Correlation and Heatmaps

This lesson will introduce how to calculate and visualize correlations between quantities in python.

WE'LL COVER THE FOLLOWING

Correlation

Correlation is a mathematical technique that shows how strongly two variables are linked. It quantifies the strength of the relationship. For instance, we know that the weight and height of a person are correlated. Taller people tend to have more weight. Hence, we say that height and weight are correlated.

Correlation is measured in terms of a number called **correlation coefficient**, which ranges from -1 to 1. The value of 1 or -1 denotes complete correlation, while 0 indicates that no correlation is present between the two variables. Negative values mean there is an inverse relationship between the two variables, while a positive value denotes a direct relationship.

Pandas has the function `corr` that can be called on a dataframe. Let's see an example of this on our Default of Credit Card Clients Dataset.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('credit_card_cleaned.csv')
# Calculate correlations
corr = df.corr()
print(corr)
```

We just use the function `corr` with the dataframe `df` in **line 5**, which gives us a table with correlation values for each pair of variables. This table is known as the **correlation matrix**. We print the correlations in the next line.

Looking at the correlation matrix, it can be very hard to study these values in this printed table. Therefore, we use *heatmaps* to visualize the correlations.

Heatmap

A **heatmap** is a graphical representation of data where individual values are represented as colors. The intensity of the colors indicates the values.

`Seaborn` is a Python module that is used for plotting. It has the function `heatmap` that we will be using to plot the heatmap. We will be concerned with the following arguments of the function:

- `data`: The data or matrix from which to plot the heatmap.
- `annot` - **optional**: Whether to write the values on each cell of the heatmap. Expects True/False. False by default.
- `vmin` - **optional**: The minimum on the color bar.
- `vmax` - **optional**: The maximum on the color bar.
- `cmap` - **optional**: The color map to use.

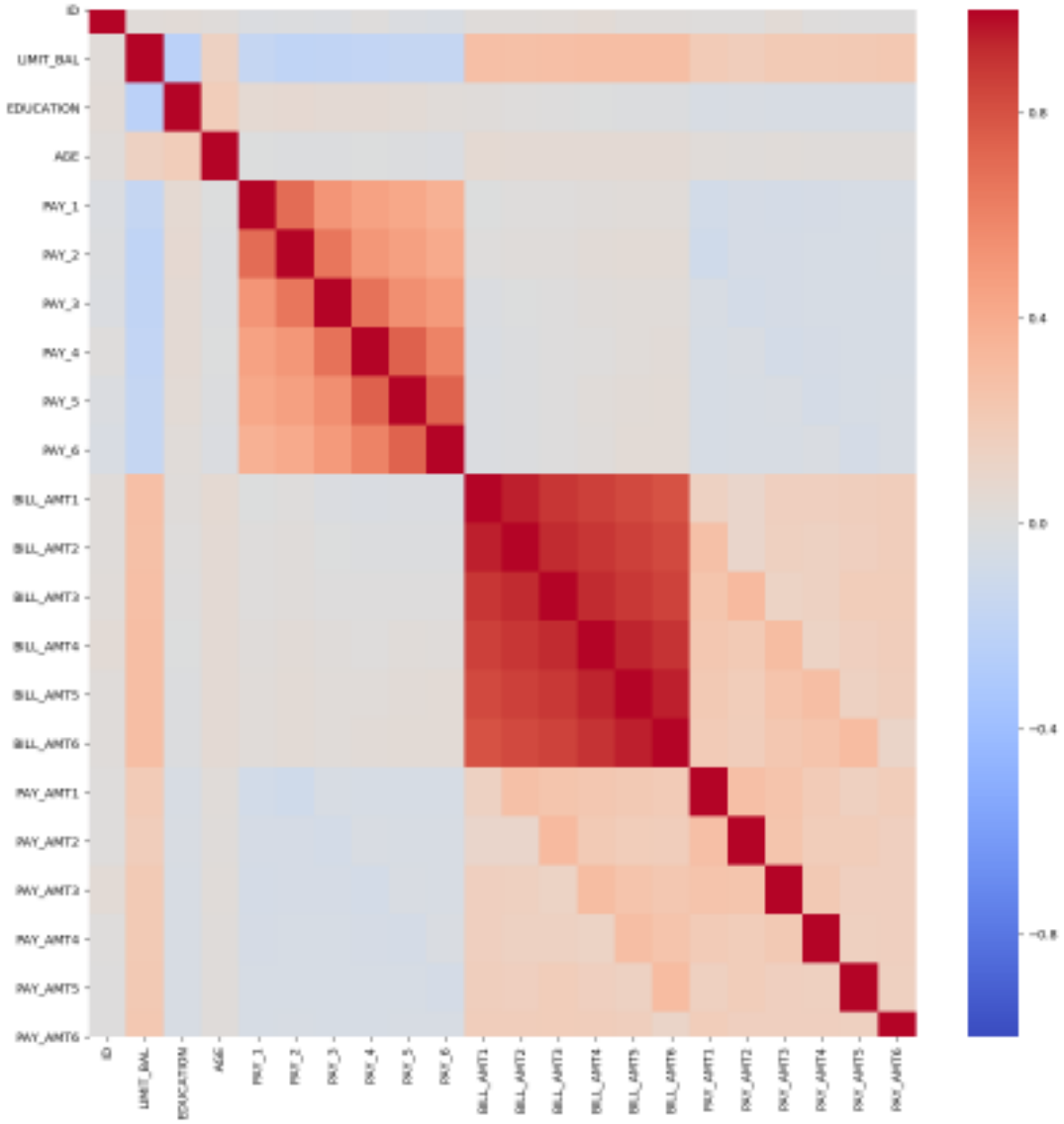
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('student-mat.csv')

# Calculate correlations
corr = df.corr()

# plot heatmap
sns.heatmap(data = corr,vmin = -1.0,vmax = 1.0,cmap = 'coolwarm')
```

We import the seaborn library in **line 3** as `sns`. Then we create the correlation matrix in **line** 7 using the function `corr`. The magic happens in **line 10** where we use the `heatmap` function. We give our correlation matrix `corr` as `data`. Then we set the color bar scales to -1 and 1 since we know that correlation coefficients range from -1 to 1. We set `cmap` to `coolwarm` as

this color map makes it easy to study heatmaps.



From the plot, we can clearly infer that the Bill amount variables (`BILL_AMT1`, `BILL_AMT2`,...) are highly correlated with each other as we suspected from the scatter plots in the last lesson. We can say the same about the payment delay variables (`PAY1`, `PAY2`,...) as well.

We also see some positive correlation between `LIMIT_BAL` and Bill amount variables (`BILL_AMT1`, `BILL_AMT2`,...) which implies that people who were given more credit (higher values of `LIMIT_BAL`) tend to have larger bills.

Interestingly, there is a slight negative correlation between `LIMIT_BAL` and payment delay variables (`PAY1`, `PAY2`,...). This implies that people who are given more credit tend to have fewer payment delays. Maybe because they earn more they are given higher credit in the first place.

earn more they are given higher credit in the first place.

This is how correlation and heatmaps help us to make sense of the data. In the next lesson, we will explore another dataset as an example and see how different techniques are used on different datasets.