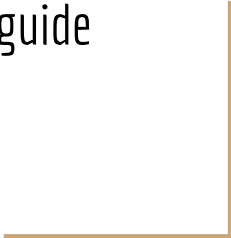





Data Science Projects

A (basic) step-by-step guide



Data Science Project Process

- 
1. Understand the problem
 2. Collect the data
 3. Explore the data
 4. Clean the data
 5. Splitting the data into training and test sets
 6. Build the model
 7. Evaluate the model
 8. Prepare the project for presentation

1. Understand the problem

- ❖ Look at the big picture and the challenge facing the client, and try to understand if machine learning, deep learning, or statistical analysis is necessary.
- ❖ What kind of model is the right solution for our data, research question, and goals?
- ❖ Should we use a white-box or black-block algorithm, or some combination of both?
 - Does the client need to be able to understand or use the algorithm, or just the results?
 - Will this be a one-time project, or will it require continuous integration?
 - What kind of computing power will we need? Will it be ongoing?

2. Collect the data

- ❖ Do we have a data set already prepared?
 - If so, is the quality high enough?
 - Are there enough observations?
- ❖ Do we need to gather data from scratch?
 - Is it feasible?
 - How long will this take? Do we have the resources?
- ❖ What other types of data can we find to supplement the understanding of the problem and its potential resolution?

3. Explore the data

- ❖ Understand the data: what variables do we have, what are their qualities, data types of each feature, etc.
- ❖ Feature engineering: Select relevant covariates and have an understanding/narrative to explain why they are relevant. If necessary, create additional features out of existing data.
- ❖ Visualize the data: Visualize variables of interest and explore the relationships before running any types of modeling. Is there really a relationship to encapsulate there? Are there things we need to lookout for?

4. Clean the data

- ❖ Drop unused rows/columns
- ❖ Make sure x is a matrix and y is a vector
- ❖ Take care of missing values: delete or impute
- ❖ Take care of categorical data (if necessary)
- ❖ Feature scaling (if necessary)
- ❖ Regularization to prevent overfitting (if necessary)
- ❖ etc.

5. Training and test data sets

- ❖ We train a model using some data set. But we need a separate data set to test to see how the model performs. So, we use a train-test split:
 - Training data set: The sample of data used to fit the model. The model sees and learns from this data.
 - Test data set: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.
- ❖ There are more complicated versions of this, but this is the general idea.
- ❖ This can be done manually, or with packages (caret in R, sklearn in Python, etc.)

6. Build the model

- ❖ This is often the simplest part of the process, once you've decided on the type of algorithm to use.
- ❖ Some possible methods:
 - All-in: put every variables in because of either prior knowledge (you know it's important) or to prepare for backward elimination
 - Backward elimination: fit and test models until the variable with the highest p-value is still smaller than your chosen significant level
 - Forward selection: incrementally add variables with smallest p-value, stop when p-value > significant level
 - Bidirectional selection: combination of backward elimination and forward selection
 - etc.

7. Evaluate the model

- ❖ This is one of the most important parts of the process. You have to confirm that your model is accurate and does not violate important assumptions before any conclusions can be made. Some ways to do this:
 - Plot model residuals
 - Classification accuracy
 - RMSE
 - Simply seeing if it makes sense to your human brain
 - etc.

8. Prepare the project for presentation

- ❖ Interpret model output
- ❖ Build visualizations, Shiny apps/dashboards, Tableau presentations, etc.
- ❖ Create the narrative and story of the project and results, and make sure you can clearly explain the results to someone who has no statistical or data science knowledge
- ❖ If needed, develop recommendations and next steps based on the results