Data Science
- Conceptual Foundations

The Role of a Data Scientist

- A 'Data Scientist' can be seen as a combination of two professions: 'Data Engineer' and 'Data Analyst'
- A 'Data Engineer' is able to find and take raw data and clean it up for efficient processing.
- A 'Data Analyst' is able to take data, organize it further, and work with it to make it sing its story.

Overview of the Process

When combining the skills of those two professions, one acts and performs as a 'Data Scientist'. This can entail the need to find relevant data; sometimes using SQL to access a database is needed or sometimes simply downloading a partially organized CSV file is the correct path. This first part is the bulk of a Data Engineer's job, the next step can then be clumped with the first part of a Data Analyst job, cleaning and organizing the data for smooth processing.

The data's labels need to be appropriate and in the proper format, usually a Pandas Dataframe. EDA (Exploratory Data Analysis) then deepens the connection to and understanding of the data. After, the data should be organized into the proper shape. The ideal shape of a dataframe is based on Rows vs. Columns and allows the models and algorithms to be used most efficiently. There are a limited number of models that can be applied to each scenario, and often only 2 or 3 are needed to be implemented and tweaked to acquire needed informative results and insights.

Exploratory Data Analysis (aka 'EDA')

This is where the data scientist gets intimate with the data. It involves repetitive actions and testing of the data from various angles; data visualizations help give an initial snapshot of the spread of the data and which next steps are necessary. If done properly, one should be aware of:

- What is missing from the data
- If there are outliers and the potential reasoning behind them
- Which features may have most of an effect on your target variable
- The value count, ranges of the data, and more

Type of Problem & Target Variable

When approaching the data, one should have an idea of what needs to be solved and the corresponding target variable. It often entails a translation of the business department's desires into technical information to be processed as numbers.

In machine learning, all issues are boiled down to two categories of problems:

Supervised Learning or Unsupervised Learning

Supervised Learning Problems

These are generally easier to work with and require less gap-filling. A dataset is approached that already contains some of the values for the specific target variable that needs to be solved. This target variable would be either numerical or categorical.

When numerical, regression techniques are used to create a fine-tuned trend line within the data.

When categorical, logistic regression is often used for classification of binary target variables; if not binary, the data's categories can be encoded and processed as numerical values within a Random Forest Classification model.

Unsupervised Learning Problems

Sometimes data is approached and it is up to the data scientist to create and define the target variable. Along the way, it may be necessary to create unique features that are composites of the already present variables. By manipulating the data and its subsets, clumps or groups of data may begin to appear; these clusters can be representative of the different values for the target variable being designed within a newly created column. The classic example used by many educational platforms is the dataset covering various species of penguins. Clustering techniques used on the properly shaped data can allow one to classify different species or genders of penguins and more with remarkable precision.

The Target Variable

Whether the target variable is already defined or needs to be designed, one should have a grasp on the potential values it may contain. Are these values naturally numerical or do they require encoding for more efficient processing? If there are just two potential values, such as True/False or Yes/No, this makes it binary and allows it to be processed more simply as a numerical Boolean (1/0).

Model Selection

A good practice for a data scientist is to process the data using at least two different algorithms / models. Once the answer to the two questions below are clear, this narrows down which models may be implemented. For simplification purposes, six of the most common and efficient models are presented further below.

1. Problem is: 'Supervised or Unsupervised' ?
2. Target Variable is: 'Numerical or Categorical' ?

The 6 Models:

- **K-Means Clustering**
- **Hierarchical Clustering (Dendrograms)**
- **Linear Regression (for linear relationships)**
- **Random Forest Regression**
- **Logistic Regression (when target variable is binary)**
- **Random Forest Classification**

Model Selection per Scenario:

- Unsupervised Learning =
  K Means Clustering / Hierarchical Clustering (Dendrograms)
- Supervised Learning & Numerical Target Variable =
  Linear Regression / Random Forest Regression
- Supervised Learning & Categorical Target Variable =
  Logistic Regression / Random Forest Classification

