenarios where the FP is having greater impact than the FN, in such case we reduce the Beta value between 0 to 1
3. In scenarios where the FN is having greater impact than the FP, in such case we increase the Beta value greater than 1 and might range between 1 to 10

A good suite of models to explore, in order from least flexible/most inferential to most flexible/least inferential would be
- regression (quantitative response): linear regression, bestglm, regression tree, random forest, extreme gradient boosting, K-nearest neighbors (large amount of data), deep learning (huge amount of data)...pick the model with the lowest test-set mean-squared error
- classification (factor response, two outcomes): logistic regression, bestglm, classification tree, random forest, extreme gradient boosting, K-nearest neighbors (large amount of data), support vector machines (small amount of data), deep learning (huge amounts of data)...pick the model with...well...this is more complicated

With few exceptions, Machine Learning algorithms don't perform well when the input numerical attributes have very different scales. This is the case for the housing data: the total number of rooms ranges from about 6 to 39,320, while the median incomes only range from 0 to 15. Note that scaling the target values is generally not required.

Have text or categorical data?
Feature Scaling
Select and Train Models
Evaluate Models based on Metric
Shortlist the best performing models
Make a table of results and select the best model
Continuously test model against new observations

Predict [How much will this car cost when it sells?]
Column Mean or Median (or even Mode)
- MCAR: Missing completely at random.
- MAR: Missing at random, or
- MNAR: Missing not at random.

Inspect data and check for correlations
Split Data into Training and Test Set
Clean data, fill or drop missing values
Imputation Options
**Warning: Make sure you split the data BEFORE imputation to avoid bias and data leakage. Set the random seed, save the test set in a variable, or better yet use each UID.
Have text or categorical data?
Highly Correlated Features? You may have Multicollinearity.
① Check for redundancy in the correlation matrix -> consider removing one of them or combining them
② Try forward selection, backward elimination, or recursive feature elimination to select only the most important features
③ Do a Principal Component Analysis to reduce dimensionality. PCA creates linear combinations of features, potentially reducing multicollinearity.
④ Limit yourself to only regularization methods for regression (Lasso L1 or Ridge L2)
Change the decision threshold probability in model
Imbalanced Classes
Have lots of data?
Undersampling
Have moderate amount of data?
Oversampling
SMOTE: Synthetic Minority Over-sampling Technique (creates new examples by interpolating between existing examples of the minority class).
Hierarchical Categories (Good, Average, Bad) --> Ordinal Encoder
Regular Categories (Ocean-Front, Mountain, Plains) --> One Hot Encoder
Have hundreds or more categories?
Implement Sparse Matrix (SciPy rather than Numpy Array)
See Text Mining Under Detect
Topic Modelling
When should I sample in general?
Every element of p has equal chance of being affilicted?
Simple Random Sample
Systemic (every 15th person)
Population has natural and meaningful strata (groups)?
Stratified random sample
Too hard to get a complete list of all elements (e.g. US population)?
Cluster Sampling (e.g. a city/ a school)
Hard to reach subjects? Hidden population?
Snowballing Sampling (ask a few and then ask for referrals of others who are the same)

Infer [how can I be sure that my city's car policy is having an effect?]
Casual Inference
Instrumental Variables

Attempting to estimate the causal effect of variable X ("explanatory variable") on Y ("dependent variable"), an instrument is a third variable Z affects Y only through its effect on X.

Scientist wants to estimate the effect of smoking (X) on health (Y). Correlation between smoking and population health does not necessarily imply smoking CAUSES bad health (could be because of drinking or depression or being illiterate or extra terrestrial contact). It is impossible to conduct a controlled lab experiment. So the scientist estimates the effect of smoking on health from observation data using the tax rate for tobacco products as an instrumental variable (Z). The tax rate for tobacco products is a reasonable choice for an instrument because the researcher assumes that it can only be correlated with health through its effect on smoking. If the researcher then finds tobacco taxes and state of health to be correlated, this may be viewed as evidence that smoking causes changes in health.

