Solution Fit for University Admit Eligibility Predictor

In order to solve this problem, I'm going to take you through what I call a
"Data Science Pipeline". Given any problem statement, follow this pipeline so that your approach is structured.

1.Define the problem -

Write down the problem statement and understand what you're trying to solve. In this case, our objective is to predict whether a student will get an admit or not. So it means, that this is a binary classification problem.

2. Generate your own hypothesis -

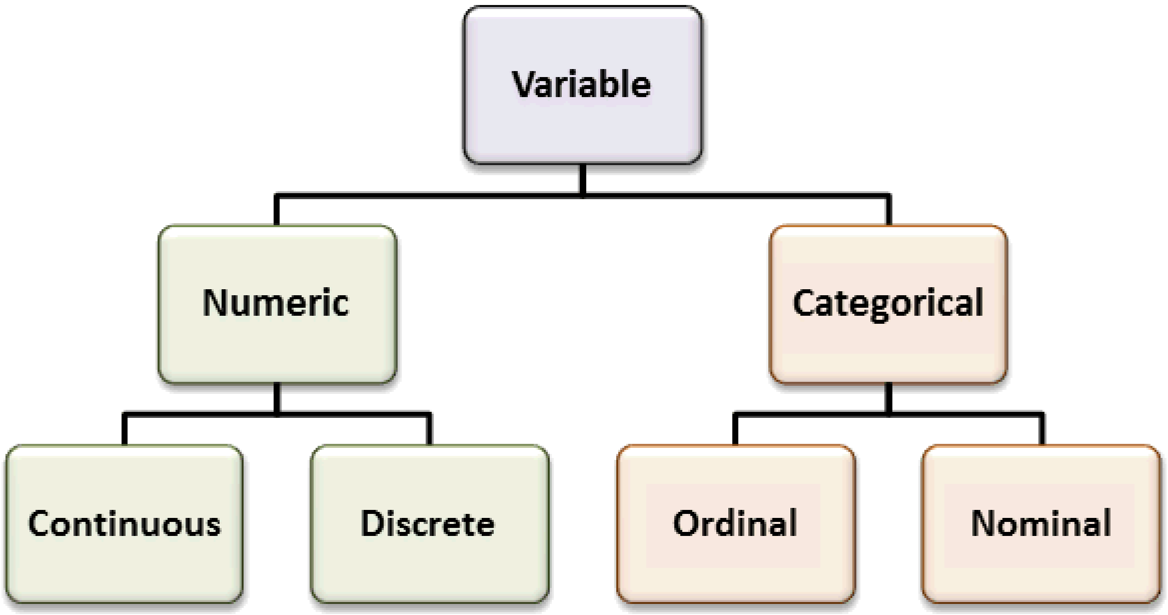
Next, list down all the things that you think can affect our objective viz. List down all the possible features with respect to our target feature.
In our case, we got ask this question — *What are the factors that can affect a student's admission?*
Take a piece of paper and write down your own hypothesis. I spent some time and came up with the following features
- GRE Score
- TOEFL Score
- Statement of Purpose (SOP)
- Letter of Recommendation (LOR)
- Academic Performance (GPA)

- Extra Curricular Activities (Sports, Math Olympiad etc..)
- Outstanding Achievements
- Projects and Research

Make sure that you write down at least 10–20 for any problem statement. This helps us get a deeper understand of what we are trying to solve and prompts us to think beyond the available dataset.

