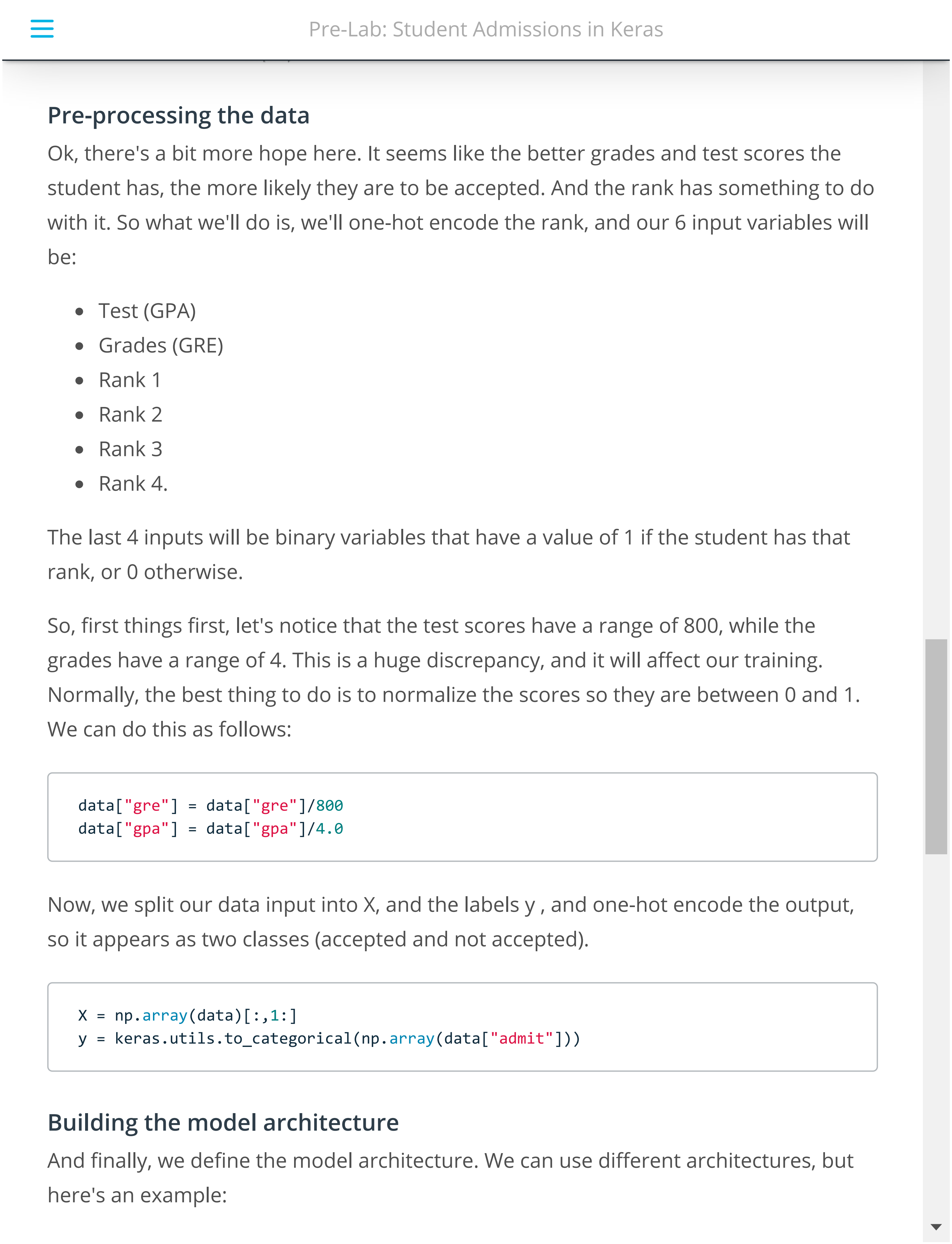

## Pre-processing the data

Ok, there's a bit more hope here. It seems like the better grades and test scores the student has, the more likely they are to be accepted. And the rank has something to do with it. So what we'll do is, we'll one-hot encode the rank, and our 6 input variables will be:

- Test (GPA)
- Grades (GRE)
- Rank 1
- Rank 2
- Rank 3
- Rank 4.

The last 4 inputs will be binary variables that have a value of 1 if the student has that rank, or 0 otherwise.

So, first things first, let's notice that the test scores have a range of 800, while the grades have a range of 4. This is a huge discrepancy, and it will affect our training. Normally, the best thing to do is to normalize the scores so they are between 0 and 1. We can do this as follows:

```
data["gre"] = data["gre"]/800
data["gpa"] = data["gpa"]/4.0
```

Now, we split our data input into X, and the labels y , and one-hot encode the output, so it appears as two classes (accepted and not accepted).

```
X = np.array(data)[:,1:]
y = keras.utils.to_categorical(np.array(data["admit"]))
```

## Building the model architecture

And finally, we define the model architecture. We can use different architectures, but here's an example: