

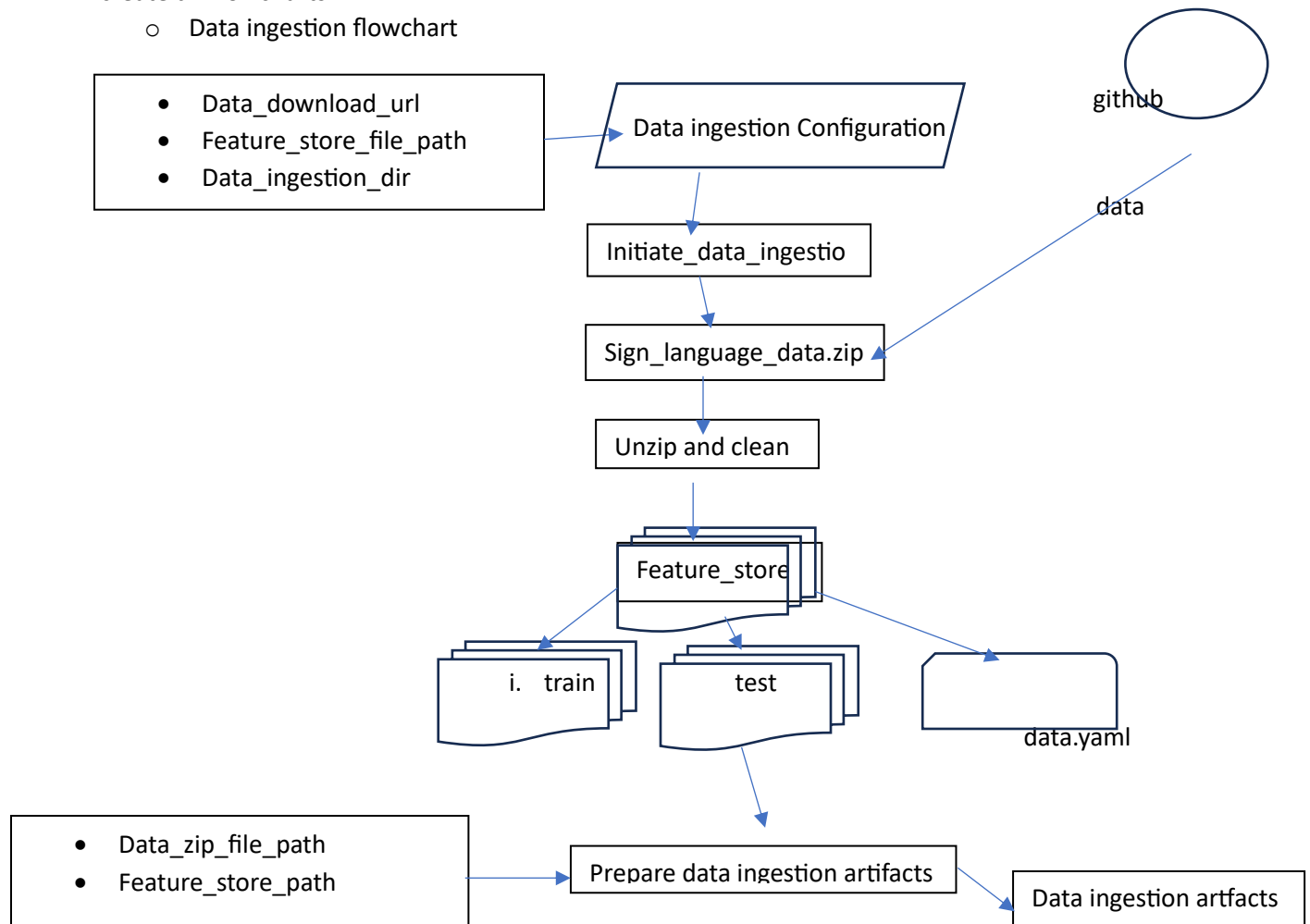
CELL SEGMENTATION (COMPUTER VISION)

STEP1:

- create virtual enviroments
- create readme.md file
- create github repository

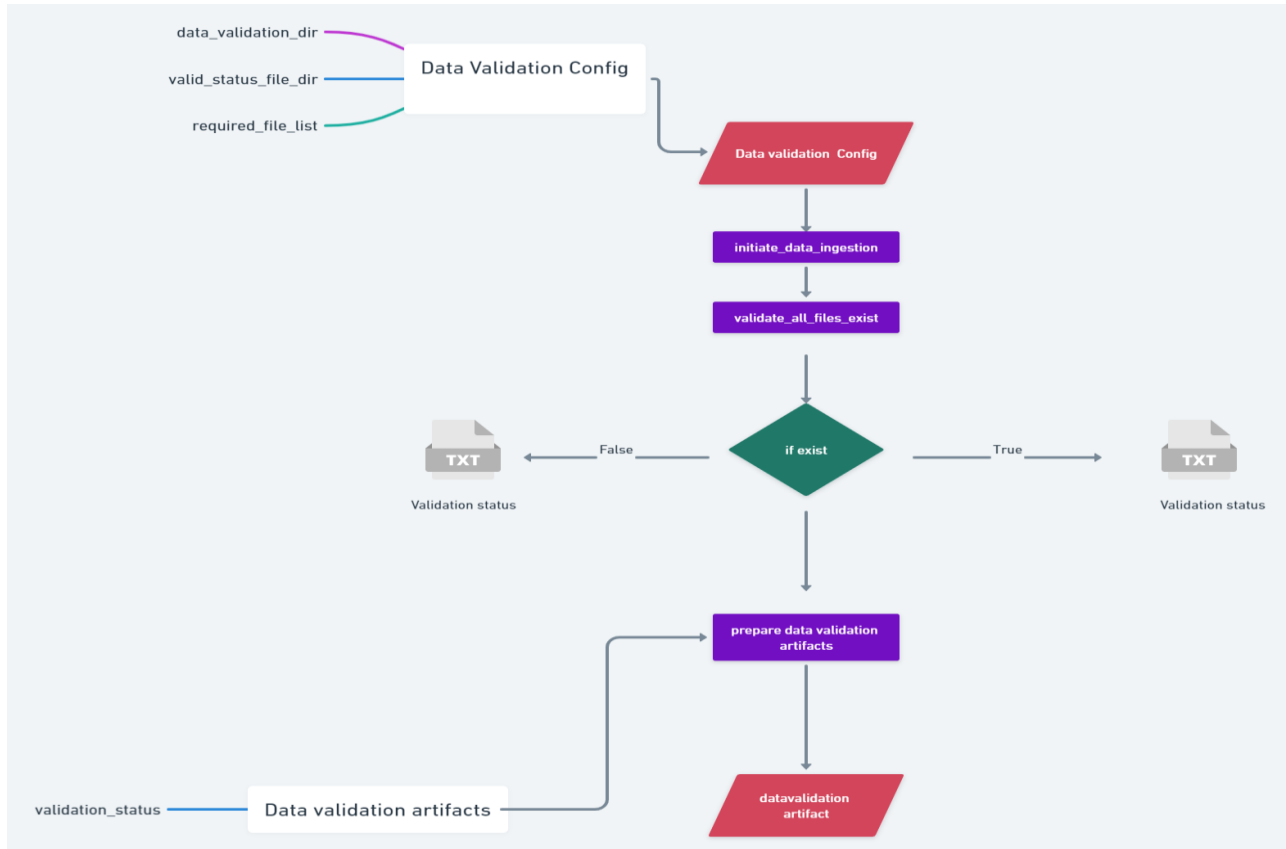
```
//>git init>git add README.md>git commit -m "first commit">git branch -M main>git remote add origin https://github.com/SAMANTA1401/cell_segmentaion.git>git push -u origin main
```
- create .gitignore inside github rep0

```
//>git pull
```
- create dir flowcharts
 - Data ingestion flowchart

