Preparing Continuous Features

Module 2 | Chapter 2 | Notebook 3

In this notebook we'll look at the assumptions of logistic regression. We'll also prepare the data to generate better predictions. By the end of this lesson you will be able to:

- What are the assumptions of logistic regression in relation to continuous values.
- How to check the assumptions of logistic regression.

Correlating features

Scenario: Pictaglam, a popular social media platform for sharing photos and videos, has received complaints about fake user accounts. Management at Pictaglam's have asked you to create a machine learning model that would help the platform to distinguish between real accounts and fake accounts.

The file *social_media_train.csv* contains data on real and fake Pictaglam user accounts.

Let's get started by importing the data. Use the `0` th column as `index_col`.

In [1]:
```
import pandas as pd

df = pd.read_csv("social_media_train.csv", index_col=[0])
df.head()
```

Out[1]:

	fake	profile_pic	ratio_numlen_username	len_fullname	ratio_numlen_fullname	sim_name_username
0	0	Yes	0.27	0	0.0	No match
1	0	Yes	0.00	2	0.0	Partial match
2	0	Yes	0.10	2	0.0	Partial match
3	0	Yes	0.00	1	0.0	Partial match
4	0	Yes	0.00	2	0.0	No match

The code for that looks like this:

Column number	Column name	Type	Description
0	`'fake'`	categorical	Whether the user account is real (`0`) or fake (`1`).

Column number	Column name	Type	Description
1	`'profile_pic'`	categorical	Whether the account has a profile picture (`'Yes'`) or not (`'No'`)
2	`'ratio_numlen_username'`	continuous (`float`)	Ratio of numeric characters in the account username to its length
3	`'len_fullname'`	continuous (`int`)	total number of characters in the user's full name
4	`'ratio_numlen_fullname'`	continuous (`float`)	Ratio of numeric characters in the account username to its length
5	`'sim_name_username'`	categorical	Whether the user's name matches their username completely (`'Full match'`), partially (`'Partial match'`) or not at all (`'No match'`)
6	`'len_desc'`	continuous (`int`)	Number of characters in the account's description
7	`'extern_url'`	categorical	Whether the account description contains a URL (`'Yes'`) or not (`'No'`)
8	`'private'`	categorical	Whether the user's contributions are only visible to their followers (`'Yes'`) or to all Pictaglam users (`'No'`)
9	`'num_posts'`	continuous (`int`)	Number of posts by the account
10	`'num_followers'`	continuous (`int`)	Number of Pictaglam users who follow the account
11	`'num_following'`	continuous (`int`)	Number of Pictaglam users the account is following

Each row of `df` represents a user or user account.

First you should create a `list` called `features_cont` so you can start preparing the continuous features. Store the `df` column names of all continuous columns as `str` entries in it.

In [3]:
```
features_cont = ['ratio_numlen_username', 'len_fullname', 'ratio_numlen_fullname',
                 'len_desc', 'num_posts', 'num_followers', 'num_following']
```

Just like linear regression, logistic regression makes a number of assumptions. The following are relevant for continuous data:

1. The features should not correlate strongly with each other.
2. There should be a linear relationship between the features and the sigmoid-transformed probabilities.

Let's start with the first assumption.

This assumption is shared by logistic regression and linear regression (see *Multiple Linear Regression in Module 1, Chapter 1*). So we'll calculate a correlation matrix for the continuous features here as well. Use the `.my_df.corr()` method and store the result in the variable `corr`. Then print `corr`.

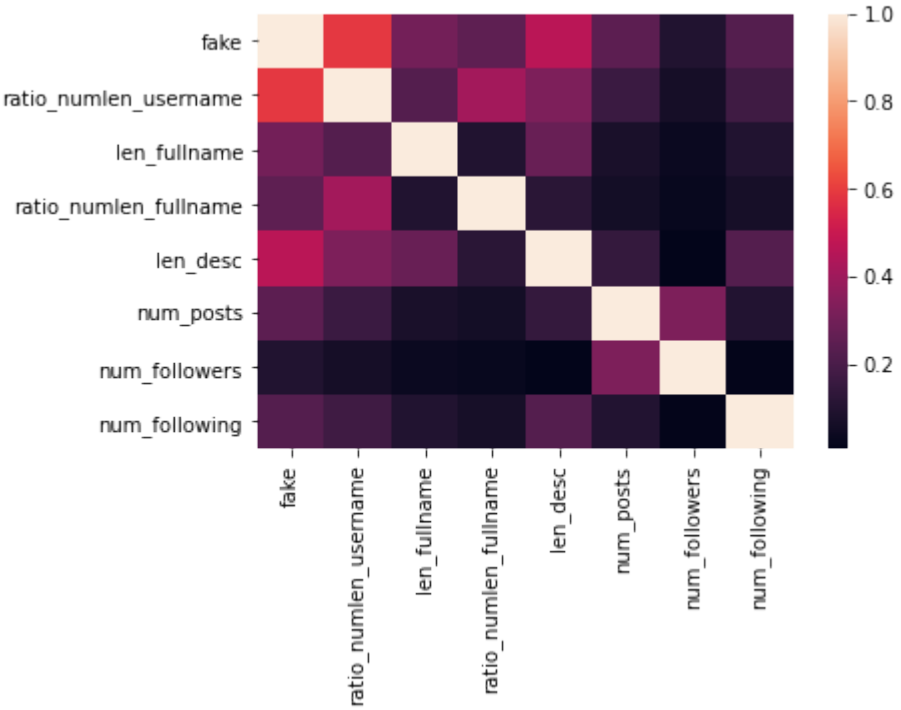
In [7]:
```
corr = df.corr()
```

To make it easier to interpret the correlation matrix, we'll visualize it in a heat map. We only consider the absolute values, since it doesn't matter whether there is a strong positive or a strong negative correlation. Run the following cell to do this.

In [8]:
```
import seaborn as sns

sns.heatmap(corr.abs()) # Heat map of absolute values of correlation matrix
```

Out[8]:
```
<AxesSubplot:>
```



The color scale on the right indicates which colour belongs to which correlation coefficient. Remember that correlation coefficients are between -1 (perfect negative correlation) and +1 (perfect positive correlation) and 0 represents no correlation. A logistic regression assumes a correlation of 0 between the features.

In the case of our correlation matrix, you can see correlation coefficients of 1 in light beige on the diagonal. They show the correlation of features with themselves. So these correlations are always 1.

It becomes interesting with the colors that are not on the diagonal. They represent correlations between different features. If these are above 0.5 (in our case orange and brighter) or below -0.5, you should be worried. If the correlation is above 0.9 or below -0.9, you have a problem with **collinearity**, because these features are responsible for the same dispersion in the data: If

you plotted them, the data points of strongly correlating variables lie almost on a straight line. This leads to the fact that the variables can no longer be considered independently of each other and we are therefore unable to estimate exactly what influence both variables have on the variance of the model (shown as the coefficient of determination R^2 in regression models). This makes our model's predictions very imprecise.

Then you have the following options to deal with collinearity:

- Use regularization to weight the columns differently (see Regularization (Module 1 Chapter 1)*).
- Use PCA to extract the most important features (see *Principal Component Analysis (Module 1, Chapter 4)*).
- Use domain knowledge to either choose only one of the two columns or create a new feature from both columns and discard the original columns.

In our case, all the correlation coefficients between features are relatively close to 0 (violet in our case). So the first assumption seems to be correct.

The second assumption, that there is a linear relationship between features and sigmoid-transformed probabilities, isn't so easy to verify directly. But you should keep in mind what it means for the outlier problem. The assumption of a linear relationship between sigmoid-transformed probabilities and feature values allows extremely unusual data points to have an oversized influence.

Important: Just like a linear regression, logistic regression is not robust against outliers in the data, see *Robuste Regression (Chapter 5)*. This is one way in which it differs from k-Nearest Neighbors. The k-Nearest-Neighbors classification method is extremely robust, since only the local neighborhood is used for classification. Outliers are most likely extreme values outside the neighborhood and are ignored as a result.

Congratulations: You looked at both assumptions of logistic regression and we're in luck. The first assumption is correct. The second assumption may be incomprehensible to you because we haven't dealt with what sigmoid-transformed probabilities are yet. You'll find out what that means in the next lesson.

Remember:

- Logistic regression assumes that continuous features are not strongly correlated.
- A correlation matrix shows the correlations between continuous features.
