# Deployment on Flask

We first stored the model on joblib.

The library documentation can be read online on <https://joblib.readthedocs.io/en/latest/>

In the joblib library we used the dump() and load() the functions. The dump function is used to create a .joblib file

Graphical user interface, text

Description automatically generated

This .joblib file is the pushed into the load() function which then creates a model which is allowed to predict the sentiment used in a text.

Text

Description automatically generated

Moving on to the deployment on flask.

We import the load function in our app.py file. When used the load function returns the object stored in the .joblib file.

Text

Description automatically generated

We create an html file named index.html where we use forms to get text from our user.

Text

Description automatically generated

Once we have the required text, we pass this onto our app.py file

Text

Description automatically generated

In the image above app.route(‘/’) represents the first page that will be shown to the user, which is where he will enter his text.

The second app.route(‘/get-sentiment’) is where the page will be directed if the user decides he wants to know of the sentiment his tweet suggests. The html and the app.py files are connected using the action keyword in the form. This allows the user to be redirected to the /get-sentiment page. From there the words entered by the user in the form with id “search” are collected in the variable named user.

We then pass the text in an array to the predict() function which then returns another array which includes only one element which is the sentiment of the user.