Data Science Workflow

*Naumaan Nayyar*

# Problem Identification

* Remember to be SMART – Specific, Measurable, Attainable, Reproducible, Time-bound.
* Identify the class of the problem, for e.g. hypothesis testing, regression, classification, clustering, dimensionality reduction, time series analysis, natural language processing, anomaly/outlier detection.

# Acquisition

* Identify what data you will need, and from where you will get it.
* Does the dataset come with a data dictionary? If not, you will have to make one, preferably in consultation with a domain expert to understand nuances of the data.
* Figure out the data connections to your data science tools. Will you be using a database, CSV file, web scraping, API?

# Parsing

* Verification of data quality happens here.
* Go through the data dictionary to understand what the dataset contains.
* Identify whether there is missing data.
* Identify whether the data is clean, for e.g. absence of human input errors.
* If the data is unstructured, for e.g. freeform text, it can be broken down into constituent words here.
* Categorical data should be converted to dummy variables.

# Mining

* This is where you understand the data.
* You can identify descriptive statistics about the data: measures of central tendency (mean, median, mode) and measures of dispersion (variance, standard deviation, skewness, kurtosis). Other functions include min, max, quartiles.
* Variables can further be analyzed in isolation using visualization, such as histograms, violin plots, box plots.
* Variables be analyzed in combination using functions and plots.
  + Plots include scatter plots (pair plots make it easy to do this for the entire dataset) and correlational heat maps for continuous vs continuous data, factor plots for continuous vs categorical data, matrix representation for categorical vs categorical data.
  + Functions include:
    - Continuous vs continuous: Pearson correlation coefficient, linear regression
    - Continuous vs categorical: (binary categorical) t-test if standard deviation is not known, z-test if known. (multiple categories) one-way analysis of variance (ANOVA).
    - Categorical vs categorical: Chi-squared test.
* With the correlational analysis, what you are looking for depends on the type of variables. For comparing two independent/predictor variables, you want the correlations to be low. When you are comparing a dependent/outcome variable with an independent variable, you want the correlation to be high. Correlational relationships only measure linear relationship.
* Is there a multicollinearity issue? This can be identified by building a linear model. It will show garbage coefficient values. Also, remember that when dummy variables are created from a categorical variable, one of them should always be excluded. Otherwise, multicollinearity will happen.
* Is there confounding or interaction/effect modification? These can be detected by stratification, building linear models, etc.

# Refining

* Variable transformations happen here.
* Do you need to scale the data? What scaling will you adopt? Centering, equal standard deviation, fixing maximum absolute value, etc. Consider pros and cons of each approach. For example, all of them will require data to be interpreted in different ways. MaxAbsScaler will not impact sparsity whereas the others will.
* Do you need to reduce the number of variables/features because they are too many or because they are highly correlated? Consider using PCA, eliminating variables based on domain knowledge, using a model with LASSO or Elastic Net regularization etc.
* Do you need to create derived columns, such as cost per click, BMI, etc? This is especially important in several models as derived columns can account for non-linearity etc. that the model may not be capable of handling on its own.
* Techniques to deal with imbalanced classes:
  + Up-sampling
  + Down-sampling
  + Combine minority classes
  + Generate synthetic data
  + Change performance metric to TPR, FPR, Sensitivity, Specificity, AUC, ROC
  + Use cost-sensitive algorithms with penalties, such as Penalized SVM. Some algorithms naturally perform well if there isn’t too much imbalance, such as decision tree based models
  + Change problem type to anomaly/outlier detection
* For time-series, detrending and deseasonalizing also happen here. Remember moving averages, weighted moving averages, differencing.

# Modeling

* The actual fun modeling part!
* Hypothesis testing
  + Metrics: p-value
  + Algorithms: A/B testing, t-tests, z-tests, F-test, Linear regression
* Regression
  + Metrics: R-squared, Adjusted R-squared (adjusts for overfitting with increasing variables), Mean Squared Error, Mean Absolute Error
  + Algorithms: Linear regression, Regularized linear regression (LASSO, Ridge, ElasticNet), Gradient Descent, Neural Networks, SVM regression, Decision tree-based regression
* Classification
  + Metrics: Accuracy/Misclassification Rate, Sensitivity/Specificity, TPR/FPR, Precision/Recall, F-measure, ROC, AUC
  + Algorithms: k-Nearest Neighbors, decision-tree based methods (decision trees, random forests, gradient boosting), logistic regression, Support Vector Machines
* Clustering
  + Metrics: Silhouette Score (elbow method)
  + Algorithms: Density, distribution, hierarchical, linkage. e.g. k-means, DBSCAN, decision-tree based
* Time series analysis
  + Metrics: Mean Squared Error, Mean Absolute Error
  + Algorithms: ARIMA (extension of AR, MA, ARMA), ETS, LSTM recurrent neural networks
* Natural Language Processing
  + Count Vectorizer, Term frequency- Inverse Document Frequency Vectorizer, Bag of words classifier, tokenization, parsing, lemmatization, stemming, topic modeling (LDA), LSI, Word2Vec, etc.

# Presentation/Product

* Data science will only be a small part of the overall system/product. Consider pipelines and architectural questions when integrating your work.
  + Database types: Relational (rigid structure) vs Non-relational (not so rigid)
* For a presentation, be sure to cater to your audience: non-technical vs technical, etc.