Used Cars Market Analysis

Midterm report

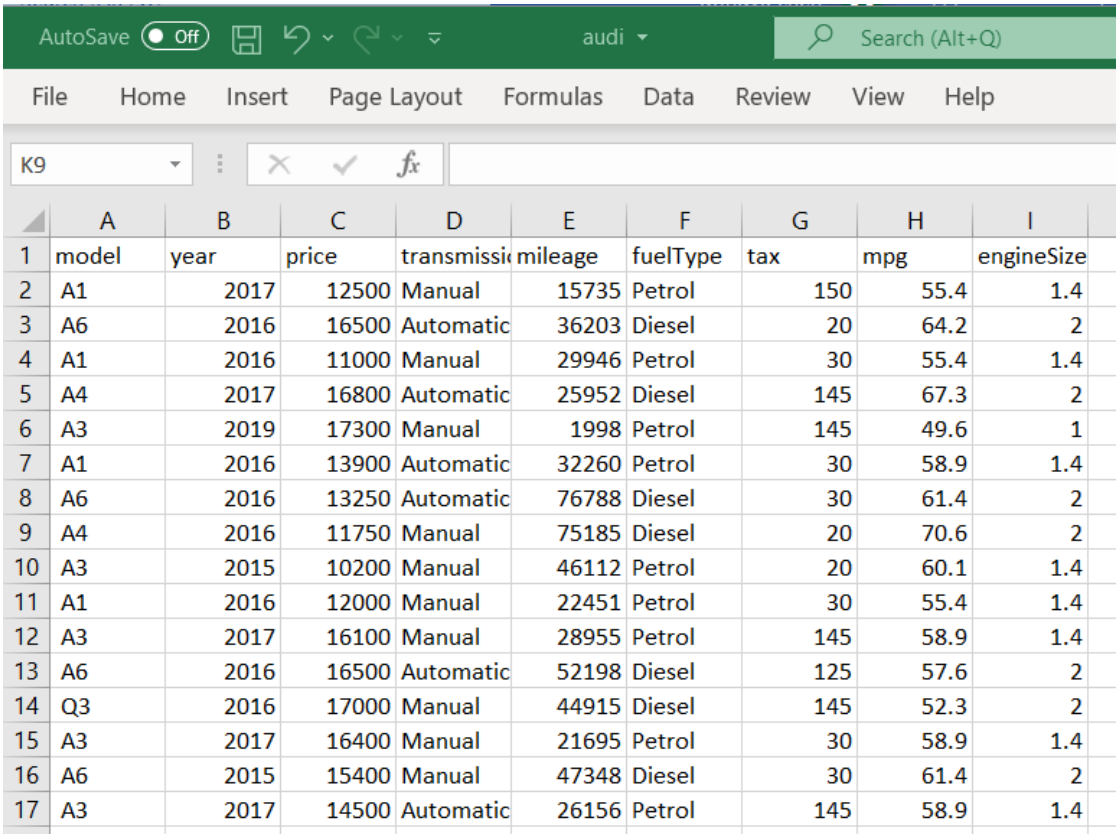
Our Progress

We are currently on track for our milestone, and we are ready to create an application with the functions we write. Currently, 2 different EDFS are built with Firebase. One EDFS is based on Firebase Python SDK and the other is based on Restful requests through python. Partition-based map reduce functions are also completed.