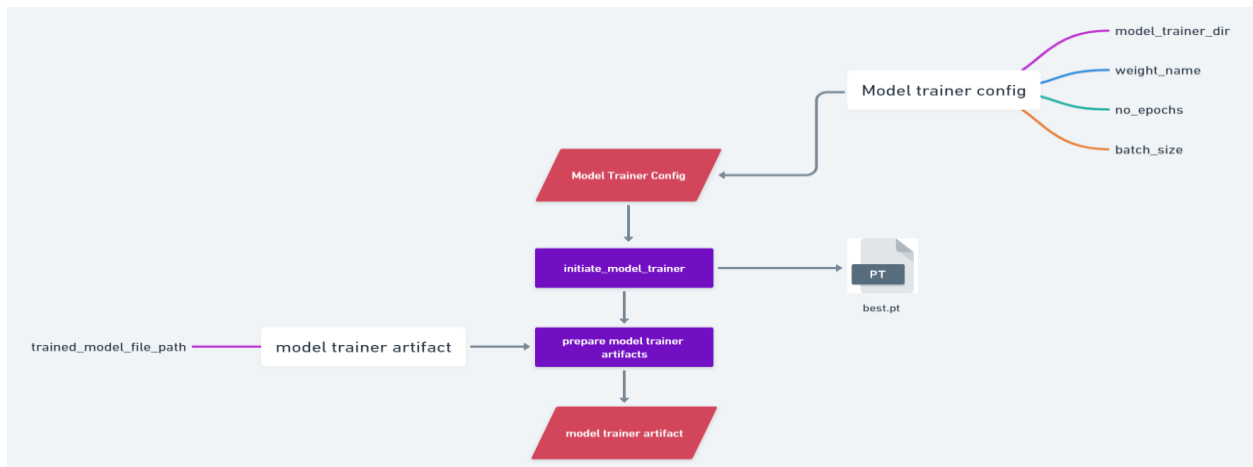


CELL SEGMENTATION (COMPUTER VISION)

○ Data validation flowchart



○ Model trainer flowchart



CELL SEGMENTATION (COMPUTER VISION)

- Create template.py
Or//
- Create exception.__init__.py
- Create logger.__init__.py
- Create research.trials.ipynb

#####

///create data ingestion config:-

- * create constant variables .constant.training_pipeline .__init__.py
-> create DATA INGESTION VAR NAME
- Create entity.config_entity.py
-> create class TrainingPipelineConfig:
-> create class DataIngestionConfig:

///initiate data ingestion:-

- Create components.data_ingestion.py
-> create class DataIngestion:
-> create function download_data():

* //create setup.py

/// install requirements.txt

- Add -e . inside requirements.txt
 - Pip install -r requirements.txt
 - Pip freeze
 - Pip freeze > requirements.txt
- * open components.data_ingestion.py
-> create class DataIngestion:
 - > create extract_zip_file()
 - > create initiate_data_ingestion()
- * create cellsegment.entity.artifacts_entity.py
-> create DataIngestionArtifacts
- * create cellsegmentation.pipeline.training_pipeline.py
-> create class TrainPipeline
 - >create start_data_ingestion()
 - > create run_pipeline()

CELL SEGMENTATION (COMPUTER VISION)

#####

///create data VALIDATION config:-

* open constant variables .constant.training_pipeline .__init__.py

-> create DATA VALIDATION VAR NAME

* open entity.config_entity.py

-> create class DataValidationConfig:

///initiate data validation:-

• create components.data_validation.py

-> create class DataValidation:

-> create validate_all_files_exist()

-> create initiate_data_validation()

* open cellsegment.entity.artifacts_entity.py

-> create DataValidationArtifacts

* create cellsegmentation.pipeline.training_pipeline.py

->create start_data_validation()

-> create run_pipeline()

#####

///create data /model training config:-

* open constant variables .constant.training_pipeline .__init__.py

-> create DATA TRAINER VAR NAME

* open entity.config_entity.py

-> create class ModelTrainerConfig:

* create components.model_trainer.py

-> create class ModelTrainer:

* open cellsegment.entity.artifacts_entity.py

-> create ModelTrainerArtifacts

CELL SEGMENTATION (COMPUTER VISION)

* create components.model_trainer.py

-> create class ModelTrainer:

-> create initiate_model_trainer()

* create cellsegmentation.pipeline.training_pipeline.py

->create start_model_trainer()

-> create run_pipeline()

#####

• create cellsegmentation.pipeline.prediction_pipeline.py

➔ create class Prediction

->create predict()

* create cellsegment.utils.utils_main.py

Create app.py