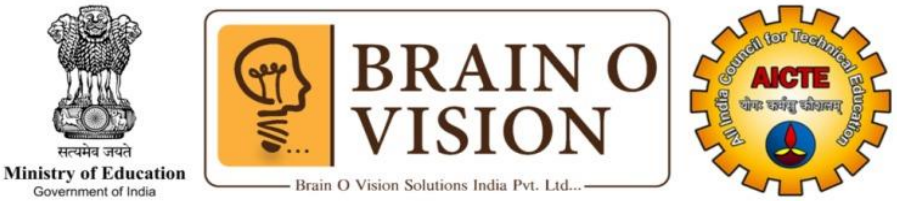
VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE - New Delhi & Affiliated to JNTU-GV, Vizianagaram)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.



Team Name: Code Freaks

Team members:-

- Malla. Madhavi sri lakshmi
- Rayi Navya
- Mycharla Deepak
- Dasireddy. Chaitanya
- Devupalli. Pardava krishnam naidu
- Avula. Ragavendra

Problem Statement:

Industry: Automobile

Department: Manufacturing pain points

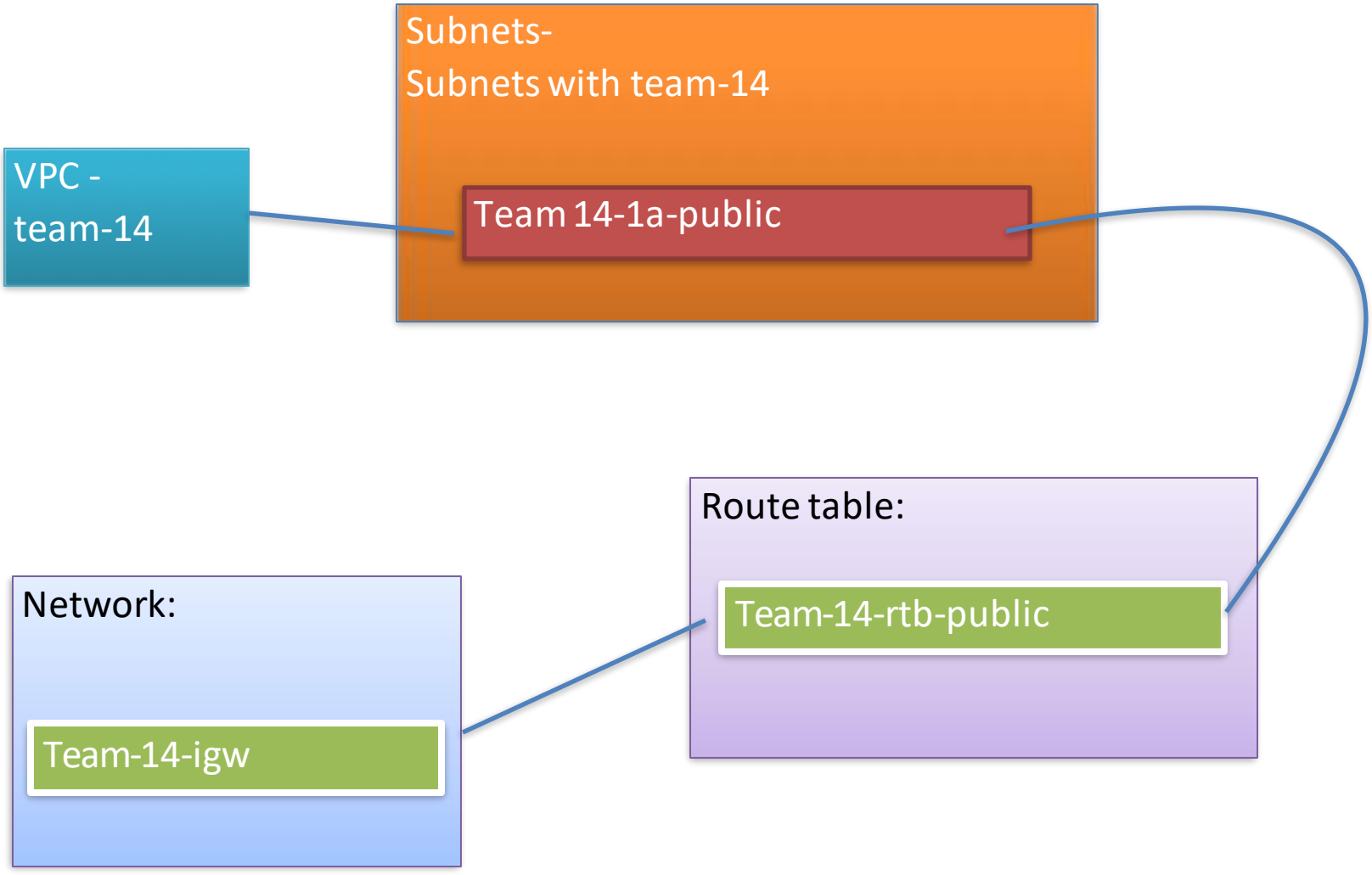
--Quality control

Problem statement:

The Ola Electric Scooter S1 is encountering safety concerns due to incidents of battery burning and break down of front fork, requiring a thorough review and redesign of the battery management system and prefabricated safety checks for enhanced rider safety.

Level 1: (Network and problem statement)

