```python
from sensor.exception import SensorException
from sensor.logger import logging
import os
import sys
from pandas import DataFrame
from sensor.entity.config_entity import DataIngestionConfig
from sensor.entity.artifact_entity import DataIngestionArtifact
from sensor.data_access.sensor_data import SensorData
from sklearn.model_selection import train_test_split

class DataIngestion:
    def __init__(self,data_ingestion_config:DataIngestionConfig):
        try:
            self.data_ingestion_config=data_ingestion_config

        except Exception as e:
            raise SensorException(e,sys)


    def export_data_into_feature_store(self) -> DataFrame:
        """
        Export mongo db collection record as data frame into
feature
        """
        try:
            logging.info("Exporting data from mongodb to feature
store")

            sensor_data = SensorData()

            dataframe =
sensor_data.export_collection_as_dataframe(collection_name=self.d
ata_ingestion_config.collection_name)

            feature_store_file_path =
self.data_ingestion_config.feature_store_file_path

            #creating folder

            dir_path = os.path.dirname(feature_store_file_path)
            os.makedirs(dir_path,exist_ok=True)
```

```
            dataframe.to_csv(feature_store_file_path,index=False,
header=True)
            return dataframe

        except  Exception as e:
            raise  SensorException(e,sys)



    def split_data_as_train_test(self, dataframe: DataFrame) ->
None:
        try:
            train_set, test_set = train_test_split(
                dataframe,
test_size=self.data_ingestion_config.train_test_split_ratio
            )

            logging.info("Performed train test split on the
dataframe")

            logging.info(
                "Exited split_data_as_train_test method of
Data_Ingestion class"
            )

            dir_path =
os.path.dirname(self.data_ingestion_config.training_file_path)

            os.makedirs(dir_path, exist_ok=True)

            logging.info(f"Exporting train and test file path.")

            train_set.to_csv(
                self.data_ingestion_config.training_file_path,
index=False, header=True
            )

            test_set.to_csv(
                self.data_ingestion_config.testing_file_path,
index=False, header=True
```

```
            )

            logging.info(f"Exported train and test file path.")
        except Exception as e:
            raise SensorData(e,sys)


    def initiate_data_ingestion(self) -> DataIngestionArtifact:
        try:
            dataframe = self.export_data_into_feature_store()



            self.split_data_as_train_test(dataframe=dataframe)

            data_ingestion_artifact =
DataIngestionArtifact(trained_file_path=self.data_ingestion_confi
g.training_file_path,
            test_file_path=self.data_ingestion_config.testing_fil
e_path)

            return data_ingestion_artifact

        except Exception as e:
            raise SensorException(e,sys)
```