Explanations of Our Original Dataset

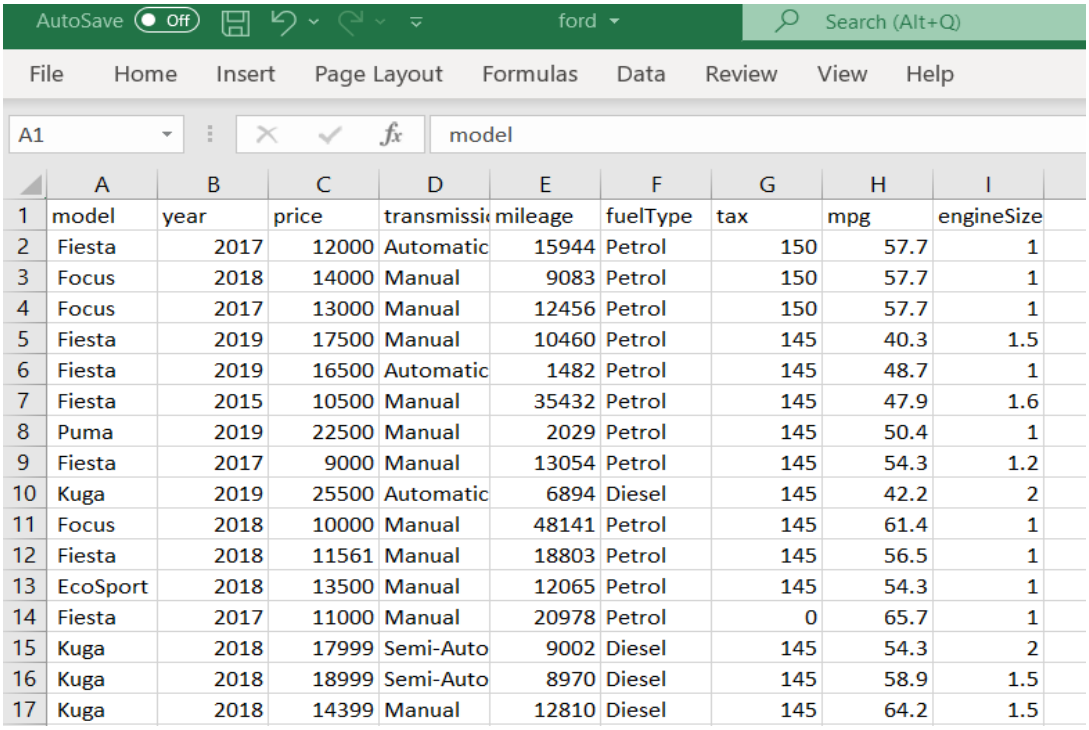
We have three csv datasets, Audi, Ford, and Toyota. Each of them contains nine columns: 1. Model, 2. Year, 3. Price, 4. Transmission, 5.mileage, 6. fuelType, 7. Tax, 8. mpg, and 9. engineSize. And there are about 10,000 records in each csv file. Below are sample data from the datasets:

Audi.csv



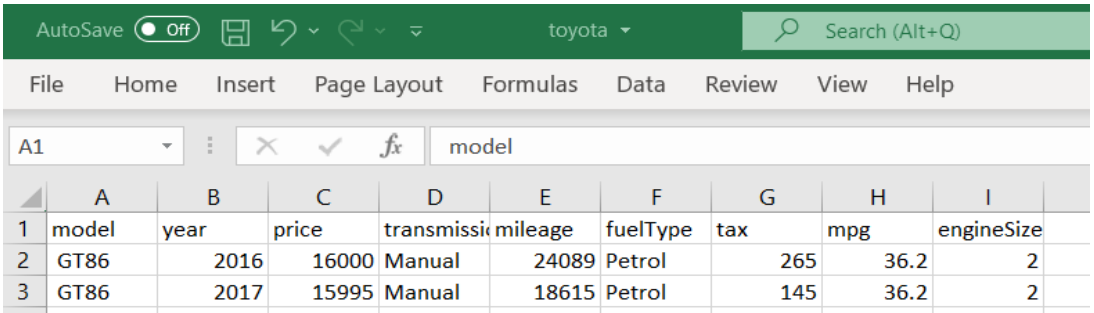
	A	B	C	D	E	F	G	H	I
1	model	year	price	transmissi	mileage	fuelType	tax	mpg	engineSize
2	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
3	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2
4	A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
5	A4	2017	16800	Automatic	25952	Diesel	145	67.3	2
6	A3	2019	17300	Manual	1998	Petrol	145	49.6	1
7	A1	2016	13900	Automatic	32260	Petrol	30	58.9	1.4
8	A6	2016	13250	Automatic	76788	Diesel	30	61.4	2
9	A4	2016	11750	Manual	75185	Diesel	20	70.6	2
10	A3	2015	10200	Manual	46112	Petrol	20	60.1	1.4
11	A1	2016	12000	Manual	22451	Petrol	30	55.4	1.4
12	A3	2017	16100	Manual	28955	Petrol	145	58.9	1.4
13	A6	2016	16500	Automatic	52198	Diesel	125	57.6	2
14	Q3	2016	17000	Manual	44915	Diesel	145	52.3	2
15	A3	2017	16400	Manual	21695	Petrol	30	58.9	1.4
16	A6	2015	15400	Manual	47348	Diesel	30	61.4	2
17	A3	2017	14500	Automatic	26156	Petrol	145	58.9	1.4

ford.csv



	A	B	C	D	E	F	G	H	I
1	model	year	price	transmissi	mileage	fuelType	tax	mpg	engineSize
2	Fiesta	2017	12000	Automatic	15944	Petrol	150	57.7	1
3	Focus	2018	14000	Manual	9083	Petrol	150	57.7	1
4	Focus	2017	13000	Manual	12456	Petrol	150	57.7	1
5	Fiesta	2019	17500	Manual	10460	Petrol	145	40.3	1.5
6	Fiesta	2019	16500	Automatic	1482	Petrol	145	48.7	1
7	Fiesta	2015	10500	Manual	35432	Petrol	145	47.9	1.6
8	Puma	2019	22500	Manual	2029	Petrol	145	50.4	1
9	Fiesta	2017	9000	Manual	13054	Petrol	145	54.3	1.2
10	Kuga	2019	25500	Automatic	6894	Diesel	145	42.2	2
11	Focus	2018	10000	Manual	48141	Petrol	145	61.4	1
12	Fiesta	2018	11561	Manual	18803	Petrol	145	56.5	1
13	EcoSport	2018	13500	Manual	12065	Petrol	145	54.3	1
14	Fiesta	2017	11000	Manual	20978	Petrol	0	65.7	1
15	Kuga	2018	17999	Semi-Auto	9002	Diesel	145	54.3	2
16	Kuga	2018	18999	Semi-Auto	8970	Diesel	145	58.9	1.5
17	Kuga	2018	14399	Manual	12810	Diesel	145	64.2	1.5

toyota.csv



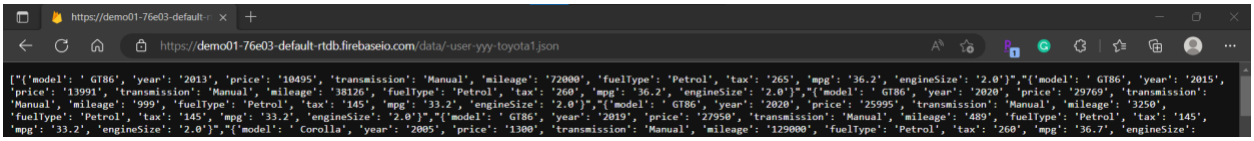
	A	B	C	D	E	F	G	H	I
1	model	year	price	transmissi	mileage	fuelType	tax	mpg	engineSize
2	GT86	2016	16000	Manual	24089	Petrol	265	36.2	2
3	GT86	2017	15995	Manual	18615	Petrol	145	36.2	2
4	GT86	2015	13998	Manual	27469	Petrol	265	36.2	2
5	GT86	2017	18998	Manual	14736	Petrol	150	36.2	2
6	GT86	2017	17498	Manual	36284	Petrol	145	36.2	2
7	GT86	2017	15998	Manual	26919	Petrol	260	36.2	2
8	GT86	2017	18522	Manual	10456	Petrol	145	36.2	2
9	GT86	2017	18995	Manual	12340	Petrol	145	36.2	2
10	GT86	2020	27998	Manual	516	Petrol	150	33.2	2
11	GT86	2016	13990	Manual	37999	Petrol	265	36.2	2
12	GT86	2013	10495	Manual	72000	Petrol	265	36.2	2
13	GT86	2017	17990	Manual	12597	Petrol	145	36.2	2
14	GT86	2017	16995	Manual	36100	Petrol	145	36.2	2
15	GT86	2019	23995	Manual	995	Petrol	145	33.2	2
16	GT86	2018	18498	Manual	35228	Petrol	145	36.2	2
17	GT86	2019	23980	Manual	1751	Petrol	145	33.2	2

Examples of Our Dataset in Firebase (two implementations)

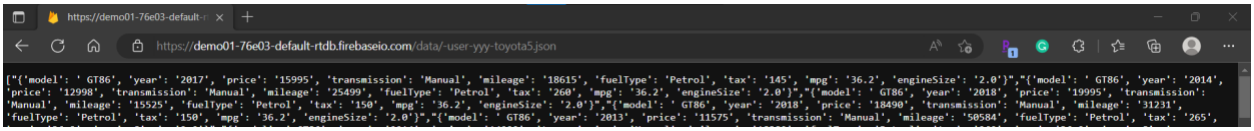
1. implementation 1

toyota.csv -> firebase -> 8 partitions

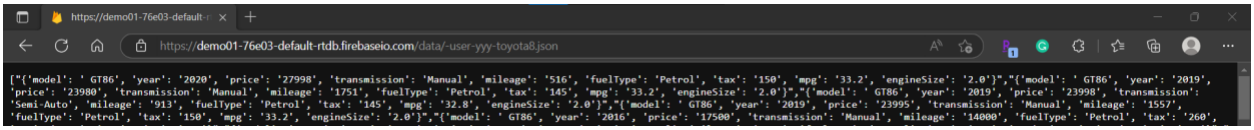
Partition 1:



Partition 5:



Partition 8

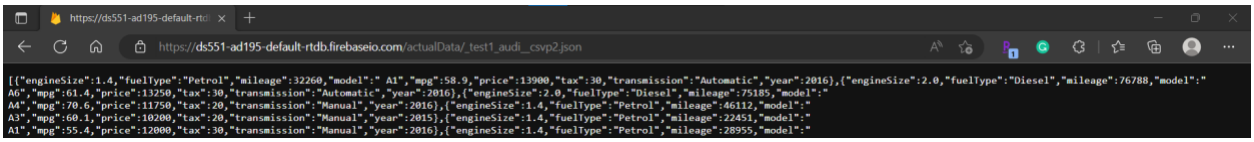


Partition 10: (out of range)

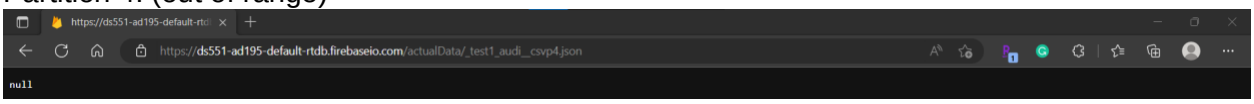
2. Implementation 2

audi.csv -> firebase -> 2 partitions

Partition 2:

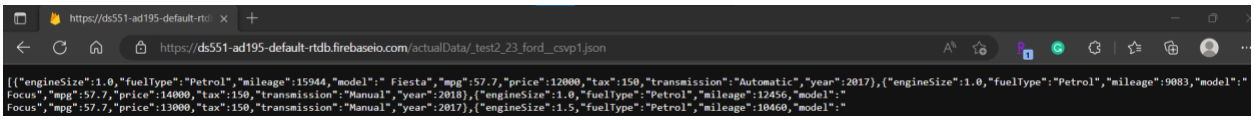


Partition 4: (out of range)

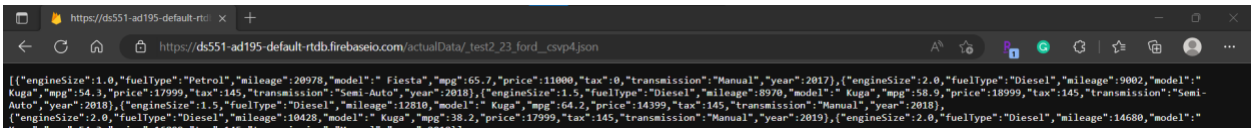


ford.csv -> firebase -> 4 partitions

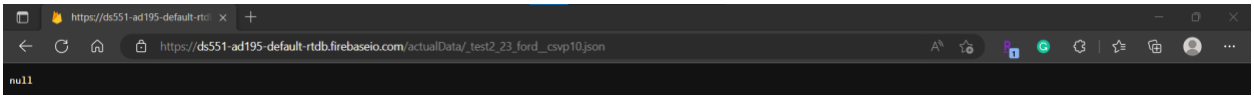
Partition 1:



Partition 4:



Partition 10: (out of range)



Example outcomes

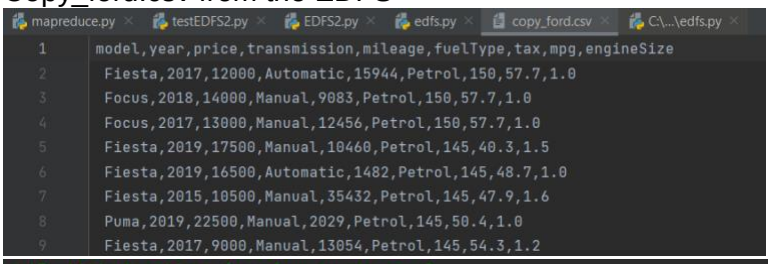
1. Task 1 (Yucheng's work)

```
C:\Users\genac\PycharmProjects\Project1\venv\Scripts\python.exe C:/Us
welcome EDFS~
mkdir /user ch
folder exists
False
rm /user/ch
{'success': False, 'data': 'args not found'}
rm /user ch

rm success
True
```

```
ls /user/yyy
{'success': True, 'data': ['toyota~csv']}
cat /user/yyy toyota8
file doesn't exsit
False
put /user/ch ford.csv
{'success': False, 'data': 'args not found'}
put /user/ch ford.csv 4
True
get /user/ch ford.csv
save done
```

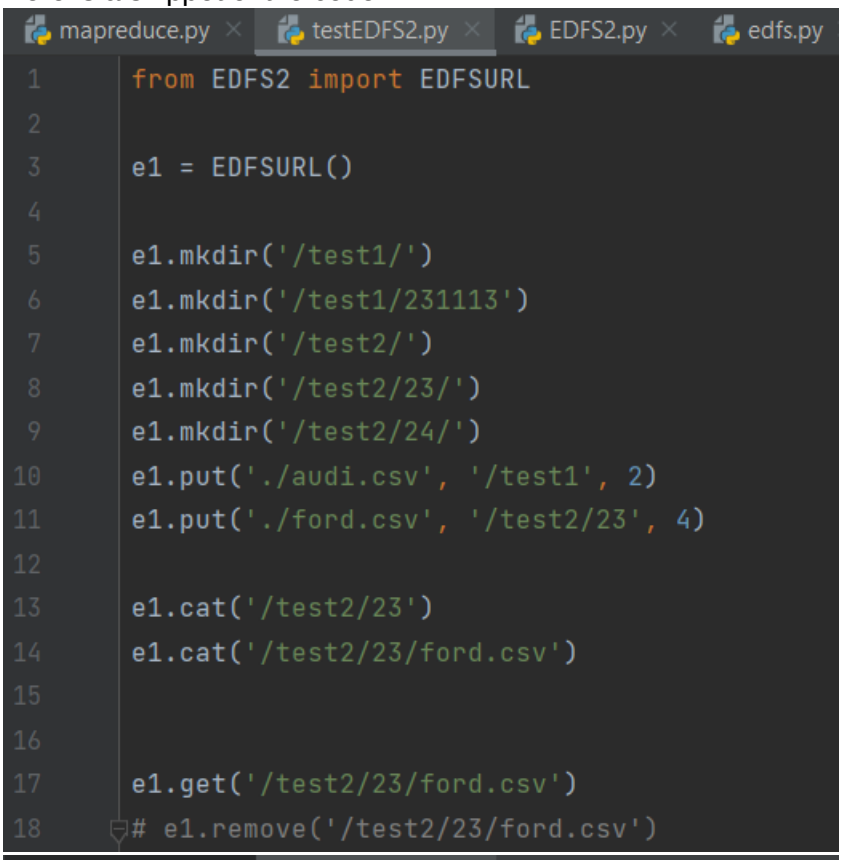
Copy_ford.csv from the EDFS




```
getPartitionLocations /user/yyy toyota.csv 2
https://demo01-76e03-default-rtdb.firebaseio.com/data/-user-yyy-toyota2.json

readPartition /user.yyy toyota
{'success': False, 'data': 'command not found'}
path is false
False
readPartition /user/yyy toyota2.csv
file doesn't exsit
False
```

2. Task 2 (Junhui's work)
   I ran the script with variations.
   Here is a snippet of the code:



(a). run with all data already available and remove them

```
C:\Users\genac\PycharmProjects\Project1\venv\Scripts\python.exe
>>>EDFS Is Running
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Write ERROR: File /test1/audi.csv already exists
Write ERROR: File /test2/23/ford.csv already exists
Cat ERROR: Wrong File Path
```

	engineSize	fuelType	mileage	model	mpg	price	tax	transmission	year
0	1.0	Petrol	15944	Fiesta	57.7	12000	150	Automatic	2017
1	1.0	Petrol	9083	Focus	57.7	14000	150	Manual	2018
2	1.0	Petrol	12456	Focus	57.7	13000	150	Manual	2017
3	1.5	Petrol	10460	Fiesta	40.3	17500	145	Manual	2019
4	1.0	Petrol	1482	Fiesta	48.7	16500	145	Automatic	2019

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 9 columns):
```

```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   engineSize    18 non-null     float64
 1   fuelType      18 non-null     object
 2   mileage       18 non-null     int64
 3   model         18 non-null     object
 4   mpg           18 non-null     float64
 5   price         18 non-null     int64
 6   tax           18 non-null     int64
 7   transmission  18 non-null     object
```

```
 8   year          18 non-null     int64
dtypes: float64(2), int64(4), object(3)
memory usage: 1.4+ KB
None
Get File: file stored in ./downloaded-test2-23-ford.csv
Remove: success
Remove: success
/test1
/test2
Ls ERROR: Wrong path /test2/23
Get Locations ERROR: Wrong path /test2/23/ford.csv
Read Partition ERROR: Wrong path /test2/23/ford.csv
Read Partition ERROR: Wrong path /test2/23/ford.csv
Read Partition ERROR: Must read a file but not a directory
Remove ERROR: Must remove a file but not a directory

Process finished with exit code 0
```

(b). run with datasets removed:

```
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Mkdir ERROR: Directory Already Exists
Mkdir: Success
Mkdir ERROR: Directory Already Exists
{"model": " A1", "year": 2017, "price": 12500, "transmission": "
{"model": " A6", "year": 2016, "price": 16500, "transmission": "
{"model": " A1", "year": 2016, "price": 11000, "transmission": "
{"model": " A4", "year": 2017, "price": 16800, "transmission": "
{"model": " A3", "year": 2015, "price": 10200, "transmission": "Manual
{"model": " A1", "year": 2016, "price": 12000, "transmission": "Manual
{"model": " A3", "year": 2017, "price": 16100, "transmission": "Manual
Write: success
{"model": " Fiesta", "year": 2017, "price": 12000, "transmission": "Au
{"model": " Focus", "year": 2018, "price": 14000, "transmission": "Man
{"model": " Focus", "year": 2017, "price": 13000, "transmission": "Man
{"model": " Fiesta", "year": 2019, "price": 17500, "transmission": "Ma
{"model": " Fiesta", "year": 2019, "price": 16500, "transmission": "Au
{"model": " Fiesta", "year": 2015, "price": 10500, "transmission": "Ma
{"model": " Puma", "year": 2019, "price": 22500, "transmission": "Manu
```

```
{"model": " Kuga", "year": 2018, "price": 16899, "transmission": "
Write: success
Cat ERROR: Wrong File Path
   engineSize fuelType  mileage    model   mpg  price  tax transmi
0         1.0   Petrol    15944   Fiesta  57.7  12000  150    Auto
1         1.0   Petrol     9083    Focus  57.7  14000  150       M
2         1.0   Petrol    12456    Focus  57.7  13000  150       M
3         1.5   Petrol    10460   Fiesta  40.3  17500  145       M
4         1.0   Petrol     1482   Fiesta  48.7  16500  145    Auto
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
```

