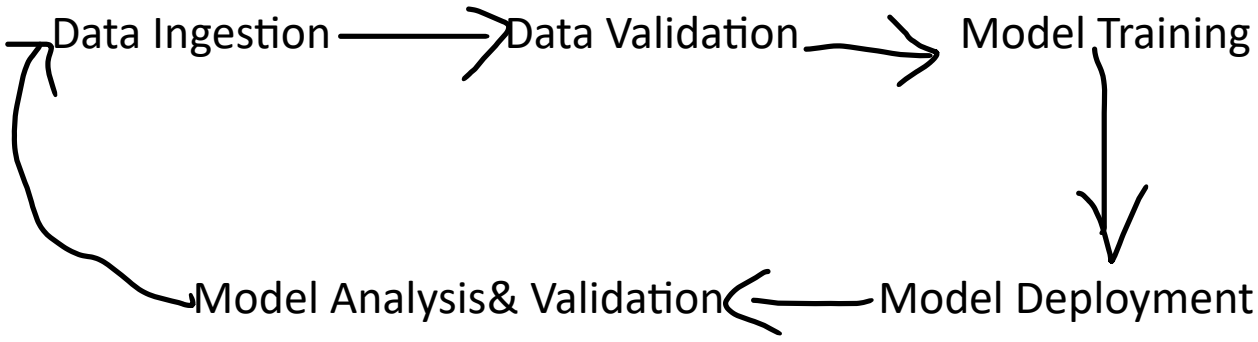
Data Science Pipeline structure



Folder Structures

Data Ingestion

Data Preparation

Data Validation

Model

Model Evaluation

Model Trainer

Utilis

Folder structures example:- *https://medium.com/analytics-vidhya/folder-structure-for-machine-learning-projects-a7e451a8caaa*

Steps in doing the project:-

1. Requirements.txt :- Here you add list of libraries that you will be using in the project
2. Setup.py:- Contains the package details, version, dependencies…. It reads the list from requirements.txt

3. Templete.py:- This creates a folder for the project, where we specify the type of folder/files we might be requiring, just like creating a package.json format

   (.env file must be created which stores the db links etc…)

4. Config.py inside the src:- This creates an environment for the data from the db(.env file) to run in project, we can specify target columns

5. Utils.py:- This is communication between the project and db(Mongodb…)

6. Predictor.py:- It is a resolver file which reads and saves the model,transformer,target files

7. Entity/config_entity.py:- Here we specify the names for files like this Create a TrainingPipelineConfig which returns its results in **artifact** folder Here we define folder/file type that is needed when executed the main file.

```
FILE_NAME="insurance.csv"
TEST_FILE="test.csv"
TRAIN_FILE="train.csv"
TRANSFORMER_OBJECT_FILE="transformer.pkl"
TARGET_ENCODER_OBJECT_FILE= "target_encoder.pkl"
MODEL_FILE="model.pkl"
```

8. Entity/artifact_entity.py:- Here the return type is specified.
9. Components/data_ingestion.py:- This gets the dataset calls both the entity files, and does the function on how the dataset must work
   The code to do EDA, split data into train and test. Create a file for train and test.
10. Components/data_validation.py:-This gets the data from ingestion and sets threshold, pvalues …..
11. Components/data_transformation.py:- This reads from ingestion, does impute fuctions, lable encoding
12. Components/model_trainer.py:- Reads data from transformation, trains the model with algorithms, check overfitting/underfitting, accuracy
13. Components/model_evaluation.py:- Reads from ingestion,transformation,trainer; compares the trained model with the

previous model and returns the best one according to predictions, accracy

14. Components/model_pusher.py:- Reads from transformation and trainer; creates transformer, model, target_encoder files and save. This can be used to deploy in cloud…..
15. App.py:- We create a UI based on the columns in the original dataset and initiate functions of the above