Model Type Explanations

- K-Means Clustering
  - K represents the number of centroids (center point of cluster) that the algorithm will try to parse out after iterating through the data.
- Hierarchical Clustering (Dendrograms)
  - Slower but can handle more arbitrary data without having to initially define the amount of centroids/clusters that are sought after.
- Linear Regression (for linear relationships)
  - Essentially creating a trend line for the data; quicker though requires simpler data with more normality.
- Logistic Regression (when target variable is binary)
  - Divides data between two sides of a manually set S-Curve; quicker and meant for simpler linear data.
- Random Forest (Regression / Classification)
  - Robust and complex algorithm that can handle any dataset, providing more accurate results at the cost of transparency.
  - Essentially creating a vast forest out of decision trees, where each tree is like a flow chart with nodes representing the features.

Ideal Shape

The ideal shape of a dataframe that is used by a data scientist should have the most concise amount of columns (features) and the most amount of rows with relevant data. Rows containing much missing data should be discarded or imputed with the appropriate values.

Depending on what you are trying to solve, 100 columns may be the best for one problem, while 6 columns for another problem. If the data is not concise and there are too many extraneous columns/features, this may lead to a more resource-intensive process and poorer results.

Model Results

Common 'Scoring' Methods for the different models:

- **Regression Metrics**
  'R2' or 'RMSE' (Root-Mean Squared Error) to show the variance of the 'trendline' from the actual data.

- **Classification Metrics**
  'Accuracy', 'Precision', 'F1-Score', etc.
  'ROC-AUC Score' can be used for binary-classification problems

- **Unsupervised Learning Metrics**
  'Silhouette Score' shows the positive attributes of separation and density of the clusters.

Initial Scores as a Baseline

Once the scores are acquired for at least two different models, a review of its inputs should be reviewed and reconsidered.

Need to check with more runs through the algorithm, perhaps with less features, to see if it results in an improved score.

Besides the data itself, it is important to work on hyperparameter tuning; tweaks to the number of iterations and other parameters defined within the algorithm can sometimes have profound effects on the results. A list of various configuration variables can also be tried via a loop when working with Python.

Common Techniques to Improve Scores:

- Dimensionality Reduction (PCA)
  Reduce the amount of features by aggregating and combining columns; easier to process and compute.

- Retrying box-plots / scatter plots
  Re-examine presence of unnecessary data per newly obtained insights; remove data deemed arbitrary.

- Cross-Validation
  Have the model run through various partitions of the data and parameters to help avoid under/over-fitting.

Performance Run

After running the models multiple times and getting a record of various scores, the performance run should be the well documented and presented version of the algorithm that shows the necessary insights most accurately and clearly.

The aim is not to acquire the most successful numbers with regards to business terms, but to obtain the most truthful insights from the data that can lead to the correct follow-up actions that will have the most positive impact. KPI results should not be sugar-coated, but the cleanest avenue to improve them is what is desired.

Final Results

After the performance run, the data scientist now has the best insights of his target variable. Perhaps the expected revenue of a product for certain months in the future was calculated as the target variable; predicting which items should be best-selling based on historical data can now be done confidently. This helps ensure to maintain stock of products leading to its sale without hassle.

Proper insights can often be even more profound, offering a deeper understanding into the moving parts of a certain system and their inner-choreography. Better understanding a system can also offer invaluable leverage when dealing with related systems.

In Conclusion...

The skills employed by a data scientist are not trivial; translating the real world into numerical data for efficient processing to deliver insightful predictions can sometimes seem intimidating.

In 2017 there was a false-scare about Amazon's ability to predict when a woman was pregnant even before she knew of it herself. In the near future, if not already, this will be a real technique used by companies to deliver things you will 'want/need' before you knew you wanted or needed it. This may be good or bad, but it just shows that there are systems that know parts of you, better than you.