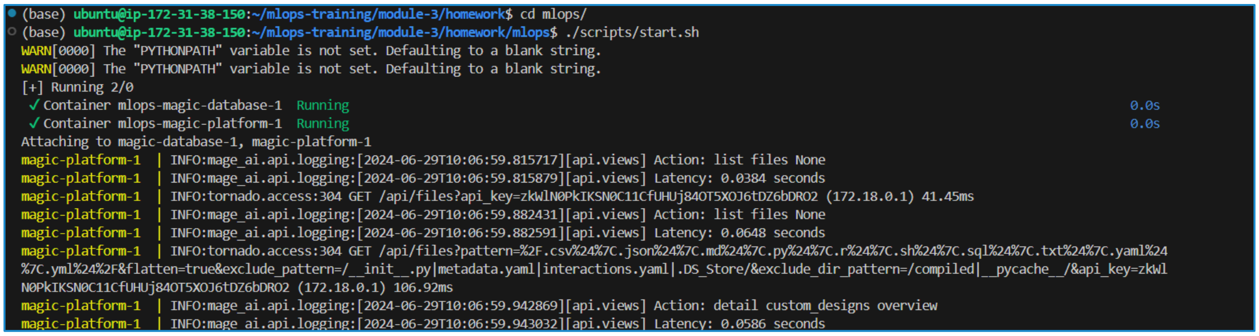
Module 3 Quiz (Mayur Brijwani)

Question 1. Run Mage

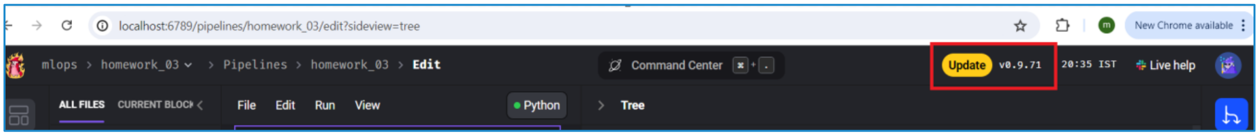
First, let's run Mage with Docker Compose. Follow the quick start guideline.
What's the version of Mage we run?

Ans : v0.9.71

- Run start.sh



```
● (base) ubuntu@ip-172-31-38-150:~/mlops-training/module-3/homework$ cd mlops/
○ (base) ubuntu@ip-172-31-38-150:~/mlops-training/module-3/homework/mlops$ ./scripts/start.sh
  WARN[0000] The "PYTHONPATH" variable is not set. Defaulting to a blank string.
  WARN[0000] The "PYTHONPATH" variable is not set. Defaulting to a blank string.
  [+] Running 2/0
   ✓ Container mlops-magic-database-1  Running                                                              0.0s
   ✓ Container mlops-magic-platform-1  Running                                                              0.0s
  Attaching to magic-database-1, magic-platform-1
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.815717][api.views] Action: list files None
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.815879][api.views] Latency: 0.0384 seconds
  magic-platform-1  | INFO:tornado.access:304 GET /api/files?api_key=zkWlN0PkIKSN0C11CfUHUj84OT5XOJ6tDZ6bDRO2 (172.18.0.1) 41.45ms
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.882431][api.views] Action: list files None
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.882591][api.views] Latency: 0.0648 seconds
  magic-platform-1  | INFO:tornado.access:304 GET /api/files?pattern=%2F.csv%24%7C.json%24%7C.md%24%7C.py%24%7C.r%24%7C.sh%24%7C.sql%24%7C.txt%24%7C.yaml%24
  %7C.yml%24%2F&flatten=true&exclude_pattern=/__init__.py|metadata.yaml|interactions.yaml|.DS_Store/&exclude_dir_pattern=/compiled|__pycache__/&api_key=zkWl
  N0PkIKSN0C11CfUHUj84OT5XOJ6tDZ6bDRO2 (172.18.0.1) 106.92ms
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.942869][api.views] Action: detail custom_designs overview
  magic-platform-1  | INFO:mage_ai.api.logging:[2024-06-29T10:06:59.943032][api.views] Latency: 0.0586 seconds
```

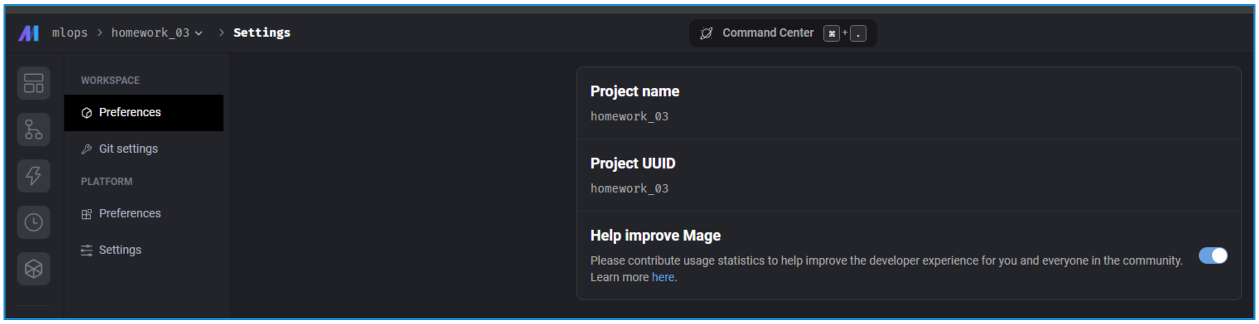


Question 2. Creating a project

Now let's create a new project. We can call it "homework_03", for example.
How many lines are in the created `metadata.yaml` file?

- 35
- 45
- 55
- 65

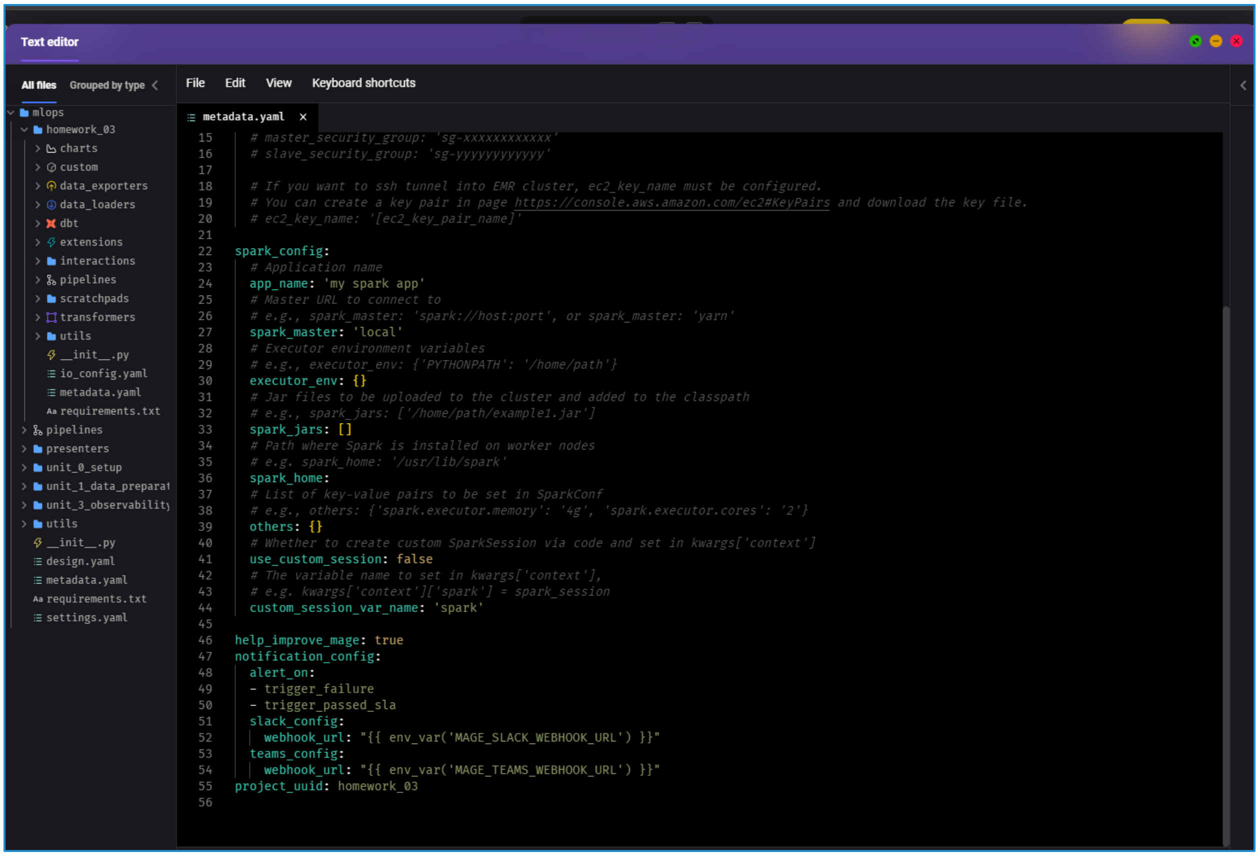
Ans : 55



Project name
homework_03

Project UUID
homework_03

Help improve Mage
Please contribute usage statistics to help improve the developer experience for you and everyone in the community. Learn more here.



```yaml
  # master_security_group: 'sg-xxxxxxxxxxxx'
  # slave_security_group: 'sg-yyyyyyyyyyyy'

  # If you want to ssh tunnel into EMR cluster, ec2_key_name must be configured.
  # You can create a key pair in page https://console.aws.amazon.com/ec2#KeyPairs and download the key file.
  # ec2_key_name: '[ec2_key_pair_name]'

spark_config:
  # Application name
  app_name: 'my spark app'
  # Master URL to connect to
  # e.g., spark_master: 'spark://host:port', or spark_master: 'yarn'
  spark_master: 'local'
  # Executor environment variables
  # e.g., executor_env: {'PYTHONPATH': '/home/path'}
  executor_env: {}
  # Jar files to be uploaded to the cluster and added to the classpath
  # e.g., spark_jars: ['/home/path/example1.jar']
  spark_jars: []
  # Path where Spark is installed on worker nodes
  # e.g. spark_home: '/usr/lib/spark'
  spark_home:
  # List of key-value pairs to be set in SparkConf
  # e.g., others: {'spark.executor.memory': '4g', 'spark.executor.cores': '2'}
  others: {}
  # Whether to create custom SparkSession via code and set in kwargs['context']
  use_custom_session: false
  # The variable name to set in kwargs['context'],
  # e.g. kwargs['context']['spark'] = spark_session
  custom_session_var_name: 'spark'

help_improve_mage: true
notification_config:
  alert_on:
  - trigger_failure
  - trigger_passed_sla
  slack_config:
    webhook_url: "{{ env_var('MAGE_SLACK_WEBHOOK_URL') }}"
  teams_config:
    webhook_url: "{{ env_var('MAGE_TEAMS_WEBHOOK_URL') }}"
project_uuid: homework_03

```

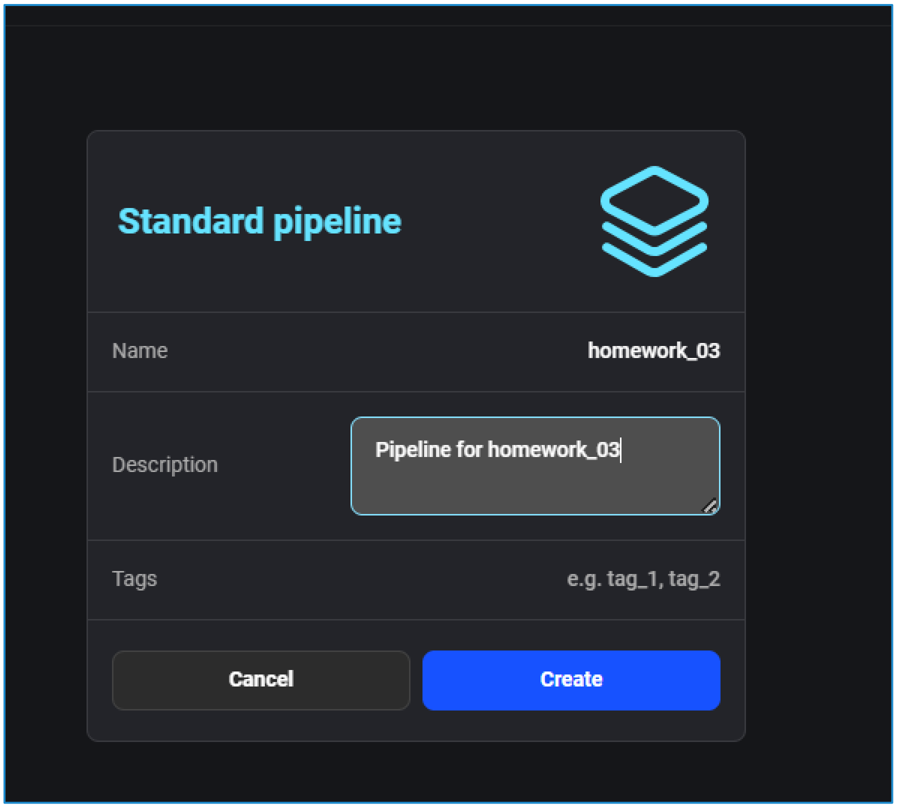
Question 3. Creating a pipeline

Let's create an ingestion code block.
In this block, we will read the March 2023 Yellow taxi trips data.
How many records did we load?

- 3,003,766
- 3,203,766
- 3,403,766
- 3,603,766

Ans : 3,403,766

- Create Standard Pipeline



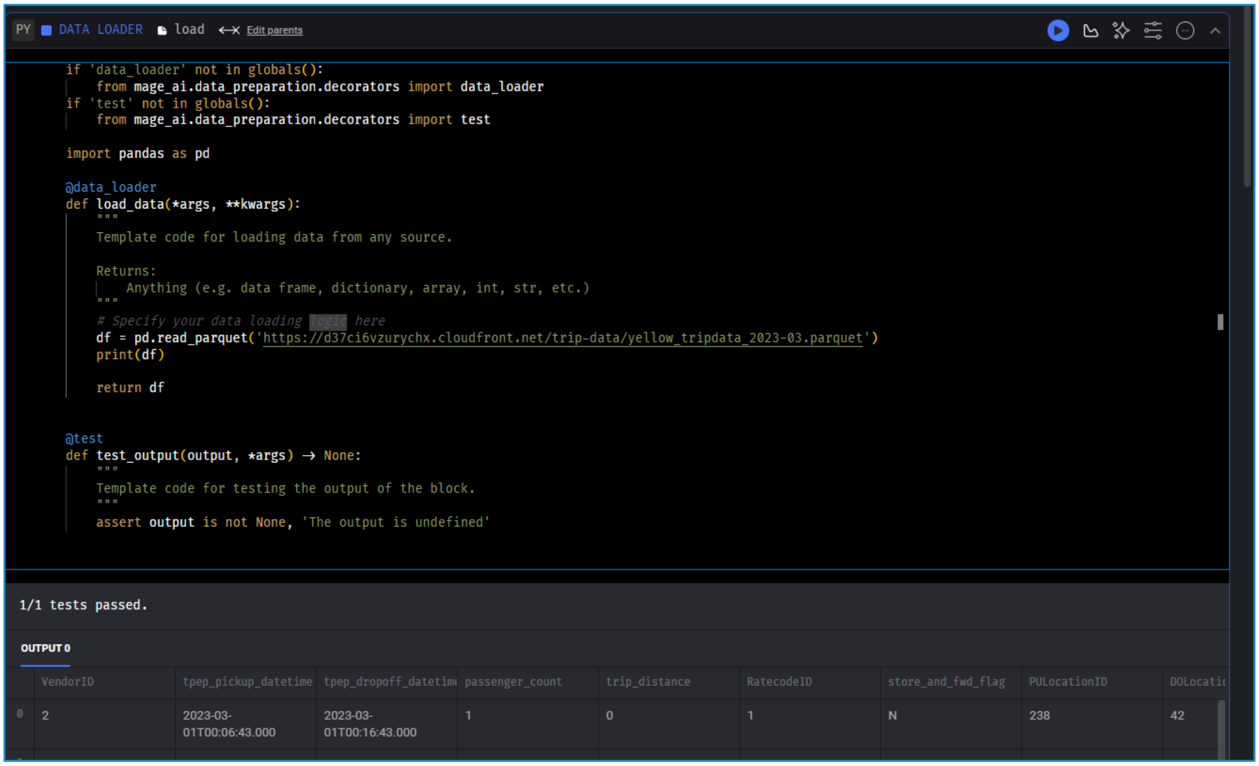
Standard pipeline

Name: homework_03

Description: Pipeline for homework_03

Tags: e.g. tag_1, tag_2

- Create Ingestion block to load data

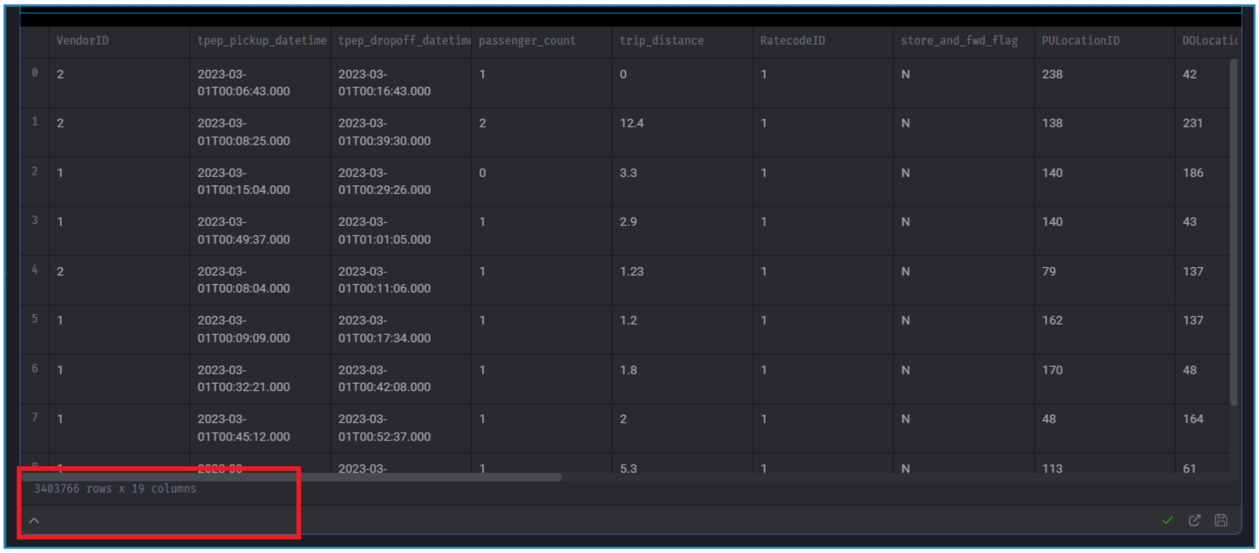


```python
if 'data_loader' not in globals():
    from mage_ai.data_preparation.decorators import data_loader
if 'test' not in globals():
    from mage_ai.data_preparation.decorators import test

import pandas as pd

@data_loader
def load_data(*args, **kwargs):
    """
    Template code for loading data from any source.

    Returns:
        Anything (e.g. data frame, dictionary, array, int, str, etc.)
    """
    # Specify your data loading logic here
    df = pd.read_parquet('https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2023-03.parquet')
    print(df)

    return df


@test
def test_output(output, *args) -> None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```

```
1/1 tests passed.
```

OUTPUT 0

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetim	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocati
0	2	2023-03-01T00:06:43.000	2023-03-01T00:16:43.000	1	0	1	N	238	42



	VendorID	tpep_pickup_datetime	tpep_dropoff_datetim	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLocati
0	2	2023-03-01T00:06:43.000	2023-03-01T00:16:43.000	1	0	1	N	238	42
1	2	2023-03-01T00:08:25.000	2023-03-01T00:39:30.000	2	12.4	1	N	138	231
2	1	2023-03-01T00:15:04.000	2023-03-01T00:29:26.000	0	3.3	1	N	140	186
3	1	2023-03-01T00:49:37.000	2023-03-01T01:01:05.000	1	2.9	1	N	140	43
4	2	2023-03-01T00:08:04.000	2023-03-01T00:11:06.000	1	1.23	1	N	79	137
5	1	2023-03-01T00:09:09.000	2023-03-01T00:17:34.000	1	1.2	1	N	162	137
6	1	2023-03-01T00:32:21.000	2023-03-01T00:42:08.000	1	1.8	1	N	170	48
7	1	2023-03-01T00:45:12.000	2023-03-01T00:52:37.000	1	2	1	N	48	164
8	1	2023-03-	2023-03-	1	5.3	1	N	113	61

```
3403766 rows x 19 columns
```

Question 4. Data preparation

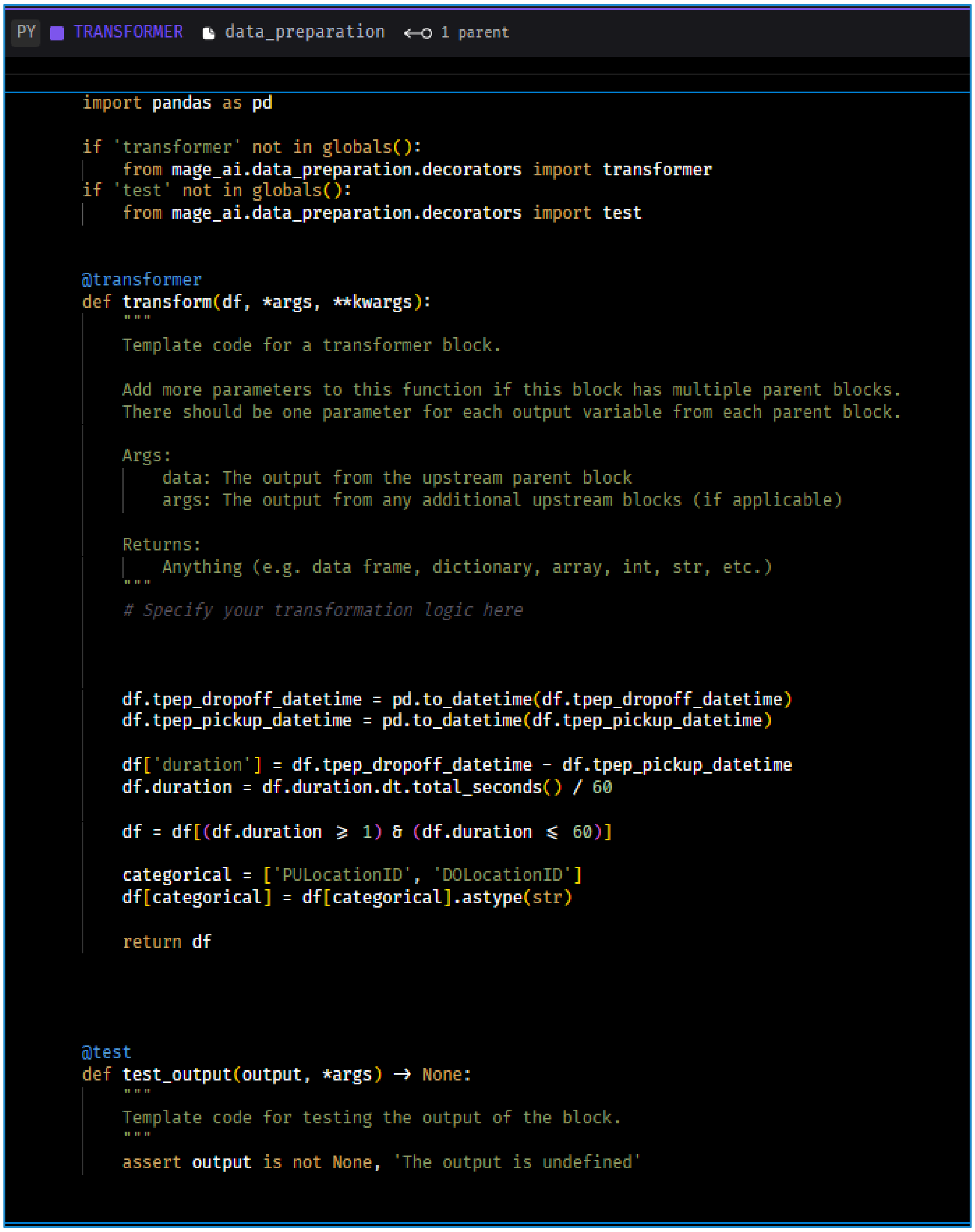
Let's use the same logic for preparing the data we used previously. We will need to create a transformer code block and put this code there.
This is what we used (adjusted for yellow dataset):

```
def read_dataframe(filename):
    df = pd.read_parquet(filename)

    df.tpep_dropoff_datetime = pd.to_datetime(df.tpep_dropoff_datetime)
    df.tpep_pickup_datetime = pd.to_datetime(df.tpep_pickup_datetime)

    df['duration'] = df.tpep_dropoff_datetime - df.tpep_pickup_datetime
    df.duration = df.duration.dt.total_seconds() / 60

    df = df[(df.duration >= 1) & (df.duration <= 60)]

    categorical = ['PULocationID', 'DOLocationID']
    df[categorical] = df[categorical].astype(str)

    return df
```

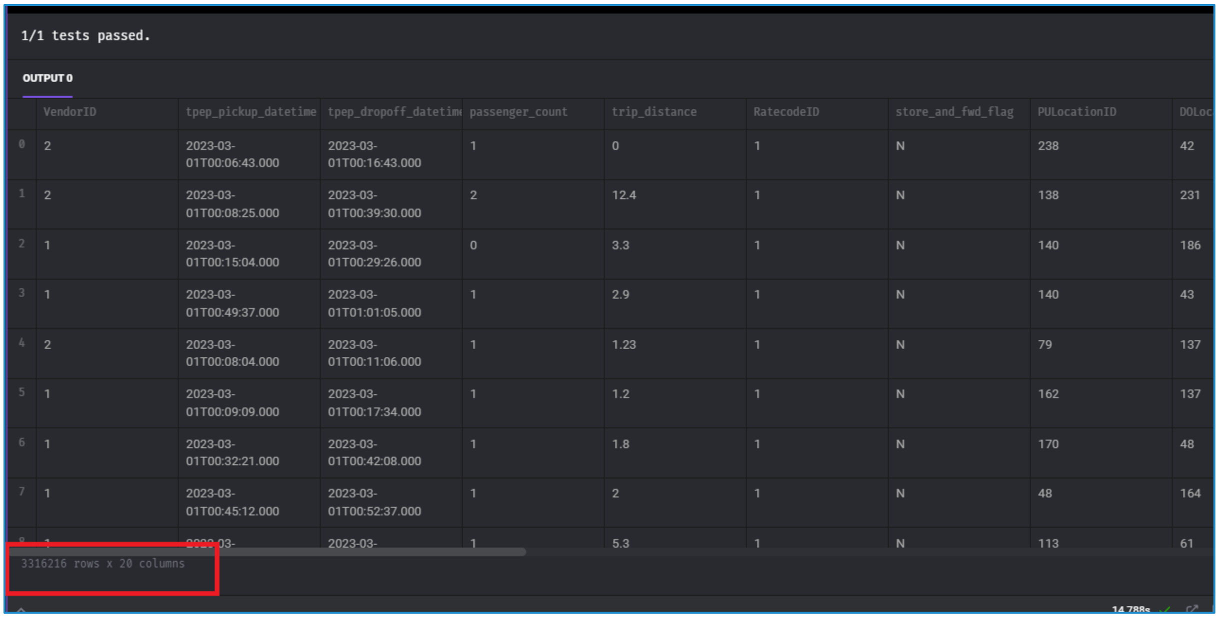
Let's adjust it and apply to the data we loaded in question 3.
What's the size of the result?

- 2,903,766
- 3,103,766
- 3,316,216
- 3,503,766

Ans : 3,316,216



```python
import pandas as pd

if 'transformer' not in globals():
    from mage_ai.data_preparation.decorators import transformer
if 'test' not in globals():
    from mage_ai.data_preparation.decorators import test


@transformer
def transform(df, *args, **kwargs):
    """
    Template code for a transformer block.

    Add more parameters to this function if this block has multiple parent blocks.
    There should be one parameter for each output variable from each parent block.

    Args:
        data: The output from the upstream parent block
        args: The output from any additional upstream blocks (if applicable)

    Returns:
        Anything (e.g. data frame, dictionary, array, int, str, etc.)
    """
    # Specify your transformation logic here



    df.tpep_dropoff_datetime = pd.to_datetime(df.tpep_dropoff_datetime)
    df.tpep_pickup_datetime = pd.to_datetime(df.tpep_pickup_datetime)

    df['duration'] = df.tpep_dropoff_datetime - df.tpep_pickup_datetime
    df.duration = df.duration.dt.total_seconds() / 60

    df = df[(df.duration >= 1) & (df.duration <= 60)]

    categorical = ['PULocationID', 'DOLocationID']
    df[categorical] = df[categorical].astype(str)

    return df




@test
def test_output(output, *args) -> None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```



```
1/1 tests passed.
```

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetim	passenger_count	trip_distance	RatecodeID	store_and_fwd_flag	PULocationID	DOLoc
0	2	2023-03-01T00:06:43.000	2023-03-01T00:16:43.000	1	0	1	N	238	42
1	2	2023-03-01T00:08:25.000	2023-03-01T00:39:30.000	2	12.4	1	N	138	231
2	1	2023-03-01T00:15:04.000	2023-03-01T00:29:26.000	0	3.3	1	N	140	186
3	1	2023-03-01T00:49:37.000	2023-03-01T01:01:05.000	1	2.9	1	N	140	43
4	2	2023-03-01T00:08:04.000	2023-03-01T00:11:06.000	1	1.23	1	N	79	137
5	1	2023-03-01T00:09:09.000	2023-03-01T00:17:34.000	1	1.2	1	N	162	137
6	1	2023-03-01T00:32:21.000	2023-03-01T00:42:08.000	1	1.8	1	N	170	48
7	1	2023-03-01T00:45:12.000	2023-03-01T00:52:37.000	1	2	1	N	48	164
8	1	2023-03-	2023-03-	1	5.3	1	N	113	61

```
3316216 rows x 20 columns
```

Question 5. Train a model

We will now train a linear regression model using the same code as in homework 1.

- Fit a dict vectorizer.
- Train a linear regression with default parameters.
- Use pick up and drop off locations separately, don't create a combination feature.

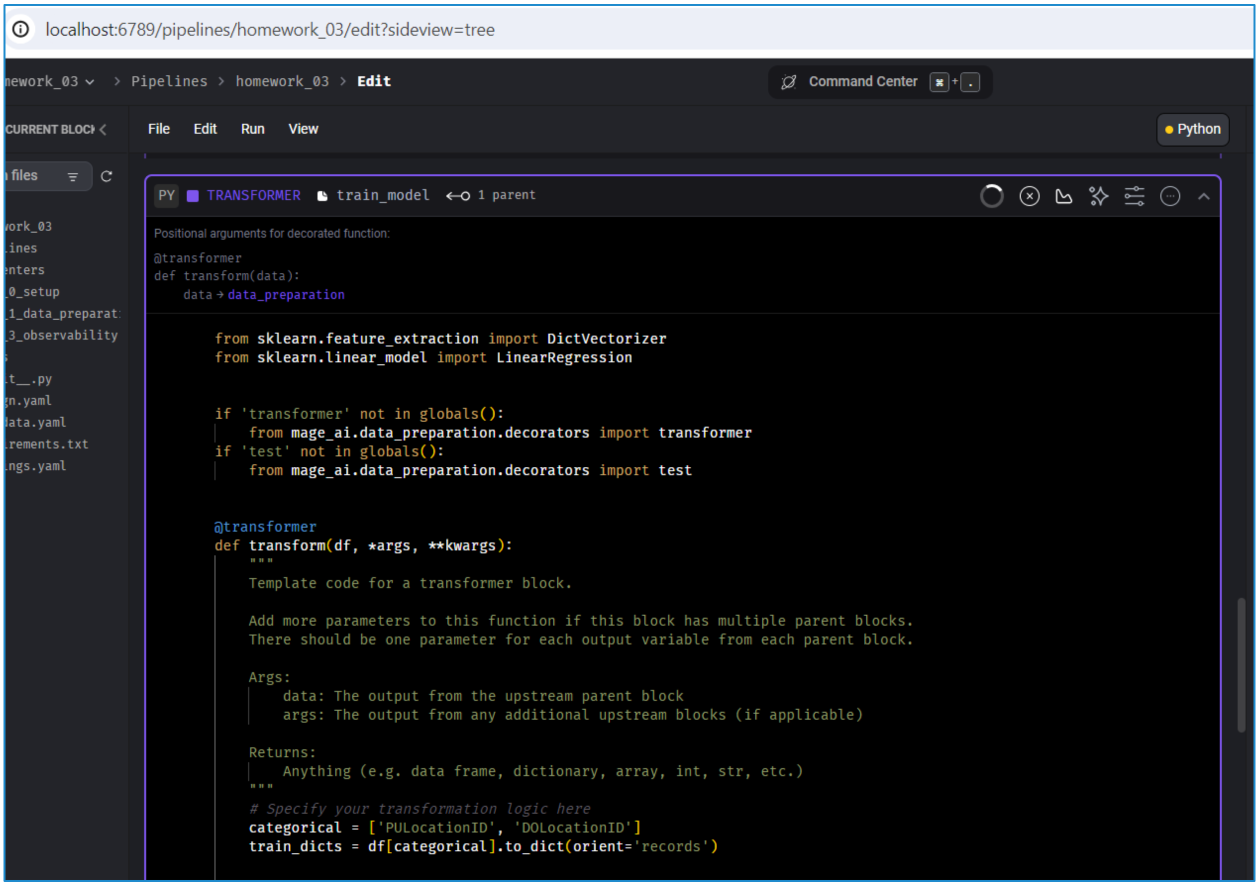
Let's now use it in the pipeline. We will need to create another transformation block, and return both the dict vectorizer and the model.

What's the intercept of the model?

Hint: print the `intercept_` field in the code block

- 21.77
- 24.77
- 27.77
- 31.77

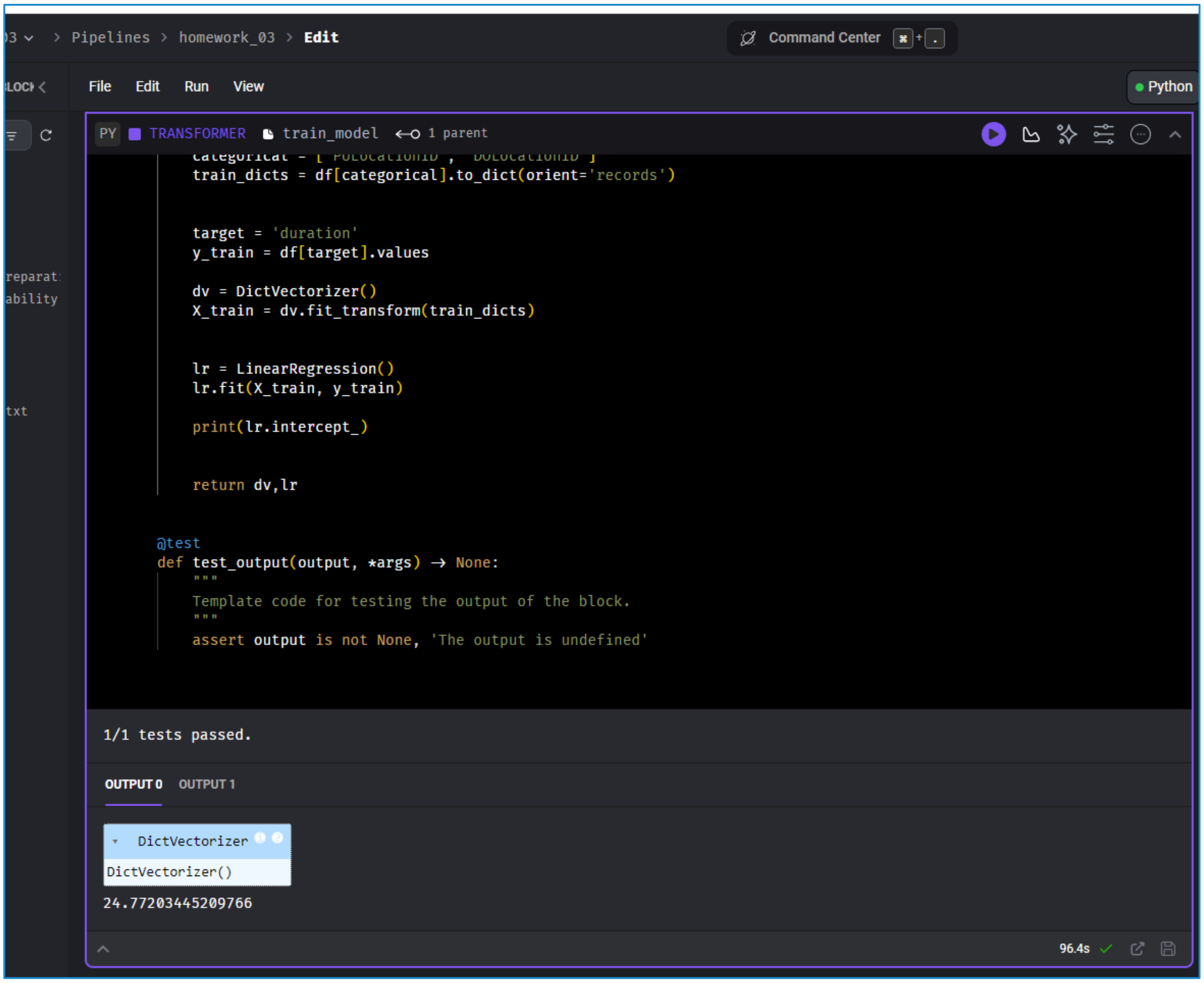
Ans : 24.77



Positional arguments for decorated function:

```
@transformer
def transform(data):
    data → data_preparation
```

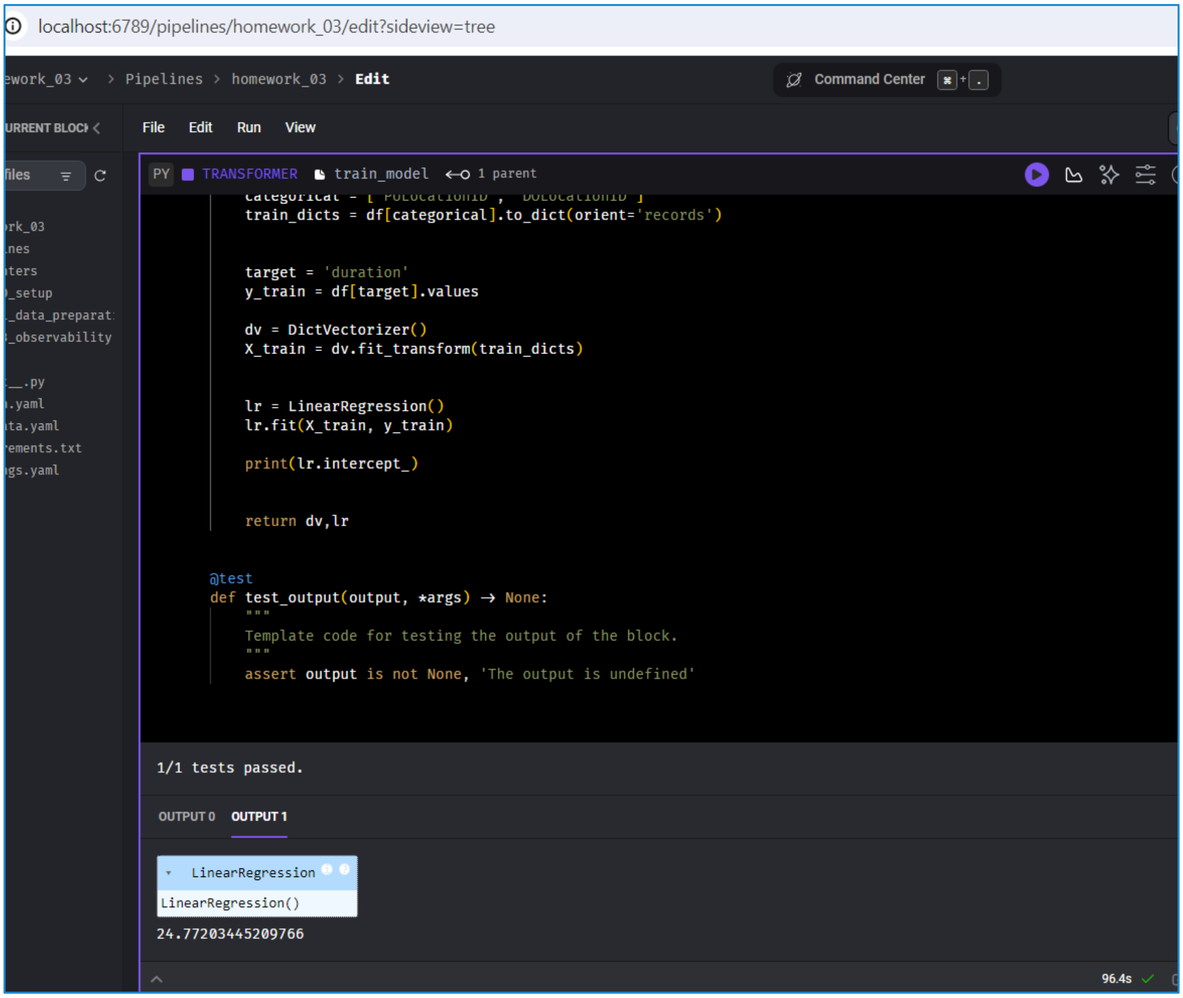
```
from sklearn.feature_extraction import DictVectorizer
from sklearn.linear_model import LinearRegression


if 'transformer' not in globals():
    from mage_ai.data_preparation.decorators import transformer
if 'test' not in globals():
    from mage_ai.data_preparation.decorators import test


@transformer
def transform(df, *args, **kwargs):
    """
    Template code for a transformer block.

    Add more parameters to this function if this block has multiple parent blocks.
    There should be one parameter for each output variable from each parent block.

    Args:
        data: The output from the upstream parent block
        args: The output from any additional upstream blocks (if applicable)

    Returns:
        Anything (e.g. data frame, dictionary, array, int, str, etc.)
    """
    # Specify your transformation logic here
    categorical = ['PULocationID', 'DOLocationID']
    train_dicts = df[categorical].to_dict(orient='records')
```



```
    categorical = ['PULocationID', 'DOLocationID']
    train_dicts = df[categorical].to_dict(orient='records')


    target = 'duration'
    y_train = df[target].values

    dv = DictVectorizer()
    X_train = dv.fit_transform(train_dicts)


    lr = LinearRegression()
    lr.fit(X_train, y_train)

    print(lr.intercept_)


    return dv,lr


@test
def test_output(output, *args) → None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```

```
1/1 tests passed.
```

```
DictVectorizer
DictVectorizer()
24.77203445209766
```



```
    categorical = ['PULocationID', 'DOLocationID']
    train_dicts = df[categorical].to_dict(orient='records')


    target = 'duration'
    y_train = df[target].values

    dv = DictVectorizer()
    X_train = dv.fit_transform(train_dicts)


    lr = LinearRegression()
    lr.fit(X_train, y_train)

    print(lr.intercept_)


    return dv,lr


@test
def test_output(output, *args) -> None:
    """
    Template code for testing the output of the block.
    """
    assert output is not None, 'The output is undefined'
```

