# Nanodegree of Data Science

#### Lesson1: The Data Science process

**The CRISP-DM Process (Cross Industry Process for Data Mining)**

The lessons leading up to the first project are about helping you go through CRISP-DM in practice from start to finish. Even when we get into the weeds of coding, try to take a step back and realize what part of the process you are in, and assure that you remember the question you are trying answer and what a solution to that question looks like.

1. **Business Understanding** - this means understanding the problem and questions you are interested in tackling in the context of whatever domain you're working in. Examples include

* How do we acquire new customers?
* Does a new treatment perform better than an existing treatment?
* How can improve communication?
* How can we improve travel?
* How can we better retain information?

2. **Data Understanding** - at this step, you need to move the questions from **Business Understanding** to data. You might already have data that could be used to answer the questions, or you might have to collect data to get at your questions of interest.

3. **Data Preparation**

Luckily stackoverflow has already collected the data for us. However, we still need to wrangle the data in a way for us to answer our questions. The wrangling and cleaning process is said to take 80% of the time of the data analysis process. You will see that will hold true through this lesson, as a majority of the remaining parts of this lesson will be around basic data wrangling strategies.

We will discuss the advantages and disadvantages of the strategies discussed in this lesson.

————-

When looking at the first two questions:

1. How to break into the field?
2. What are the placement and salaries for those who attended a coding bootcamp?

we did not need to do any predictive modeling. We only used **descriptive** and a little **inferential** statistics to retrieve the results.

Therefore, all steps of CRISP-DM were not necessary for these first two questions. CRISP-DM states 6 steps:

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Data Modeling

5. Evaluate the Results

6. Deploy

For these first two questions, we did not need step 4. In the previous notebooks, you performed steps 3 and 5 without needing step 4 at all. A lot of the hype in data science, artificial intelligence, and deep learning is integrated into step 4, but there are still plenty of questions to be answered not using machine learning, artificial intelligence, and deep learning.

**All Data Science Problems Involve**

1. Curiosity.
2. The **right** data.
3. A tool of some kind (Python, Tableau, Excel, R, etc.) used to find a solution (You could use your head, but that would be inefficient with the massive amounts of data being generated in the world today).
4. Well communicated or deployed solution.

**Extra Useful Tools to Know But That Are NOT Necessary for ALL Projects**

* Deep Learning
* Fancy machine learning algorithms

With that, you will be getting a more in depth look at these items, but it is worth mentioning (given the massive amount of hype) that they do not solve all the problems. Deep learning cannot turn bad data into good conclusions. Or bad questions into amazing results.

When looking at the first two questions:

1. How to break into the field?
2. What are the placement and salaries for those who attended a coding bootcamp?

we did not need to do any predictive modeling. We only used **descriptive** and a little **inferential** statistics to retrieve the results.

Therefore, all steps of CRISP-DM were not necessary for these first two questions. The process would look closer to the following:

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Evaluate the Results

5. Deploy

However, for the last two questions:

1. How well can we predict an individual's salary? What aspects correlate well to salary?
2. How well can we predict an individual's job satisfaction? What aspects correlate well to job satisfaction?

We will need to use a predictive model. We will need to pick up at step 3 to answer these two questions, so let's get started. The process for answering these last two questions will follow the full 6 steps shown here.

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Model Data

5. Evaluate the Results

6. Deploy

**Dealing with missing data**

Dropping the rows/columns with missing data

Though **dropping** rows and/or columns holding missing values is quite easy to do using numpy and pandas, it is often not appropriate.

Understanding why the data is missing is important before dropping these rows and columns. In this video you saw a number of situations in which dropping values was not a good idea. These included

1. Dropping data values associated with the effort or time an individual put into a survey.
2. Dropping data values associated with sensitive information.

In either of these cases, the missing values hold information. A quick removal of the rows or columns associated with these missing values would remove missing data that could be used to better inform models.

Instead of removing these values, we might keep track of the missing values using indicator values, or counts associated with how many questions an individual skipped.

In the last video, you saw cases in which dropping rows or columns associated with missing values would not be a good idea. There are other cases in which dropping rows or columns associated with missing values would be okay.

A few instances in which dropping a row might be okay are:

1. Dropping missing data associated with mechanical failures.
2. The missing data is in a column that you are interested in predicting.