3. Get the Dataset -

Next step is to get the data. We're using UCLA hypothetical data for graduate admissions. You can download the dataset here.
You can now map your hypothesis with the given dataset. See how many features you can add or modify in order to improve the quality of the dataset. Identify each feature as either
(a) Continous Variable or (b) Categorical Variable

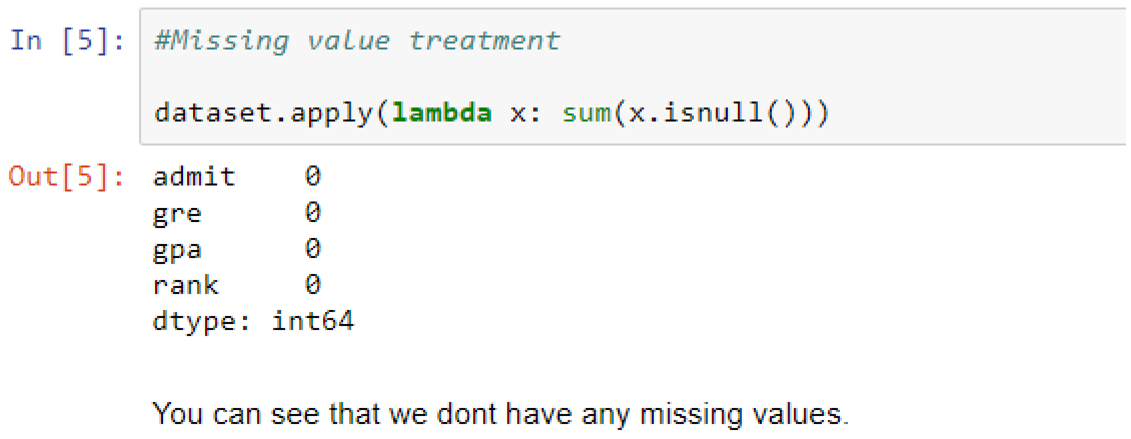


(Source: Stack Exchange)

4. Data Cleaning -

Most of the time, the dataset will have lot of anomalies like missing values, outliers and so on. Always preprocess the data before moving on to the next step.

- ***Missing Value Treatment:*** You can use Mean-imputation (for continous variables) or Mode-imputation (for categorical variables)



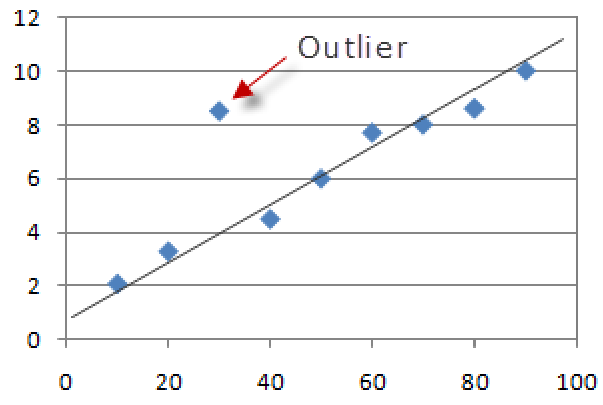
```
In [5]: #Missing value treatment

        dataset.apply(lambda x: sum(x.isnull()))

Out[5]: admit    0
        gre      0
        gpa      0
        rank     0
        dtype: int64
```

You can see that we dont have any missing values.

- ***Outlier treatment:*** An *outlier* is an observation that lies an abnormal distance from other values in a random sample.



There are four ways to treat it — Deleting observations, Imputing, Creating Bins and Treat Separately.

- ***Feature Engineering:*** Combining, adding, deleting, splitting, scaling features in order to increase the accuracy of the model

5. Exploratory Data Analysis (EDA) -

Now its time to get our hands dirty. Let us explore the data and understand the given features.
This dataset has a binary response (outcome, dependent) variable called `admit`. There are three predictor variables: `gre`, `gpa` and `rank`. We will treat the variables `gre` and `gpa` as continuous. The variable `rank`takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.

EDA helps us get a clear idea about our features. It also helps us capture any trend or seasonality of data points.
In our case, we can see that Higher the GRE score and GPA, more the chances of getting admit.

6. Predictive Modeling -

This is a binary classification problem. The output has only two possibilities either *Yes (1)* or *No (0)*. There are lot of classification algorithms out there so how do we know which one is the best?
We don't. This is something that comes with experience and expertise. But not to worry, we do have a work around. We can fit multiple algorithms onto our data points and then evaluate the model. This way we can choose the best model which has the least error.