```
1/1 tests passed.
```

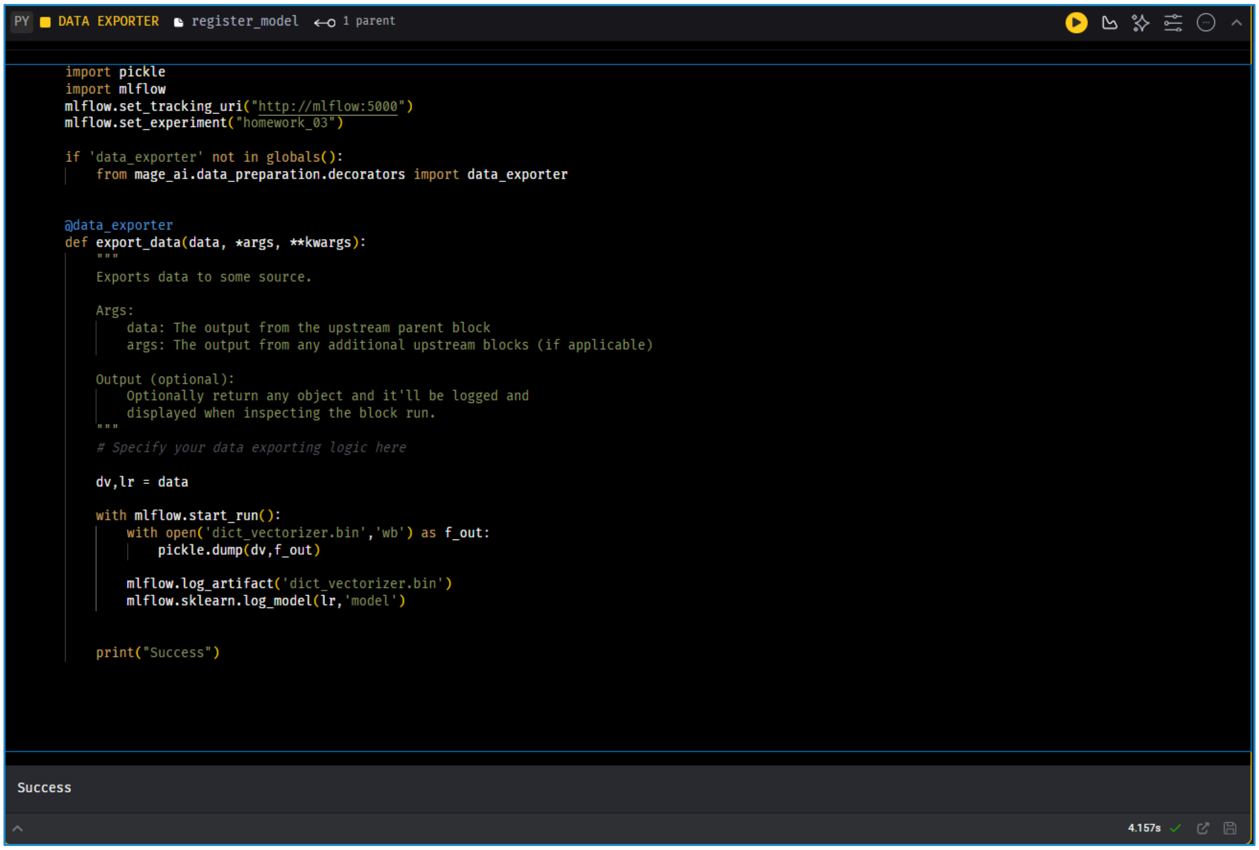
```
LinearRegression
LinearRegression()
24.77203445209766
```

Question 6. Register the model

The model is trained, so let's save it with MLFlow.
Find the logged model, and find MLModel file. What's the size of the model? (`model_size_bytes` field):

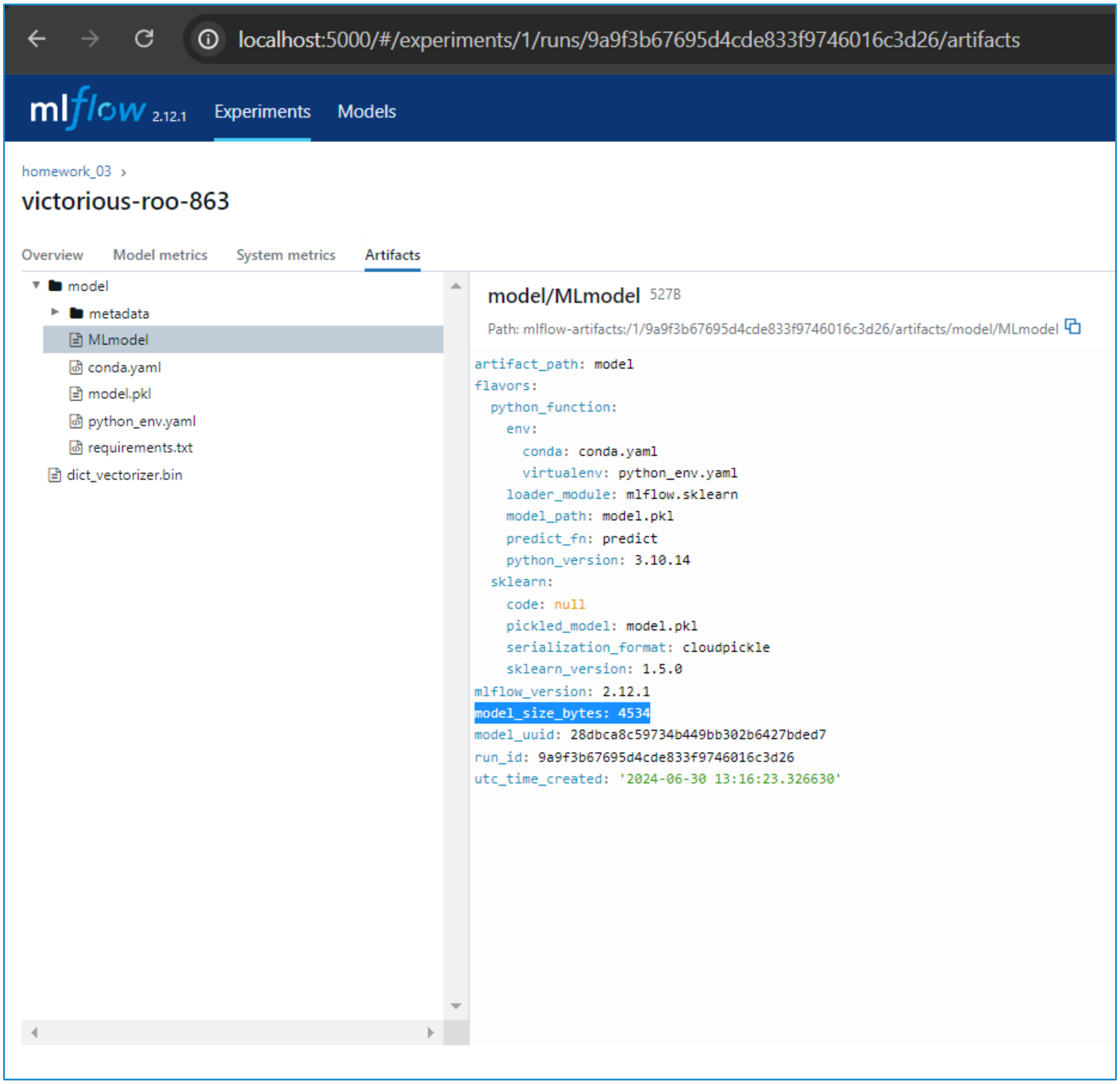
- 14,534
- 9,534
- 4,534
- 1,534

Ans: 4,534



```python
import pickle
import mlflow
mlflow.set_tracking_uri("http://mlflow:5000")
mlflow.set_experiment("homework_03")

if 'data_exporter' not in globals():
    from mage_ai.data_preparation.decorators import data_exporter


@data_exporter
def export_data(data, *args, **kwargs):
    """
    Exports data to some source.

    Args:
        data: The output from the upstream parent block
        args: The output from any additional upstream blocks (if applicable)

    Output (optional):
        Optionally return any object and it'll be logged and
        displayed when inspecting the block run.
    """
    # Specify your data exporting logic here

    dv,lr = data

    with mlflow.start_run():
        with open('dict_vectorizer.bin','wb') as f_out:
            pickle.dump(dv,f_out)

        mlflow.log_artifact('dict_vectorizer.bin')
        mlflow.sklearn.log_model(lr,'model')


    print("Success")
```

```
Success
```


localhost:5000/#/experiments/1/runs/9a9f3b67695d4cde833f9746016c3d26/artifacts
mlflow 2.12.1
Experiments
Models
homework_03 ›

victorious-roo-863

model/MLmodel 527B

Path: mlflow-artifacts:/1/9a9f3b67695d4cde833f9746016c3d26/artifacts/model/MLmodel

```
artifact_path: model
flavors:
  python_function:
    env:
      conda: conda.yaml
      virtualenv: python_env.yaml
    loader_module: mlflow.sklearn
    model_path: model.pkl
    predict_fn: predict
    python_version: 3.10.14
  sklearn:
    code: null
    pickled_model: model.pkl
    serialization_format: cloudpickle
    sklearn_version: 1.5.0
mlflow_version: 2.12.1
model_size_bytes: 4534
model_uuid: 28dbca8c59734b449bb302b6427bded7
run_id: 9a9f3b67695d4cde833f9746016c3d26
utc_time_created: '2024-06-30 13:16:23.326630'
```