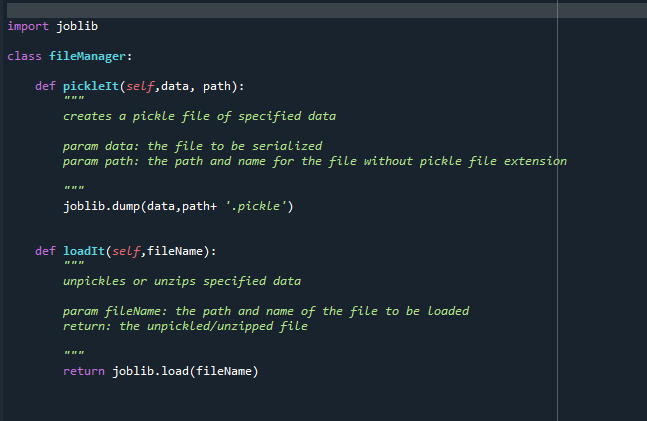
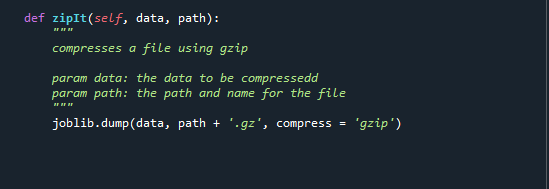
Pipeline serialisation and API integration

When creating the API, we needed a way to integrate the pipeline so that our model could be used to service requests and return predictions on passed in movie plots. To achieve this, we created a file called modelLoader.py that held a set of functions to load and serialise objects. These functions are shown below:



These functions use the joblib module to serialise and load objects. A directory was included in our project repository to store the serialised models.

The first version of the pipeline we serialised was 375 mb, this caused issues with performance and made it difficult to push the pipeline to the project repository so we decided to add a new function to the modelLoader file that also added compression to objects when serialising them. This function is shown below:



After applying this compression to our pipeline, the size of the file reduced to 5.73 mb.

This process of compression was repeated for each version of the pipeline and each new serialised model was stored in the model directory to provide access to the API and any other application that needed to access them.

Below is a snapshot of the model directory from the project repository showing each version of the model:

