Vehicle Sales Profit Analysis

Big Data Applications – Mini Project

Arnaav Anand

12/06/2024

1. Introduction

1.1. Project Goal

The goal of this project is to build a data processing pipeline that leverages Amazon Web Services (AWS) infrastructure to manage and analyze data effectively. This pipeline aims to process the car_prices dataset, extract meaningful insights, and deliver notifications about critical trends or updates in the data pipeline. Additionally, the pipeline is integrated with AWS QuickSight to provide periodic visual reporting, ensuring stakeholders can access up-to-date information through dynamic dashboards. The data is also leveraged in Amazon's Sagemaker Studio to train multiple models and find the best one to predict the profit based on

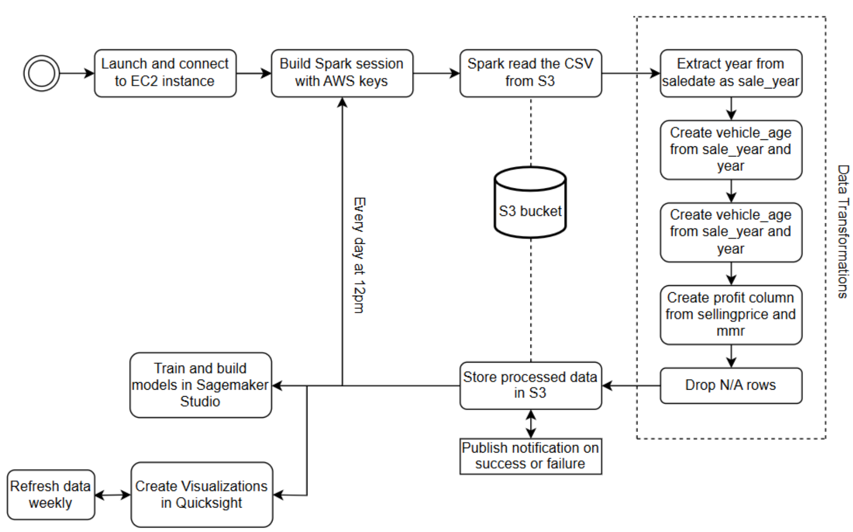


Fig. 1: Project Workflow

contributing features. The automation removes manual intervention, ensuring seamless and timely updates while optimizing computational resources and costs. The summarized workflow can be seen in Fig. 1.

1.2. Dataset

The car_prices dataset (Fig. 2) serves as a comprehensive record of historical car sales, offering a wealth of information to support detailed analyses of vehicle pricing, trends, and market dynamics. The dataset, downloaded from Kaggle, contains 558,838 rows and 16 columns. Each entry in the dataset represents a single car sale, accompanied by key attributes that define the vehicle's characteristics, condition, and sales specifics.

The dataset has general vehicle details, such as the year of manufacture, make, model, and trim level, which define the car’s identity and specifications. These attributes are critical for distinguishing among different versions of the same vehicle and understanding how variations like premium trims or specific model years impact value. The body type and transmission type provide additional insights into the car’s design and market segment.

To ensure vehicle uniqueness, the dataset includes a VIN (Vehicle Identification Number), a universally recognized identifier, which can be useful for tracking individual cars. Geographic context is also captured through the state field, indicating where the sale occurred.

The dataset evaluates the vehicle’s physical and mechanical state through the condition field, represented on a numerical scale, and the odometer reading, which quantifies the mileage. Lower mileage and better conditions often correlate with higher selling prices, making these critical factors for pricing analysis. The exterior and interior color details provide additional dimensions for market preference studies.

Seller-related information, including the seller's name or organization, offers insights into the types of entities involved in the sale, such as dealerships, leasing companies, or rental agencies.

Financial data within the dataset includes the Manheim Market Report (MMR) value, which represents an estimated market price, and the actual selling price, allowing for a comparison between expected and actual outcomes. Each transaction is time-stamped with a precise sale date and time.

Overall, the dataset’s rich set of features provides a robust foundation for exploring questions around vehicle valuation, buyer preferences, market trends, and the economics of car sales.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	year	make	model	trim	body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice	saledate
2	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5	16639	white	black	kia motors america inc	20500	21500	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
3	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5	9393	white	beige	kia motors america inc	20800	21500	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
4	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	45	1331	gray	black	financial services remarketing (lease)	31900	30000	Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
5	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	41	14282	white	black	volvo na rep/world omni	27500	27750	Thu Jan 29 2015 04:30:00 GMT-0800 (PST)
6	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	43	2641	gray	black	financial services remarketing (lease)	66000	67000	Thu Dec 18 2014 12:30:00 GMT-0800 (PST)
7	2015	Nissan	Altima	2.5 S	Sedan	automatic	1n4al3ap1fn326013	ca	1	5554	gray	black	enterprise vehicle exchange / tra / rental / tulsa	15350	10900	Tue Dec 30 2014 12:00:00 GMT-0800 (PST)
8	2014	BMW	M5	Base	Sedan	automatic	wbsfv9c51ed593089	ca	34	14943	black	black	the hertz corporation	69000	65000	Wed Dec 17 2014 12:30:00 GMT-0800 (PST)
9	2014	Chevrolet	Cruze	1LT	Sedan	automatic	1g1pc5sb2e7128460	ca	2	28617	black	black	enterprise vehicle exchange / tra / rental / tulsa	11900	9800	Tue Dec 16 2014 13:00:00 GMT-0800 (PST)
10	2014	Audi	A4	2.0T Premium Plus quattro	Sedan	automatic	wauffafl3en030343	ca	42	9557	white	black	audi mission viejo	32100	32250	Thu Dec 18 2014 12:00:00 GMT-0800 (PST)
11	2014	Chevrolet	Camaro	LT	Convertible	automatic	2g1fb3d37e9218789	ca	3	4809	red	black	d/m auto sales inc	26300	17500	Tue Jan 20 2015 04:00:00 GMT-0800 (PST)
12	2014	Audi	A6	3.0T Prestige quattro	Sedan	automatic	wauhgafc0en062916	ca	48	14414	black	black	desert auto trade	47300	49750	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
13	2015	Kia	Optima	LX	Sedan	automatic	5xxgm4a73fg353538	ca	48	2034	red	tan	kia motors finance	15150	17700	Tue Dec 16 2014 12:00:00 GMT-0800 (PST)
14	2015	Ford	Fusion	SE	Sedan	automatic	3fa6p0hdxfr145753	ca	2	5559	white	beige	enterprise vehicle exchange / tra / rental / tulsa	15350	12000	Tue Jan 13 2015 12:00:00 GMT-0800 (PST)
15	2015	Kia	Sorento	LX	SUV	automatic	5xyktca66fg561407	ca	5	14634	silver	black	kia motors america inc	20600	21500	Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
16	2014	Chevrolet	Cruze	2LT	Sedan	automatic	1g1pe5sbxe7120097	ca		15686	blue	black	avis rac/san leandro	13900	10600	Tue Dec 16 2014 12:00:00 GMT-0800 (PST)
17	2015	Nissan	Altima	2.5 S	Sedan	automatic	1n4al3ap5fc124223	ca	2	11398	black	black	enterprise vehicle exchange / tra / rental / tulsa	14750	14100	Tue Dec 23 2014 12:00:00 GMT-0800 (PST)
18	2015	Hyundai	Sonata	SE	Sedan	automatic	5npe24af4fh001562	ca		8311	red	—	avis tra	15200	4200	Tue Dec 16 2014 13:00:00 GMT-0800 (PST)
19	2014	Audi	Q5	2.0T Premium Plus quattro	SUV	automatic	wa1lfafpxea085074	ca	49	7983	white	black	audi north scottsdale	37100	40000	Thu Dec 18 2014 12:30:00 GMT-0800 (PST)
20	2014	Chevrolet	Camaro	LS	Coupe	automatic	2g1fa1e39e9134494	ca	17	13441	black	black	wells fargo dealer services	17750	17000	Tue Dec 30 2014 15:00:00 GMT-0800 (PST)

Fig 2: Top 20 rows of the data set

2. Methodology

2.1. AWS Setup

The first step in the project involved setting up the AWS environment, which served as the backbone for the entire data pipeline. An EC2 instance was provisioned to act as the primary computational resource, with the Amazon Linux 2023 operating system selected for its compatibility and performance. A security group was configured to allow SSH access and other necessary protocols, ensuring secure communication with the instance. Additionally, IAM roles were assigned to the EC2 instance to grant permissions for accessing S3 buckets and other AWS services without embedding credentials, enhancing security.

A Zero budget billing alert was created to ensure that no additional costs were incurred or would be caught immediately. IAM Roles were configured to handle various AWS features such as Administrator Access, EC2 for SSM, and SNS Full Access.

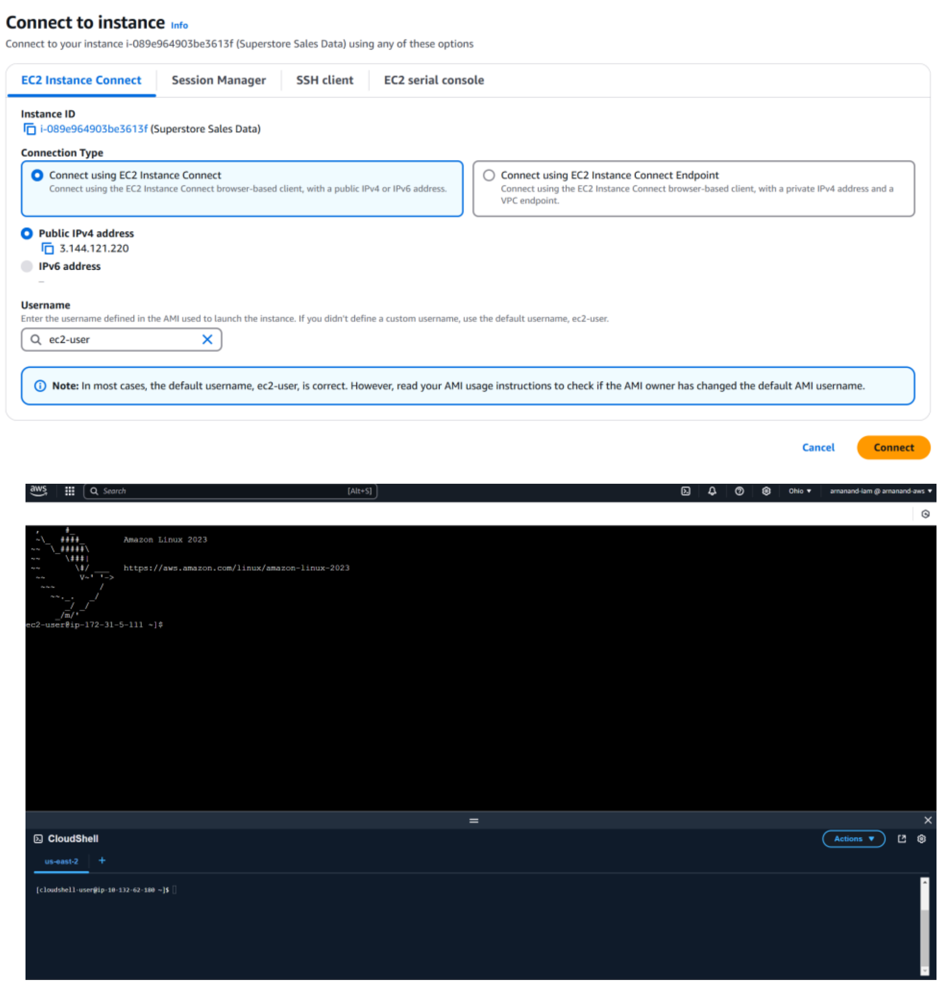


Fig. 3: Connecting to the initialized EC2 instance

After initializing the EC2 instance (Fig. 3), it was linked with the AWS Management Console to provide a seamless interface for monitoring and managing resources. With this setup, the AWS environment was ready to host the various components of the project pipeline.

2.2. S3 Bucket Creation

The next step was to create an S3 bucket, which served as the central storage repository for the dataset and intermediate outputs. Using the AWS Management Console, a bucket was created in the us-east-2 region with a unique name to ensure global accessibility. Proper bucket policies and permissions were implemented to restrict access to authorized users and the EC2 instance alone.

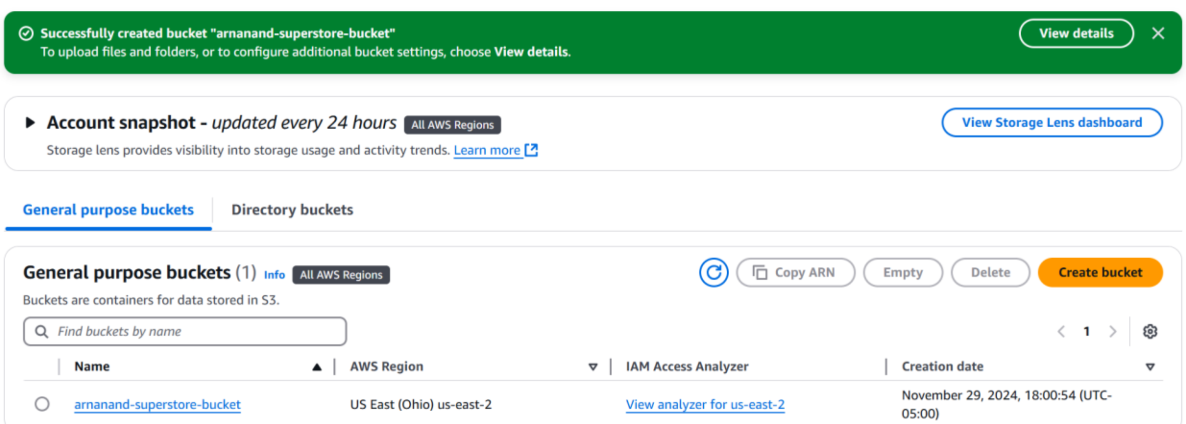


Fig. 4: Raw data stored in the Bucket s3://arnanand-superstore-bucket

The raw CSV data was successfully uploaded into the bucket (Fig. 4) as car_prices.csv.

2.3. CLI Configurations & Installations

The AWS Command Line Interface (CLI) was installed and configured on the EC2 instance to streamline interactions with AWS services. The configurations were based on the access key ID and secret access password key along with the region (us-east-2) and format (JSON). The installation phase primarily involved installing packages using yum. The instance was then authenticated using IAM roles, eliminating the need for access keys.

In addition to the AWS CLI, other essential tools such as Python, Spark, and dependencies for data processing were installed. Environment variables were configured in a new pyspark environment `python3 -m venv ~/pyspark-env` and the Jupyter notebook was also set up with a custom password for easy visualization of the preprocessing. The following are the command line codes executed for the installations.

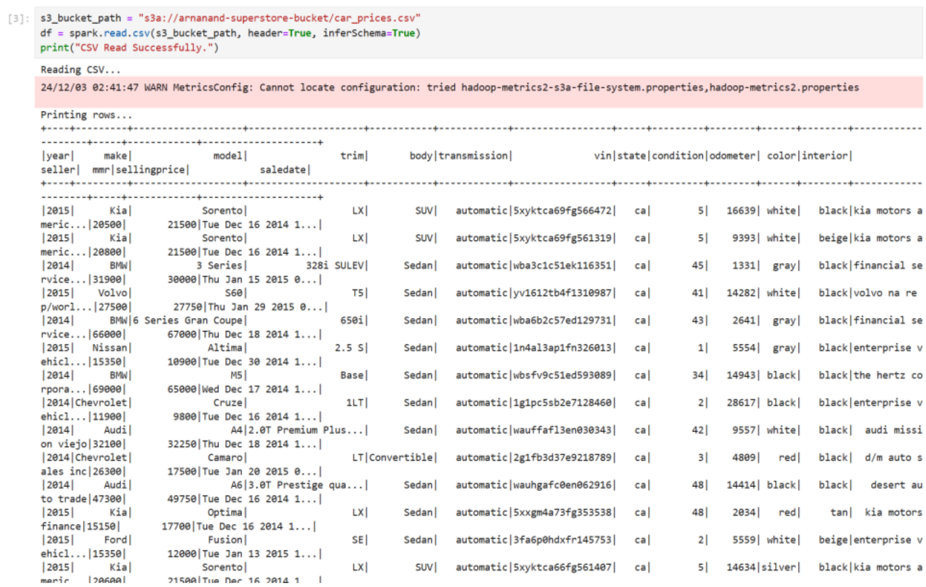
```
Java

wget https://github.com/adoptium/temurin11-binaries/releases/download/jdk-
11.0.19+7/OpenJDK11U-jdk_x64_linux_hotspot_11.0.19_7.tar.gz

tar -xzf OpenJDK11U-jdk_x64_linux_hotspot_11.0.19_7.tar.gz

sudo mv jdk-11.0.19+7 /usr/local/java

echo "export PATH=$PATH:/usr/local/java/bin" >> ~/.bashrc

echo "export JAVA_HOME=/usr/local/java/jdk-11.0.19+7" >> ~/.bashrc

echo "export PATH=\$JAVA_HOME/bin:\$PATH" >> ~/.bashrc

source ~/.bashrc


Pyspark

wget https://archive.apache.org/dist/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

tar -xvzf spark-3.4.1-bin-hadoop3.tgz

sudo mv spark-3.4.1-bin-hadoop3 /usr/local/spark

echo "export SPARK_HOME=/usr/local/spark" >> ~/.bashrc

echo "export PATH=$PATH:$SPARK_HOME/bin" >> ~/.bashrc
```

```
source ~/.bashrc


Hadoop
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.1/hadoop-3.4.1.tar.gz
 tar -xzvf hadoop-3.4.1.tar.gz
sudo mv hadoop-3.4.1 /usr/local/Hadoop
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
source ~/.bashrc


Jupyter
sudo yum install python3 -y  # For Amazon Linux
pip3 install notebook
jupyter notebook password
```

2.4. Data Ingestion

The Pyspark session was initialized with all the access tokens and necessary configurations. The data was then pulled from S3 within a jupyter notebook environment. To verify successful ingestion, the bucket's contents were listed, confirming the presence of the uploaded dataset (Fig. 5).

The dataset was now accessible for processing by other AWS services and tools configured within the pipeline.



```
[3]: s3_bucket_path = "s3a://arnanand-superstore-bucket/car_prices.csv"
df = spark.read.csv(s3_bucket_path, header=True, inferSchema=True)
print("CSV Read Successfully.")

Reading CSV...
24/12/03 02:41:47 WARN MetricsConfig: Cannot locate configuration: tried hadoop-metrics2-s3a-file-system.properties,hadoop-metrics2.properties

Printing rows...
+----+---------+------------------+--------------------+----------+-----------+-----------------+-----+---------+--------+------+--------+-----------
--------+-----+-----------+--------------------+
|year|     make|             model|                trim|      body|transmission|              vin|state|condition|odometer| color|interior|
seller|  mmr|sellingprice|            saledate|
+----+---------+------------------+--------------------+----------+-----------+-----------------+-----+---------+--------+------+--------+-----------
--------+-----+-----------+--------------------+
|2015|      Kia|           Sorento|                  LX|       SUV|   automatic|5xyktca69fg566472|   ca|        5|   16639| white|   black|kia motors a
meric...|20500|       21500|Tue Dec 16 2014 1...|
|2015|      Kia|           Sorento|                  LX|       SUV|   automatic|5xyktca69fg561319|   ca|        5|    9393| white|   beige|kia motors a
meric...|20800|       21500|Tue Dec 16 2014 1...|
|2014|      BMW|          3 Series|          328i SULEV|     Sedan|   automatic|wba3c1c51ek116351|   ca|       45|    1331|  gray|   black|financial se
rvice...|31900|       30000|Thu Jan 15 2015 0...|
|2015|    Volvo|               S60|                  T5|     Sedan|   automatic|yv1612tb4f1310987|   ca|       41|   14282| white|   black|volvo na re
p/worl...|27500|       27750|Thu Jan 29 2015 0...|
|2014|      BMW|6 Series Gran Coupe|                650i|     Sedan|   automatic|wba6b2c57ed129731|   ca|       43|    2641|  gray|   black|financial se
rvice...|66000|       67000|Thu Dec 18 2014 1...|
|2015|   Nissan|            Altima|               2.5 S|     Sedan|   automatic|1n4al3ap1fn326013|   ca|        1|    5554|  gray|   black|enterprise v
ehicl...|15350|       10900|Tue Dec 30 2014 1...|
|2014|      BMW|                M5|                Base|     Sedan|   automatic|wbsfv9c51ed593089|   ca|       34|   14943| black|   black|the hertz co
rpora...|69000|       65000|Wed Dec 17 2014 1...|
|2014|Chevrolet|             Cruze|                 1LT|     Sedan|   automatic|1g1pc5sb2e7128460|   ca|        2|   28617| black|   black|enterprise v
ehicl...|11900|        9800|Tue Dec 16 2014 1...|
|2014|     Audi|                A4|2.0T Premium Plus...|     Sedan|   automatic|wauffafl3en030343|   ca|       42|    9557| white|   black|  audi missi
on viejo|32100|       32250|Thu Dec 18 2014 1...|
|2014|Chevrolet|            Camaro|                  LT|Convertible|   automatic|2g1fb3d37e9218789|   ca|        3|    4809|   red|   black|  d/m auto s
ales inc|26300|       17500|Tue Jan 20 2015 0...|
|2014|     Audi|                A6|3.0T Prestige qua...|     Sedan|   automatic|wauhgafc0en062916|   ca|       48|   14414| black|   black|   desert au
to trade|47300|       49750|Tue Dec 16 2014 1...|
|2015|      Kia|            Optima|                  LX|     Sedan|   automatic|5xxgm4a73fg353538|   ca|       48|    2034|   red|     tan|  kia motors
finance|15150|       17700|Tue Dec 16 2014 1...|
|2015|     Ford|            Fusion|                  SE|     Sedan|   automatic|3fa6p0hdxfr145753|   ca|        2|    5559| white|   beige|enterprise v
ehicl...|15350|       12000|Tue Jan 13 2015 1...|
|2015|      Kia|           Sorento|                  LX|       SUV|   automatic|5xyktca66fg561407|   ca|        5|   14634|silver|   black|kia motors a
meric...|20600|       21500|Tue Dec 16 2014 1...|
```

Fig. 5: Displaying initial data pulled from S3

2.5. Data Processing

During the data processing stage, Spark was used to clean and transform the dataset efficiently. One key transformation involved calculating the vehicle's age and profit margin, which required extracting and manipulating specific data points from the dataset.

The saledate column was processed using a regular expression to extract the year component, which was then cast into an integer. This extracted year (sale_year) was used to compute the vehicle's age by subtracting the year column. The derived vehicle_age column provided an essential feature for further analysis. To ensure data consistency, any cases where vehicle_age was negative (due to cases with models having nomenclature of the subsequent year) were converted to zero. Additionally, rows with null values in the vehicle_age column were removed, ensuring clean and reliable data for analysis.

Another derived feature was the profit margin, calculated as the difference between the sellingprice and the mmr. This profit column offered valuable insight into pricing patterns and vehicle valuation.

Finally, the processed DataFrame underwent a final cleaning step to remove any remaining null values across all columns, ensuring the dataset was fully prepared for downstream analysis and machine learning tasks (Fig. 6).

```
[4]: print(f"Count of vehicle_age below 0: {df.filter(col('vehicle_age') < 0).count()}")
     print(f"Count of vehicle_age null: {df.filter(col('vehicle_age').isNull()).count()}")

     Count of vehicle_age below 0: 201
     [Stage 7:>                                                          (0 + 1) / 1]
     Count of vehicle_age null: 33
```

```
[5]: from pyspark.sql.functions import when

     # convert negatives to 0 and remove nulls
     df = df.withColumn("vehicle_age", when(col("vehicle_age") < 0, 0).otherwise(col("vehicle_age"))) \
            .na.drop(subset=["vehicle_age"])
     df.show()
```

```
+----+---------+--------------------+--------------------+-----------+------------+-----------------+-----+---------+--------+------+--------+-----------
--------+-----+-----------+--------------------+-----------+------+
|year|     make|               model|                trim|       body|transmission|              vin|state|condition|odometer| color|interior|
seller|  mmr|sellingprice|            saledate|vehicle_age|profit|
+----+---------+--------------------+--------------------+-----------+------------+-----------------+-----+---------+--------+------+--------+-----------
--------+-----+-----------+--------------------+-----------+------+
|2015|      Kia|             Sorento|                  LX|        SUV|   automatic|5xyktca69fg566472|   ca|        5|   16639| white|   black|kia motors a
meric...|20500|       21500|Tue Dec 16 2014 1...|          0|  1000|
|2015|      Kia|             Sorento|                  LX|        SUV|   automatic|5xyktca69fg561319|   ca|        5|    9393| white|   beige|kia motors a
meric...|20800|       21500|Tue Dec 16 2014 1...|          0|   700|
|2014|      BMW|            3 Series|          328i SULEV|      Sedan|   automatic|wba3c1c51ek116351|   ca|       45|    1331|  gray|   black|financial se
rvice...|31900|       30000|Thu Jan 15 2015 0...|          1| -1900|
|2015|    Volvo|                 S60|                  T5|      Sedan|   automatic|yv1612tb4f1310987|   ca|       41|   14282| white|   black|volvo na re
p/worl...|27500|       27750|Thu Jan 29 2015 0...|          0|   250|
|2014|      BMW|6 Series Gran Coupe|                650i|      Sedan|   automatic|wba6b2c57ed129731|   ca|       43|    2641|  gray|   black|financial se
rvice...|66000|       67000|Thu Dec 18 2014 1...|          0|  1000|
|2015|   Nissan|              Altima|               2.5 S|      Sedan|   automatic|1n4al3ap1fn326013|   ca|        1|    5554|  gray|   black|enterprise v
ehicl...|15350|       10900|Tue Dec 30 2014 1...|          0| -4450|
|2014|      BMW|                  M5|                Base|      Sedan|   automatic|wbsfv9c51ed593089|   ca|       34|   14943| black|   black|the hertz co
rpora...|69000|       65000|Wed Dec 17 2014 1...|          0| -4000|
|2014|Chevrolet|               Cruze|                 1LT|      Sedan|   automatic|1g1pc5sb2e7128460|   ca|        2|   28617| black|   black|enterprise v
ehicl...|11900|        9800|Tue Dec 16 2014 1...|          0| -2100|
|2014|     Audi|                  A4|2.0T Premium Plus...|      Sedan|   automatic|wauffafl3en030343|   ca|       42|    9557| white|   black|  audi missi
on viejo|32100|       32250|Thu Dec 18 2014 1...|          0|   150|
|2014|Chevrolet|              Camaro|                  LT|Convertible|   automatic|2g1fb3d37e9218789|   ca|        3|    4809|   red|   black|  d/m auto s
ales inc|26300|       17500|Tue Jan 20 2015 0...|          1| -8800|
|2014|     Audi|                  A6|3.0T Prestige qua...|      Sedan|   automatic|wauhgafc0en062916|   ca|       48|   14414| black|   black|   desert au
to trade|47300|       49750|Tue Dec 16 2014 1...|          0|  2450|
|2015|      Kia|              Optima|                  LX|      Sedan|   automatic|5xxgm4a73fg353538|   ca|       48|    2034|   red|     tan|  kia motors
finance|15150|       17700|Tue Dec 16 2014 1...|          0|  2550|
|2015|     Ford|              Fusion|                  SE|      Sedan|   automatic|3fa6p0hdxfr145753|   ca|        2|    5559| white|   beige|enterprise v
ehicl...|15350|       12000|Tue Jan 13 2015 1...|          0| -3350|
```

Fig. 6: The final processed dataset

2.5. Data Aggregation

5 Aggregation queries were run in a separate .py file with the processed dataset. Jupyter notebook was not capable of running these operations as the computational load often caused it to stall and hang, so these were created in the file system and executed via spark-submit (Fig. 7).

4) Average Vehicle Age by Body Type

body	avg_vehicle_age
G Convertible	3.38
Koup	3.21
Quad Cab	6.43
van	3.94
crew cab	4.65
G Sedan	2.79
Access Cab	6.22
Extended Cab	8.26
Transit Van	0.0
crewmax cab	3.48
Hatchback	3.88
cts wagon	5.0
supercab	6.04
Club Cab	10.05
Ram Van	15.0
G Coupe	2.85
Q60 Coupe	0.81
g coupe	2.68
Convertible	6.28
minivan	4.29

only showing top 20 rows

5) Maximum Profit for Each Seller

seller	max_profit
balboa thrift & l...	2025
california auto w...	9900
repo remarketing/...	750
low gos used cars	1700
jaguar land rover...	3300
montclair auto sl...	1400
pa distributors	6000
bailey auto plaza	1250
autolenders liqui...	2125
southern auto fin...	4200
premier toyota of...	1275
rock chevrolet	900
autonation honda ...	3025
hyundai of everet...	350
select remarketin...	1425
onemain rem/ulric...	275
grossinger toyota...	750
central florida p...	2200
frank kent honda	2850
hincklease	2400

only showing top 20 rows

1) Top 10 makes based on the average selling price

make	avg_selling_price
Rolls-Royce	153456.25
Ferrari	128852.94
Lamborghini	111500.0
Bentley	72713.33
Tesla	67054.35
Aston Martin	55500.0
Fisker	46461.11
Maserati	43729.82
Lotus	40800.0
Porsche	38932.11

2) Top 10 total profit by vehicle

make	total_profit
HUMMER	214263
Aston Martin	34200
Suzuki	23845
Acura	950
Lotus	500
Daewoo	-75
Geo	-350
Isuzu	-6075
Lamborghini	-6500
Tesla	-17450

3) Number of Vehicles Sold by Condition

condition	vehicles_sold
31	7942
34	15096
28	16650
26	10370
27	14173
44	22091
12	86
22	5235
47	9743
1	5805
13	74
16	151
3	9190
48	10884
5	9414
19	36647
41	19889
43	21593
15	116
37	22680

only showing top 20 rows

Fig. 7: Outputs of the Data aggregation queries

2.6. Storing Processed Data to S3

The output is sent back to S3 using the Spark write function. However, it always defaulted to creating a folder having the same name of the output file name, with a part file containing the processed CSV inside. This had to be manually moved back to the root of the storage (Fig. 8).

```
Data successfully written to s3a://arnanand-superstore-bucket/car_data_processed_testing.csv
[ec2-user@ip-172-31-11-64 ~]$
```

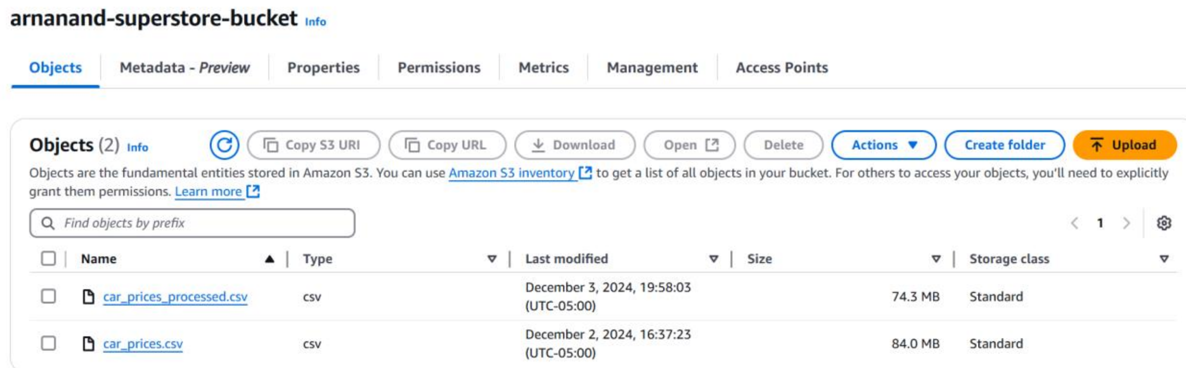


Fig. 8: The processed CSV data in the S3 bucket

2.7. Data Analysis Using SQL

5 Queries were run using Spark SQL to provide useful insights into the data distribution (Fig. 9)

```
1. Total number of cars sold per make

+---------+---------------+
|     make|total_cars_sold|
+---------+---------------+
|     Ford|          81013|
|Chevrolet|          54150|
|   Nissan|          44043|
|   Toyota|          35313|
|    Dodge|          27181|
|    Honda|          24781|
|  Hyundai|          18659|
|      BMW|          17509|
|      Kia|          15828|
| Chrysler|          15133|
+---------+---------------+
```

```
3. Top 5 makes with the highest average profit

+------------+----------+
|        make|avg_profit|
+------------+----------+
|Aston Martin|    1425.0|
|       Lotus|     500.0|
|      HUMMER|    278.99|
|      Suzuki|     24.41|
|       Acura|      0.21|
+------------+----------+
```

```
2. Average selling price by car condition

+---------+-----------------+
|condition|avg_selling_price|
+---------+-----------------+
|        5|         24582.62|
|       49|         22686.27|
|       48|         21377.92|
|       47|         20881.72|
|       46|         20366.42|
|       45|         19892.98|
|       44|          19118.5|
|       43|         18408.09|
|       42|         17791.51|
|        4|          17621.0|
|       41|         17138.33|
|       39|         15893.33|
|       38|         15357.55|
|       37|         14786.36|
|       36|         13964.99|
|       35|         13290.61|
|       34|         12657.39|
|        3|         12514.41|
|       33|         12144.13|
|       32|         11784.98|
+---------+-----------------+
only showing top 20 rows
```

```
4. Number of vehicles sold by body type

+------------+-------------+
|        body|vehicles_sold|
+------------+-------------+
|       Sedan|       174647|
|         SUV|       100347|
|       sedan|        36651|
|         suv|        20621|
|   Hatchback|        19351|
|     Minivan|        18305|
|       Coupe|        13121|
|       Wagon|        12023|
|    Crew Cab|        11508|
| Convertible|         7725|
|   SuperCrew|         6195|
|     G Sedan|         5644|
|   hatchback|         4470|
|     minivan|         3633|
|    SuperCab|         3471|
| Regular Cab|         3395|
|Extended Cab|         3329|
|         Van|         3069|
|    Quad Cab|         2937|
|       coupe|         2839|
+------------+-------------+
only showing top 20 rows
```

```
5. Maximum selling price for each year

+----+-----------------+
|year|max_selling_price|
+----+-----------------+
|1990|            11500|
|1991|             6750|
|1992|            13700|
|1993|             4350|
|1994|             9800|
|1995|            13600|
|1996|            12500|
|1997|            10900|
|1998|            17100|
|1999|            18100|
|2000|            25400|
|2001|            45750|
|2002|            38100|
|2003|            29600|
|2004|            81000|
|2005|            54500|
|2006|            66500|
|2007|           117500|
|2008|           117000|
|2009|            96000|
+----+-----------------+
only showing top 20 rows
```

Fig. 9: Outputs of the Spark SQL queries

2.8. Machine Learning with AWS Sagemaker

The project leveraged AWS SageMaker to develop and train machine learning models. The processed data was split into training and test sets, which were then uploaded to SageMaker's environment. A regression model was trained to predict car prices using attributes. Overall, the model training and building process took about 4 hours.

2.9. Visualization

To communicate findings effectively, Amazon QuickSight was used for creating an interactive dashboard. The processed data in the S3 bucket was connected to QuickSight, and visualizations such as bar charts, scatter plots, and time-series analyses were built to illustrate key trends like yearly average profits.

Schedules

Refresh type	Occurrence	Start time	Timezone	Actions
Full refresh	Weekly (Mon)	15:53	America/Indianapolis	⋮

History

Show times within Last 90 days with status of All

Refresh start	Status	Duration	Skipped rows	Ingested ro...	Dataset rows	Refresh type
December 6, 2024 at 12:00 AM EST	Completed	41 seconds	0	472325	472325	Manual, Initial

1-1 of 1

Fig. 10: Periodic refresh of the data in Quicksight

The Dashboard was configured to refresh periodically (Fig. 10.), ensuring they displayed the latest insights from the dataset.

2.10. Automation of the pipeline

Automation was achieved by integrating all pipeline stages into a cron-based script on the EC2 instance (Fig. 11). The script orchestrated tasks such as fetching new data, processing it with Spark, and storing the processed output back to S3.

The cron job (0 12 * * * python3 automation.py) ensured that the script ran at 12pm every day.

```
2024-12-06 08:26:37,275 [INFO]: Spark session created successfully.
2024-12-06 08:26:51,639 [INFO]: Data successfully loaded from S3.
2024-12-06 08:26:51,963 [INFO]: Data transformations applied successfully.
2024-12-06 08:27:07,044 [INFO]: Data successfully written to s3a://arnanand-superstore-bucket/car_data_processed.csv
2024-12-06 08:27:07,044 [INFO]: Processed file s3a://arnanand-superstore-bucket/car_data_processed.csv
2024-12-06 08:27:07,115 [INFO]: Notification sent: Success
2024-12-06 08:27:07,116 [INFO]: Closing down clientserver connection
```

```
[ec2-user@ip-172-31-11-64 ~]$ sudo service crond start
Redirecting to /bin/systemctl start crond.service
[ec2-user@ip-172-31-11-64 ~]$ sudo chkconfig crond on
Note: Forwarding request to 'systemctl enable crond.service'.
[ec2-user@ip-172-31-11-64 ~]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
[ec2-user@ip-172-31-11-64 ~]$
```

Fig .11: A cron job to automate the script

Notifications were integrated using AWS SNS to alert users about the pipeline's progress or errors (Fig. 12). This automation significantly reduced manual intervention and ensured timely updates, making the pipeline both robust and efficient.

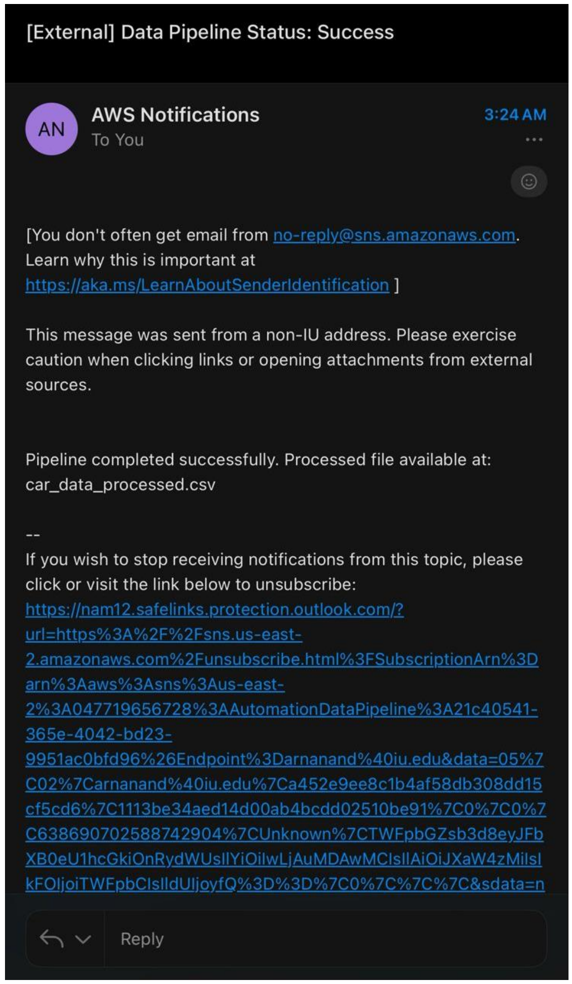


Fig. 12: Successful receipt of pipeline notification via email

3. Results

3.1. Sagemaker Autopilot

The autopilot experiment yielded satisfactory results based on the input processed data, along with some useful intermediary visualizations during the training process. Fig. 13 shows the RMSE of 62.266 and MSE of 3877.055 for the predicted profit value based on the best-performing model. This result is great in terms of accuracy, given the large value range of the profit column in the thousands and tens of thousands. Moreover, it also shows that mmr and selling price have the most impact on the final prediction of profit.

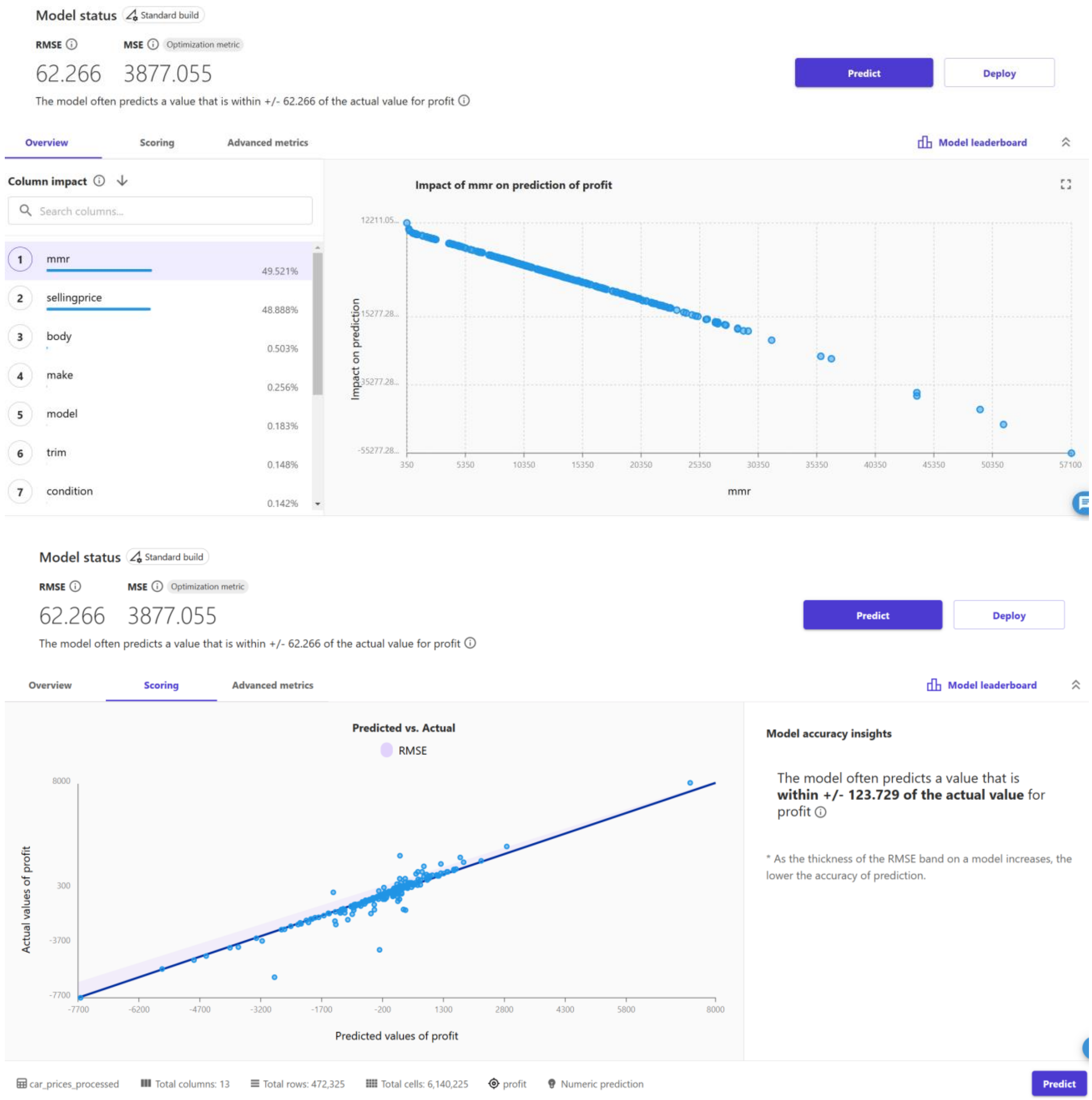


Fig. 13: Results of the Autopilot experiment

The model leaderboard (Fig. 14) shows the top 10 performing models generated and trained by Autopilot. The best performing model sits at the top, namely, FULL-t1047719656728Canvas1733446000994.

Its R^2 value of 99.87% shows the model captures almost all the variance in the data, meaning it's highly accurate. A low RMSE coupled with a near-perfect R^2 indicates the model performs exceptionally well for most predictions.

Select | Build | Analyze | Predict | Deploy

Model leaderboard

Search leaderboard

Model name ↓	MSE Optimization	MAE	RMSE	R2	Inference latency (seconds)
FULL-t1047719656728Canvas1733446000994 Default model	3877.055	22.670	62.266	99.870%	0.452
FULL-t8047719656728Canvas1733446000994	35877.922	61.489	189.415	98.794%	0.170
FULL-t7047719656728Canvas1733446000994	75229.703	66.982	274.280	97.470%	0.115
FULL-t6047719656728Canvas1733446000994	75229.695	66.982	274.280	97.470%	0.117
FULL-t5047719656728Canvas1733446000994	75229.703	66.982	274.280	97.470%	0.117
FULL-t4047719656728Canvas1733446000994	35877.926	61.489	189.415	98.794%	0.166
FULL-t3047719656728Canvas1733446000994	35877.926	61.489	189.415	98.794%	0.164
FULL-t2047719656728Canvas1733446000994	75229.695	66.982	274.280	97.470%	0.117
FULL-t10047719656728Canvas1733446000994	65017.969	130.468	254.986	97.814%	0.182
L1-FULL-t9047719656728Canvas1733446000994	455522.125	275.202	674.924	84.683%	0.136

Fig. 14: Model leaderboard

3.2. Visualization

The Quicksight dashboard (Fig. 15) using the processed data contains 4 visualizations

.

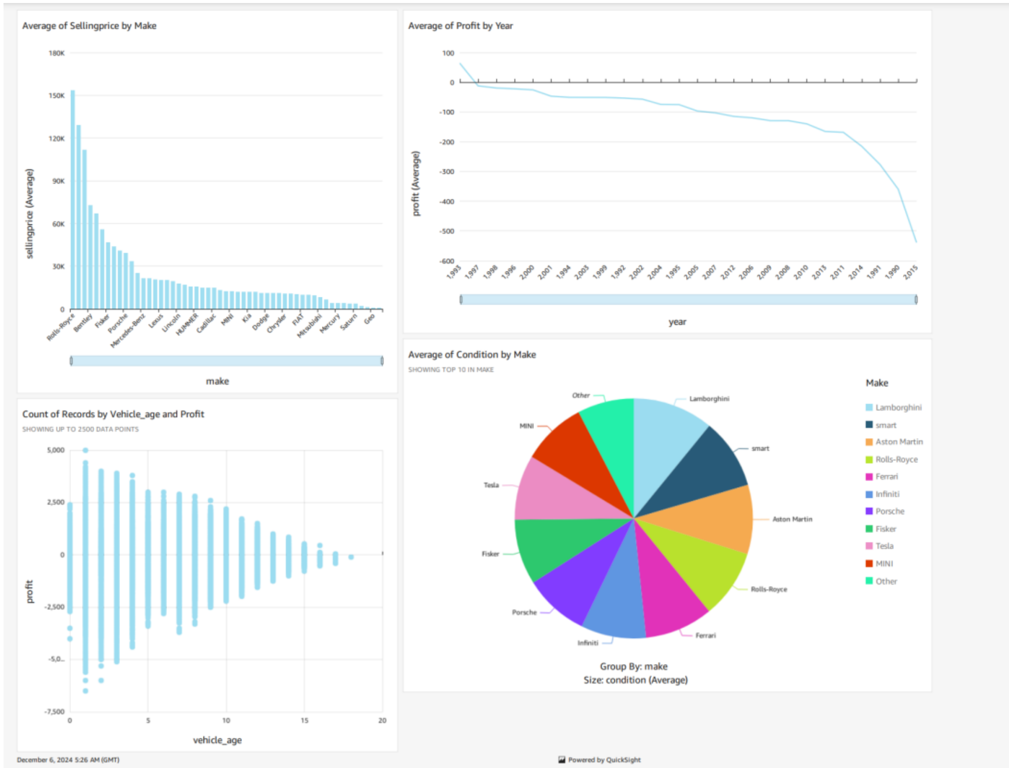


Fig. 15: Quicksight Dashboard

This dashboard provides an analysis of car sales data, focusing on key metrics such as average selling price, profit trends, vehicle condition, and age across various car makes. The top-left bar chart illustrates the average selling price by make, highlighting luxury brands like Lamborghini and Rolls-Royce with significantly higher prices. The top-right line graph tracks average profit trends over time, revealing a decline in profitability in recent years. The bottom-left scatterplot examines the relationship between vehicle age and profit, suggesting that older cars generally yield lower profits. Finally, the bottom-right pie chart represents the average condition of vehicles grouped by make, providing insights into the quality distribution of cars in the dataset.

4. Conclusion

The analysis of car sales data revealed significant insights into the relationship between vehicle characteristics and profitability. The dashboard highlights the importance of factors such as vehicle make, age, and condition in determining average selling prices and profits. Luxury car brands, such as Lamborghini and Rolls-Royce, consistently achieved higher selling prices, while older vehicles generally resulted in lower profit margins. A downward trend in profitability over the years was also evident, suggesting potential market shifts or changes in consumer preferences.

Leveraging advanced analytics and machine learning via AWS infrastructure provided a robust framework for handling and analyzing the extensive dataset. Integrating tools like QuickSight and SageMaker allowed for dynamic data visualization and accurate prediction models, which captured nearly all variance in profit outcomes. Automation of the data pipeline further ensured efficiency and minimized manual intervention. The comprehensive analysis demonstrates the power of combining big data applications with cloud-based tools to drive actionable insights in automotive sales and beyond.

5. References

1. Amazon Web Services. (2024). *AWS Command Line Interface Documentation*. Retrieved from AWS CLI Documentation
2. Apache Software Foundation. (2024). *Apache Spark Documentation*. Retrieved from Apache Spark Documentation
3. Kaggle. (2024). Vehicle Sales *Data*. Retrieved from Kaggle Car Prices
4. Amazon Web Services. (2024). *Amazon SageMaker Overview*. Retrieved from AWS SageMaker

5. Hadoop Apache Foundation. (2024). *Apache Hadoop Documentation*. Retrieved from Hadoop Documentation
6. Amazon Web Services. (2024). *Amazon QuickSight Documentation*. Retrieved from AWS QuickSight
7. Spark SQL. (2024). *Structured Data Processing*. Retrieved from Spark SQL
8. Stack Overflow. (2024). *Configuring Environment Variables for AWS*. Retrieved from Stack Overflow AWS Setup
9. Medium. (2024). *Big Data Pipelines with AWS*. Retrieved from Medium Big Data AWS
10. Amazon Web Services. (2024). *Amazon Simple Notification Service Documentation*. Retrieved from AWS SNS
11. Towards Data Science. (2024). *Building Machine Learning Models with SageMaker*. Retrieved from TDS SageMaker