Ordinary Least Squares
Regression
Check Assumptions
Assumptions are never perfectly met, but they should be reasonable
Linear: A change in the independent variable is proportional to the change in the response variable
Not sure if it is linear?
1. Create scatter plots of the independent variable(s) against the dependent variable. Look for a roughly linear relationship. Points should cluster around a straight line
2. Plot the residuals (the differences between the observed and predicted values) against the fitted values (the predicted values). The plot should not show a clear pattern; it should be evenly distributed around the horizontal line at zero.
Homoscedascity: the variance among errors is constant; consistent noise among the data
Spread of the residuals not consistent?
Breusch-Pagan Test
This is a statistical test that checks for heteroscedasticity in the residuals. A significant p-value suggests that heteroscedasticity may be present
Independent Errors: the residual (errors) for one observation should not provide info about error of another observation
No auto-correlation: The observations are independent of each other (time series data)
Think you may have auto correlation?
Durbin Watson Test
Takes values between 0 and 4. A value around 2 indicates no autocorrelation. Values significantly different from 2 may suggest autocorrelation.
Plot Residuals vs Time
Plot the residuals against the order or time at which they were observed. There should be no discernible pattern or trend in the plot, indicating independence.
Ljung-Box Test
Tests for significance for autocorrelation at lags up to a specified number. A low p-value suggests the presence of autocorrelation.
Do you have access to the same data at specific and regular intervals?
There is a subtle difference between the error and the residual (but to know the true error, you'd need to know the true line not the estimated fit)
Normal Errors: Errors are normally distributed
Errors not following a normal distribution?
Quantile Quantile Plot or Probability Plot (Q-Q Plot)
Plot the standardized residuals against theoretical quantiles from a normal distribution. If the points closely follow a straight line, it suggests that the residuals are normally distributed.
Shapiro-Wilk Test
Test for normality: The null hypothesis is that the errors are normally distributed. A high p-value suggests that the data does not significantly deviate from normality.
No multicollinearity: There should be no high correlation among independent variables.
No perfect colinearity: there should not be a perfect linear relationship among some or all of the predictors.
Create correlation matrix as you normally do
Seeing high numbers?
VIF (Variance Inflation Factor)
Measures how much the variance of an estimated regression coefficient increases when your predictors are correlated. A high VIF (typically > 10) suggests multicollinearity.
Auxiliary regression
Run one X on all the other X's and test hypothesis with formula
Test statistic: F=[R2/(k-1)] / [(1-R2)/(n-k)]

Figure 4.1 *Residuals vs. fitted*

Heteroscedastic Robust Standard Errors: technique to obtain unbiased standard errors of OLS coefficients under heteroscedasticity
Weighted Least Squares (WLS): assigns different weights to observations based on their estimated variances.

Have a treatment and control group?
Identify Treatment and control group
Select pre and post treatment periods
Ensure Parallel Trends Assumption

The Diff-in-Diff model is typically estimated using a regression framework. The model takes the following form:

$$Y_{it} = \beta_0 + \beta_1 \cdot Treatment_i + \beta_2 \cdot Post_t + \beta_3 \cdot (Treatment_i \times Post_t) + \epsilon_{it}$$

- Y_{it} is the outcome variable for individual i at time t.
- $Treatment_i$ is a binary variable indicating whether individual i belongs to the treatment group.
- $Post_t$ is a binary variable indicating whether time t is after the treatment.
- ϵ_{it} is the error term.

- **Treatment Group**: Individuals who participated in the job training program.
- **Control Group**: Individuals who did not participate in the program.
- **Pre-Treatment Period**: Year 1 (Before program implementation)
- **Post-Treatment Period**: Year 2 (After program implementation)

$$Earnings_{it} = \beta_0 + \beta_1 \cdot Treatment_i + \beta_2 \cdot Post_t + \beta_3 \cdot (Treatment_i \times Post_t) + \epsilon_{it}$$

```
| Year | Group     | Earnings |
|------|-----------|----------|
| 1    | Treatment | $30,000  |
| 1    | Control   | $28,000  |
| 2    | Treatment | $32,000  |
| 2    | Control   | $29,000  |
```

1. **Intercept (β0)**: This represents the expected outcome (earnings) for the control group in the pre-treatment period (Year 1).
2. **Coefficient for Treatment (β1)**: … control group in the pre-treatment period … statistically significant, it suggests … control group in the pre-treatment period.
3. **Coefficient for Post-Treatment Period (β2)**: … pre-treatment period (Year 1) … statistically significant, it suggests … from Year 1 to Year 2.
4. **Coefficient for Treatment × Post-Treatment Period Interaction (β3)**: … It represents the additional change … program (Year 2). If β3 is positive … greater increase in earnings …

Interpreting Coefficients with the Example Data

Assuming that all coefficients are …
- If β1 is positive, it means that … control group.
- If β2 is positive, it means that … treatment.
- If β3 is positive, it means that … group after the training program …

Overall, a positive and statistically … treatment group compared to the control group.

Best GLM
BSS output is:

$$C_p = \frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2)$$
$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2k\hat{\sigma}^2) = \frac{C_p}{\hat{\sigma}^2}$$
$$\text{BIC} = \frac{1}{n}(\text{RSS} + \log(n)k\hat{\sigma}^2)$$

Worried about underfitting?
AIC: you can be confident that your selected variables include all of the important ones, but the final list might include a handful of unimportant ones as well. Tends to overfit
Worried about overfitting?
BIC: you can be confident that every selected variables are important but other important variables might be left out of the final list. Tends to underfit

… print a bunch of stuff just ask for the best combo of arguments....

```
…_search.best_estimator_
…restRegressor(bootstrap=True, criterion='mse', max_depth=None,
      max_features=8, max_leaf_nodes=None, min_impurity_decrease=0.0,
      min_impurity_split=None, min_samples_leaf=1,
      min_samples_split=2, min_weight_fraction_leaf=0.0,
      n_estimators=30, n_jobs=None, oob_score=False, random_state=None,
      verbose=0, warm_start=False)
```

6. Reciprocal: $Y_i = B_1 + B_2\,1/X_i$ … $-B_2 * 1/XY$
7. Log Reciprocal: $\ln Y_i = B_1 + B_2\,1/X_i$ … $B_2 * 1/X$
8. Interaction: $Y_i = B_1 + B_2 X_{2i}$ … $(B_2+B_4Z)*X/Y$

if you have a linear regression model predicting sales (Y) based on advertising spending (X), and the coefficient for X is 0.5, it means that a one-unit increase in advertising spending is associated with, on average, a 0.5 unit increase in sales, assuming all other factors remain constant.

if you have a log-linear regression model, a one-unit increase in the independent variable is associated with a percentage change in the dependent variable. Specifically, if the coefficient for the log-transformed X is 0.1, it means that a 1% increase in X is associated with, on average, a 0.1 unit increase in the dependent variable, assuming all other factors remain constant.

if you have a log-log model, a one percent increase in the independent variable is associated with a β percent change in the dependent variable, where β is the coefficient. Specifically, if the coefficient for the log-transformed X is 0.2, it means that a 1% increase in X is associated with a 0.2% increase in the dependent variable, assuming all other factors remain constant.

Survival Analysis
Do we need to act before observing all events?
Does time-to-event matter?
Survival analysis
Regression
Binary classifier

Transforming data for survival analysis requires deciding the censored data. Consider a loan: Censoring occurs when the loan is under repayment at the moment of data gathering. The origin of all loans (the birth event) is known, but for some loans, a default has not yet occurred (the death event). These people are still in the process and might repay the loan or default. These people are "censored."