```
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   engineSize    18 non-null     float64
 1   fuelType      18 non-null     object
 2   mileage       18 non-null     int64
 3   model         18 non-null     object
 4   mpg           18 non-null     float64
 5   price         18 non-null     int64
 6   tax           18 non-null     int64
 7   transmission  18 non-null     object
```

```
Get Locations: The 4 part of the file is stored in datanode _test2_23_ford__csvp4
Read Partition: The 2 part of the file is:
   engineSize fuelType  mileage    model   mpg  price  tax transmission  year
0         1.0   Petrol     1482   Fiesta  48.7  16500  145    Automatic  2019
1         1.6   Petrol    35432   Fiesta  47.9  10500  145       Manual  2015
2         1.0   Petrol     2029     Puma  50.4  22500  145       Manual  2019
3         1.2   Petrol    13054   Fiesta  54.3   9000  145       Manual  2017
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
```

```
dtypes: float64(2), int64(4), object(3)
memory usage: 1.4+ KB
None
Get File: file stored in ./downloaded-test2-23-ford.csv
/test1
/test2
/test2/23/ford.csv
Get Locations: The 1 part of the file is stored in datanode _test2_23_ford__csvp1
Get Locations: The 2 part of the file is stored in datanode _test2_23_ford__csvp2
Get Locations: The 3 part of the file is stored in datanode _test2_23_ford__csvp3
Get Locations: The 4 part of the file is stored in datanode _test2_23_ford__csvp4
```

#	Column	Non-Null Count	Dtype
0	engineSize	4 non-null	float64
1	fuelType	4 non-null	object
2	mileage	4 non-null	int64
3	model	4 non-null	object
4	mpg	4 non-null	float64
5	price	4 non-null	int64
6	tax	4 non-null	int64
7	transmission	4 non-null	object
8	year	4 non-null	int64

```
 7   transmission  4 non-null      object
 8   year          4 non-null      int64
dtypes: float64(2), int64(4), object(3)
memory usage: 416.0+ bytes
None
Read Partition ERROR: Wrong partition number 10
Read Partition ERROR: Must read a file but not a directory
Remove ERROR: Must remove a file but not a directory

Process finished with exit code 0
```

3. Task 2: MapReduce (Carra's work)

year	price
1998	19990.00
1999	1995.00
2000	2695.00
2002	1698.83
2003	2286.88
2004	3028.33
2005	2582.50
2006	3077.19
2007	2920.73
2008	4155.88
2009	4265.67
2010	5020.28
2011	5158.11
2012	6428.46
2013	8644.56
2014	8942.82
2015	9979.49
2016	11639.35
2017	12251.42
2018	12457.62
2019	16586.60
2020	22509.19

Changes in proposed items

We originally want to commit one Firebase implementation and one MySQL implementation. However, wo considered that using MySQL to build a table of folder parent-child relationships and get the file directories or paths from the table is more cumbersome than using Firebase. Thus, we decided to do two Firebase implementation, but in 2 different ways, namely Firebase SDK and Restful requests.

We also implemented analytics function using pure Python, instead of Python and Java, because it is more convenient to do.

Challenges

Connecting to and implementing CRUD on Firebase through firebase's Python SDK, the firebase toolkit, is somewhat difficult. But the Firebase development documentation is very helpful.

Since MySQL in the AWS EC2 environment can only be connected via SSH, it took us a while to find a correct way using Python's toolkit, sshTunnel and pymysql, and generated keys to connect and modify the MySQL database.

Tracker status

We are currently on track for our milestones as that we have functioning code for task 1 and 2, yet we need a little bit of code improvement to make it more readable and easier to use.

We may also change the analytics function if we feel like that another function is more insightful.

Conclusion

It is a fun project to implement, It deepens our understanding of distributed file systems, Firebase, Python-firebase connections, and MapReduce model. We wrote two implementations of EDFSs and one MapReduce function using Firebase with Python scripts, interacted with Firebase via terminal with the help of Python scripts, and wrote a MapReduce function which is able to calculate the mean of car price for each year. Later in task3, we will design and implement more complex search and analytics modules.