VPC block diagram:



Note: problem statement is picked according documentation

About VPC:

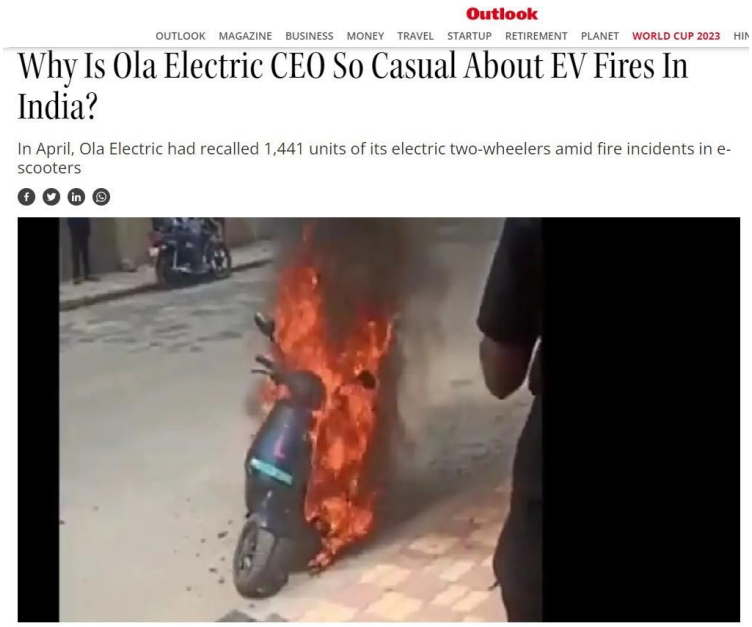
With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

FEATURES:

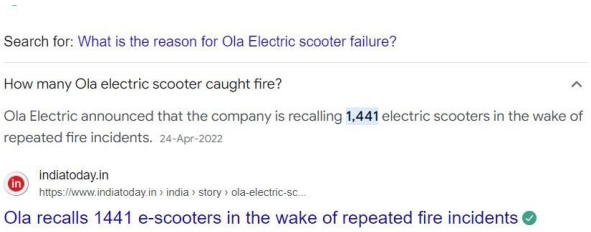
- Virtual private clouds (VPC)
- Subnets
- IP addressing
- Routing
- Gateways and endpoints
- Transit gateways
- VPN connectionS

Issues:

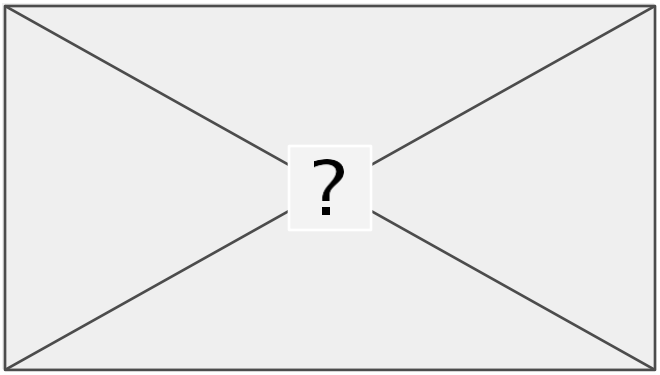
Ref: https://www.cartoq.com/ola-s1-pro-fire-in-bhopal/



Ref: https://www.indiatoday.in/india/story/ola-electric-scooter-recall-fire-incident-battery-system-1941302-2022-04-24



Ref: https://www.moneycontrol.com/news/



Ref: https://www.outlookindia.com/business/electric-scooter-fire-


outlookindia.com/bus

BUSINESS

Why Is Ola Electric CEO So Casual About EV Fires In Indi...

recalling defective batches.

"Several mishaps involving electric two-wheelers have come to light in the last two months. It is most unfortunate some people have lost their lives and several have been injured. We have constituted an expert committee to enquire into these incidents and make recommendations on remedial steps. Based on the reports, we will issue necessary orders on the defaulting companies," Gadkari said.

But not just the government, even the auto industry veterans have slammed the new EV companies, questioning their underlying "process of manufacturing."

Bajaj Auto Managing Director Rajiv Bajaj earlier this month said the people who have no business being in the business of EVs are trying to be in this space.

"The issue is not the fire itself. This (such incidents) happened in the vehicles with internal combustion engines as well. The

Level 2: Feature engineering

Step 1: selection

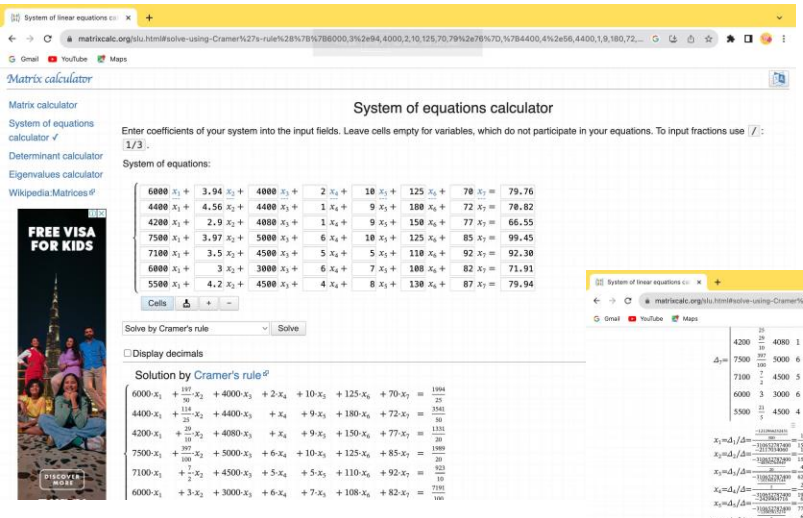
```
Step 1: Selection
Attributes:
design ----------x
software integration -----------------x
company brand---------x
battery capacity +++++++++
tyre width and height--x
acceleration +++++++++
range +++++++
speed +++++++++++++
motor type++++++++
lights ----------x
load capacity ++++++
base height--------------x
battery charging speed ------x
breaks ----------------x
color ---------x
battery power ++++++++++++++++
storage space --------x
gps -------x
features ---------x
```

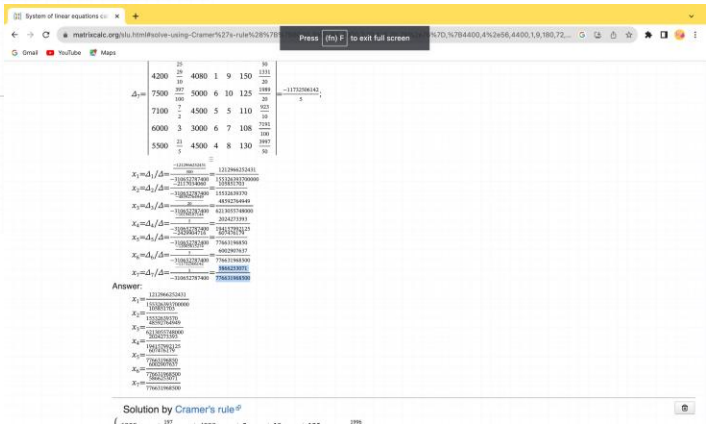
Step 2:Transformation

here
Motor type(kw)
Battery capacity(in kwh)
Battery power (in w)
range(100-110=1, 110-120=2,