Transform Data
Event Log Table (row is transaction)
Duration Table (row is customer)
Need to predict if/ when a single subject will survive or die?
Train_Test_Split
Hazard Curve
Cox Partial Hazard fitter (CoxPH) - e.g.: duration_col = 'duration', event_col = 'has_transaction', formula = 'age + entry_month'
What output do we need? (scikit-survival)
Survival Curves for specific UID
Median/Mean Survival time for each UID
Note that it is possible to get a median that is infinitiy
The point at which that UID crosses the 50% (how many days until that UID crosses the coin flip threshold)
Rank Order (survivor time) for each UID
Kaplan-Meier Survival fitter (lifelines py-package)
Need to forecast a new survival curve (say a new month of sales data)?
Weibull fitter (lifelines py-package)
Are two survival curves the same or different?
logrank test (simple hypothesis test)
It is 70% likely that a filling will last longer than 5 years
Survival rate
Time

LONG FORMAT DATA: It contains values that repeat in the first column. In this format, each row is a one-time point per subject.

WIDE FORMAT DATA: In the Wide Format Data, the data's repeated responses will be in a single row, and each response can be recorded in separate columns.

Fill missing data:
Last Obs Carried Forward
Next Obs Carried Backward
Rolling Statistic
Interpolate
We never drop so (fill_value=0)

Time Series Forecasting
Generative Adversarial Networks (GANs)
Variational Autoencoders (VAEs)

Set Index: Pandas DatetimeIndex - dt_index = pd.date_range(start='2023-09-01', periods=3, freq='D')
https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html#timeseries-offset-aliases; note may need time zone as well
Handle business days with pandas_market_calendars
Resample at desired frequency like upsample/ downsample: e.g. sales_new.resample('W').sum() is total for week
Simple Weighted/ Exponential Moving Average to get the next datum
Actual Forecasting
Visualize and check for issues
Check Non-stationarity
Stationarity requires:
- constant mean
- constant variance
- constant autocorre
-no peridoic comp
Conduct a Augmented Dickey-Fuller (ADF) Test. low p-value means non-stationarity
Hetero-scedascity
Plot the residuals and look for horn and funnel shapes
Autocorrelation
Generate autocorrelation plot (ACF plot) and Partial ACF plot
What can we do to fix this?
Remove the Trend and Seasonality with StatsModel
Remove Changing Variance with Log Transformation
Remove Auto-correlation with Differencing
Train_test_split: temporal splitting can be a sliding window or an expanding window
Note that some straight up ML options can still be used: e.g. xgboost, rather than adopting one of these forecasting specific models
Good tutorial here in two parts: https://youtu.be/vV12dGe_Fho?si=FyhCWAtJ_aV9eXX1
Remove Trend and Seasonality; generate ACF and PACF Plots to figure out what model you need
Check the RMSE and MSE for the training and test set (and validation if present)

Model	AR(p) Autoregressive	MA(q) Moving Average	ARMA(p, q) Both
ACF Plot	Tails off gradually	Cuts off after lag q	Tails off gradually
PACF Plot	Cuts of after lag p	Tails of gradually	Tails off gradually
	$y_t = \sum_{i=1}^{p}\phi_i y_{t-i} + \epsilon_t$	$y_t = c + \sum_{i=1}^{q}\theta_i\epsilon_{t-i} + \epsilon_t$	$y'_t = c + \sum_{i=1}^{p'}\phi_i y'_{t-i} + \sum_{i=1}^{q}\theta_i\epsilon_{t-i} + \epsilon_t$

