**Pre-processing**

We use ColumnTransformer from scikit-learn where MinMaxScaler is used to normalize each data point. Majority of variables have values 0/1 which makes it a sparse dataset. Two-third of the dataset is split into training and one-third is used for evaluating the model. For deployment all the data is used for model training.

**ML Model**

We are using Random Forest Classifier which is an ensembled model built on top of decision trees. Since our data majorly consists of categorical variables, we preferred tree algorithm over other classifiers e.g., Nearest Neighbours and SVM etc.

We treat each act as a label independent from other labels which is why a list of models are trained for the data. Number of model estimators and iterations are kept default as per scikit-learn documentation.

Both the ColumnTransformer and the ML model are pipelined for end-to-end training. This makes it easier to get prediction through the API.

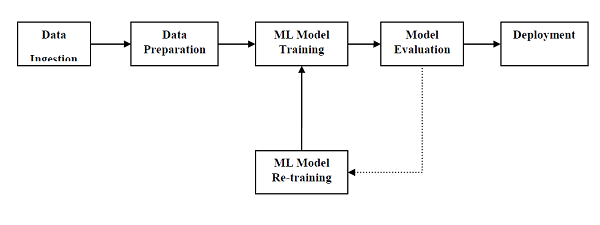


Figure 1: Machine Learning Pipeline (https://www.tutorialspoint.com/machine\_learning\_with\_python)

**Deployment**

Once we have the trained model, we deploy it on localhost through Flask API. For remote tunnelling we are using Ngrok which gives us prediction through all act’s classifier under 2 seconds. Prediction from model can be taken through:

URL : <http://c41fe2352ab4.ngrok.io/api/predict>

Future plans

**AI model improvements:**

* Once we have a substantial amount of user data our existing generated data will start getting replaced with real user’s data. This will give strength to the model as it will begin finding patterns that affect the act’s we are predicting.
* We propose to use collaborative filtering based recommendation engine which can recommend acts based on user’s history and his attributes. We assume that if a user is following a set of acts, he will be interested in another act that is close to his cluster of interest.
* We plan to have the option of adding dynamic tasks in our dataset. These new tasks will be recommended to the users with the help of language models like BERT. We can compute semantic similarity between the task sentence and our data so we can place it under the relevant interests categories. For computing similarity, we will be interested in using pre-trained language models.