Other cases when you should consider dropping data that are not associated with missing data:

1. Dropping columns with no variability in the data.
2. Dropping data associated with information that you know is not correct.

In handling removing data, you should think more about why is this missing or why is this data incorrectly input to see if an alternative solution might be used than dropping the values.

One common strategy for working with missing data is to understand the proportion of a column that is missing. If a large proportion of a column is missing data, this is a reason to consider dropping it.

There are easy ways using pandas to create dummy variables to track the missing values, so you can see if these missing values actually hold information (regardless of the proportion that are missing) before choosing to remove a full column.

Imputing missing data

Imputation is likely the most common method for working with missing values for any data science team. The methods shown here included the frequently used methods of imputing the mean, median, or mode of a column into the missing values for the column.

There are many advanced techniques for imputing missing values including using machine learning and bayesian statistical approaches. This could be techniques as simple as using k-nearest neighbors to find the features that are most similar, and using the values those features have to fill in values that are missing or complex methods like those in the very popular [**AMELIA library**](https://cran.r-project.org/web/packages/Amelia/Amelia.pdf).

Regardless your imputation approach, you should be very cautious of the **BIAS** you are imputing into any model that uses these imputed values. Though imputing values is very common, and often leads to better predictive power in machine learning models, it can lead to over generalizations. In extremely advanced techniques in Data Science, this can even mean [**ethical implications**](https://intelligence.org/files/EthicsofAI.pdf). Machines can only 'learn' from the data they are provided. If you provide biased data (due to imputation, poor data collection, etc.), it should be no surprise, you will achieve results that are biased.

One of the most common methods for working with missing values is by imputing the missing values. Imputation means that you input a value for values that were originally missing.

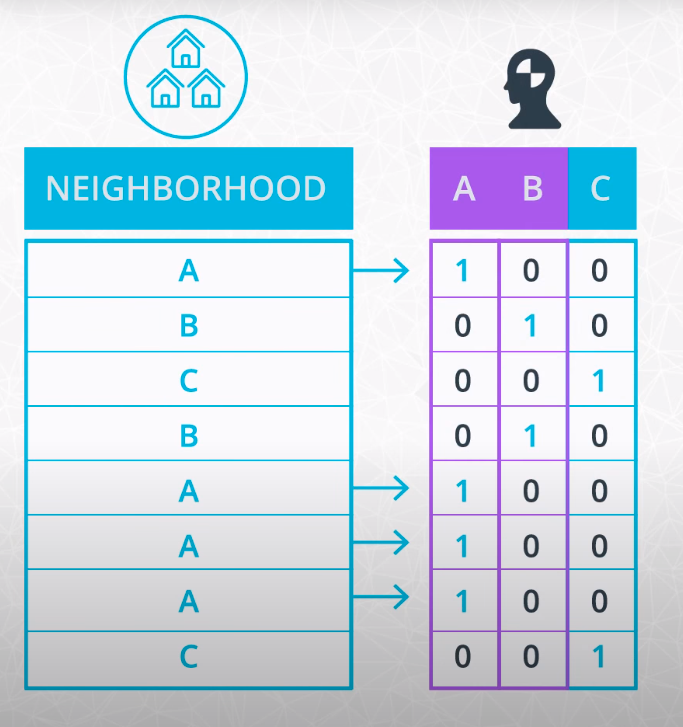
It is very common to impute in the following ways:

* Impute the mean of a column.
* If you are working with categorical data or a variable with outliers, then use the mode of the column.
* Impute 0, a very small number, or a very large number to differentiate missing values from other values.
* Use knn to impute values based on features that are most similar.

In general, you should try to be more careful with missing data in understanding the real world implications and reasons for why the missing values exist. At the same time, these solutions are very quick, and they enable you to get models off the ground. You can then iterate on your feature engineering to be more careful as time permits.

**Categorical variables**

The best way to work with categorical variables is with ones and zeros as seen in the image.



**Overfitting**

Overfitting is a common problem when our model does not generalize to data it has not seen before. Assuring you build models that not only work for the data the model was trained on, but also generalize to new (test) data, is key to building models that will be successful to deploy and that will become successful in production.