other options?
SARIMA
GARCH
EGARC
GJR

Wait, how do I decompose into Trend, Seasonal, and Residual Components??
Multiplicative
y(t) = Trend(t) * Seasonality(t) * Residual(t)
Use when magnitude does change. Captures non-linear relationships
consider monthly sales data from a manufacturer. In quarters with high sales (product launch/holidays), the profits experience substantial increases and are substantially lower in others. A multiplicative model may be appropriate
Additive
y(t) = Trend(t) + Seasonality(t) + Residual(t)
Use when magnitude of seasonality does not change with level of time. Easy to interpret
consider monthly sales data for a retail store. If the sales exhibit a seasonal pattern (e.g., higher sales in summer and lower sales in winter), and this pattern remains relatively consistent regardless of whether the total sales volume is high or low, then an additive decomposition model may be appropriate.

```python
import pandas as pd
import numpy as np
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

# Generate example data (monthly sales with seasonality)
np.random.seed(0)
dates = pd.date_range(start='2023-01-01', periods=36, freq='M')
sales = 100 + np.arange(36) + np.random.normal(0, 10, 36)

# Create a DataFrame with the time series data
df = pd.DataFrame({'Date': dates, 'Sales': sales})
df = df.set_index('Date')

# Perform additive decomposition
result_add = seasonal_decompose(df['Sales'], model='additive', period=12)

# Perform multiplicative decomposition
result_mul = seasonal_decompose(df['Sales'], model='multiplicative', period=12)

# Plot the original time series and the decomposed components
plt.figure(figsize=(12, 8))

plt.subplot(4, 1, 1)
plt.plot(df['Sales'], label='Original Time Series')
plt.legend(loc='upper left')

plt.subplot(4, 1, 2)
plt.plot(result_add.trend, label='Trend (Additive)')
plt.legend(loc='upper left')

plt.subplot(4, 1, 3)
plt.plot(result_add.seasonal, label='Seasonality (Additive)')
plt.legend(loc='upper left')

plt.subplot(4, 1, 4)
plt.plot(result_add.resid, label='Residual (Additive)')
plt.legend(loc='upper left')

plt.tight_layout()
plt.show()
```

Just trying to figure out the general trend?
Smoothing
Simple Smoothing
Exponential Smoothing
Note that smoothing techniques do not FORECAST a value. They estimate a past trend. The objective is different!
Exponential Smoothing- more weight placed on recent data
Single Exponential (no trend or seasonality) - use if data lacks trend or seasonality
Double Exponential (just captures trend) - use if you have a trend in your data
Triple Exponential (captures both trend and seasonality) - use if you have both trend and season in data

Simulation
Continuous (ex: fluid flow)
Define initial conditions and dynamics
Select the time step for simulation
Choose the numerical integration method
Write simulation code and initialize variables
Sensitivity analysis
Euler's method
Runge-Kutta methods
Discrete Event (ex: Queuing)
Example: Modeling arrivals
Poisson Distribution of arrivals *assumed independent*
Define parameter of interest lambda e.g. average 30 customers per hour. lambda=30
Specify Probability Mass Function
Simulate arrivals by generating random number from Poisson lambda = 30
Poisson λ = 3
The probability mass function of the Poisson distribution gives the probability of observing k events in a fixed interval of time, given the average rate of occurrence (λ). It is given by:
$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$

Optimize [how many of each type of car should I manufacture?]
Goal Programming
More than one Objective?
Stochastic Processes
local min
local max
saddle point
List constraints and resources
Explore Solution space
Evaluate objective function
Determine optimal solution if possible
Need to improve solution?
Branch and Bound
Branch and Cut
Dynamic Programming
Greedy Algorithms
Genetic Algorithms
Simulated Annealing
tabu Search
Ant Colony Optimization
Particle Swarm Optimization
Constraint Programming
Integer Programming
Solvers (e.g., CPLEX, Gurobi)
Not cutting it? Need more specific tools?
julia PROGRAMMING for OPERATIONS RESEARCH Second Edition

General optimization methodologies
- Deterministic
  - Convex
    - Linear programming (LP)
    - Nonlinear programming (NLP)
  - Nonconvex
    - Continuous (NLP)
    - Discrete
      - Integer
        - Mixed integer linear programming (MILP)
        - Integer programming (IP)
      - Mixed integer nonlinear programming (MINLP)
- Stochastic
  - Decision tree models
  - Game theory
  - Stochastic process
  - Monte Carlo simulation
- Artificial intelligence
  - Neural networks
  - Support vector machines
  - Genetic algorithms
  - Ant colony
  - Particle swarm optimization
  - Simulated annealing

© Brandon Wallace
brandonlwallace