A	B	C	D	E	F	G	H
Motor type(kw)	Battery capacit	Battery power(i	range(100-110=1,	acceleration(0-4(	load capacity(in kg)	satisfaction rate(%)	performance rate
6000	3.94	4000	2	10	125	70	79.76673697
4400	4.56	4400	1	9	180	72	70.82696664
4200	2.9	4080	1	9	150	77	66.55651902
7500	3.97	5000	6	10	125	85	99.44512147
7100	3.5	4500	5	5	110	92	92.3013827
6000	3	3000	6	7	108	82	71.91625654
5500	4.2	4500	4	8	130	87	79.94062964

Cramer's rule:

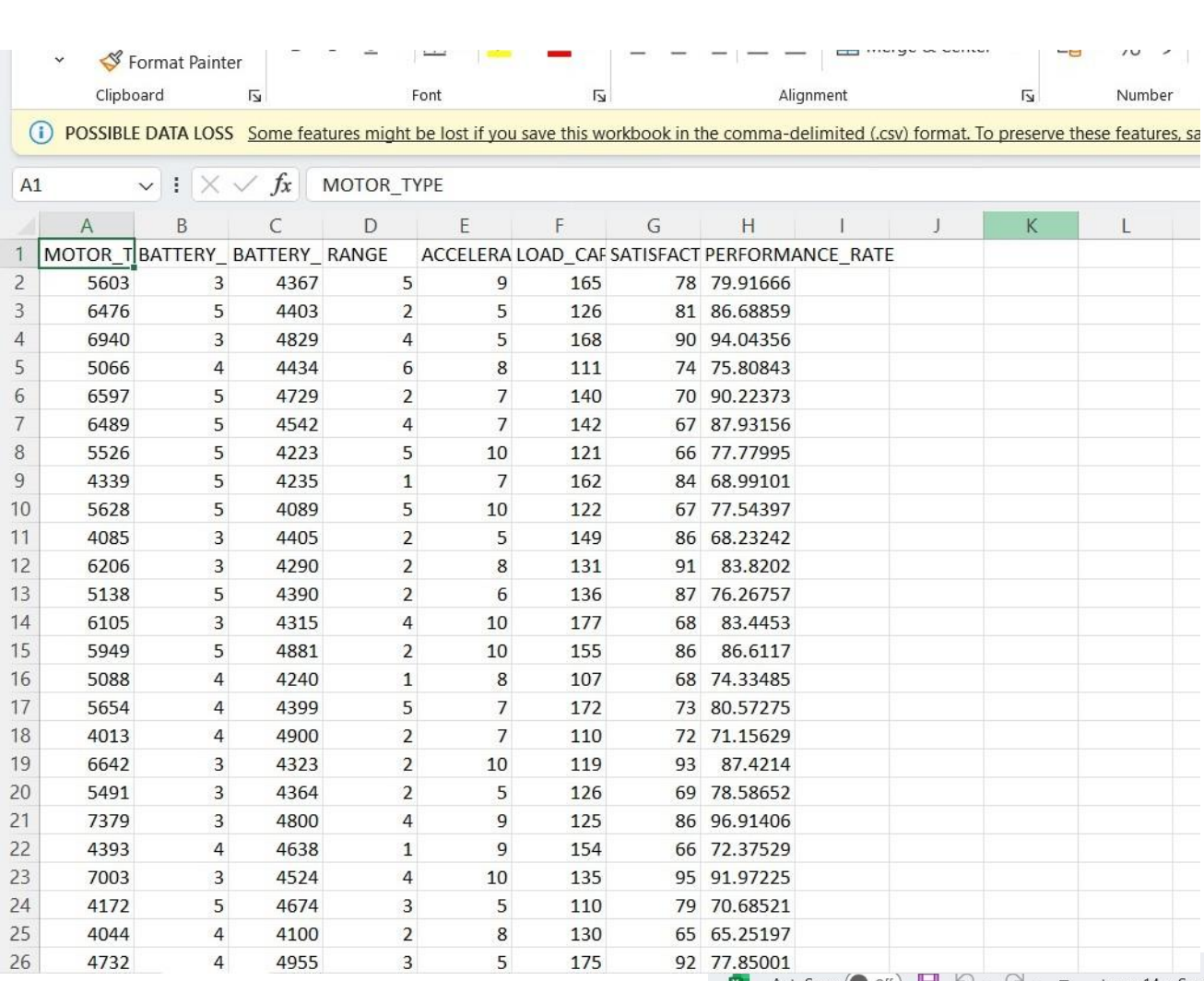




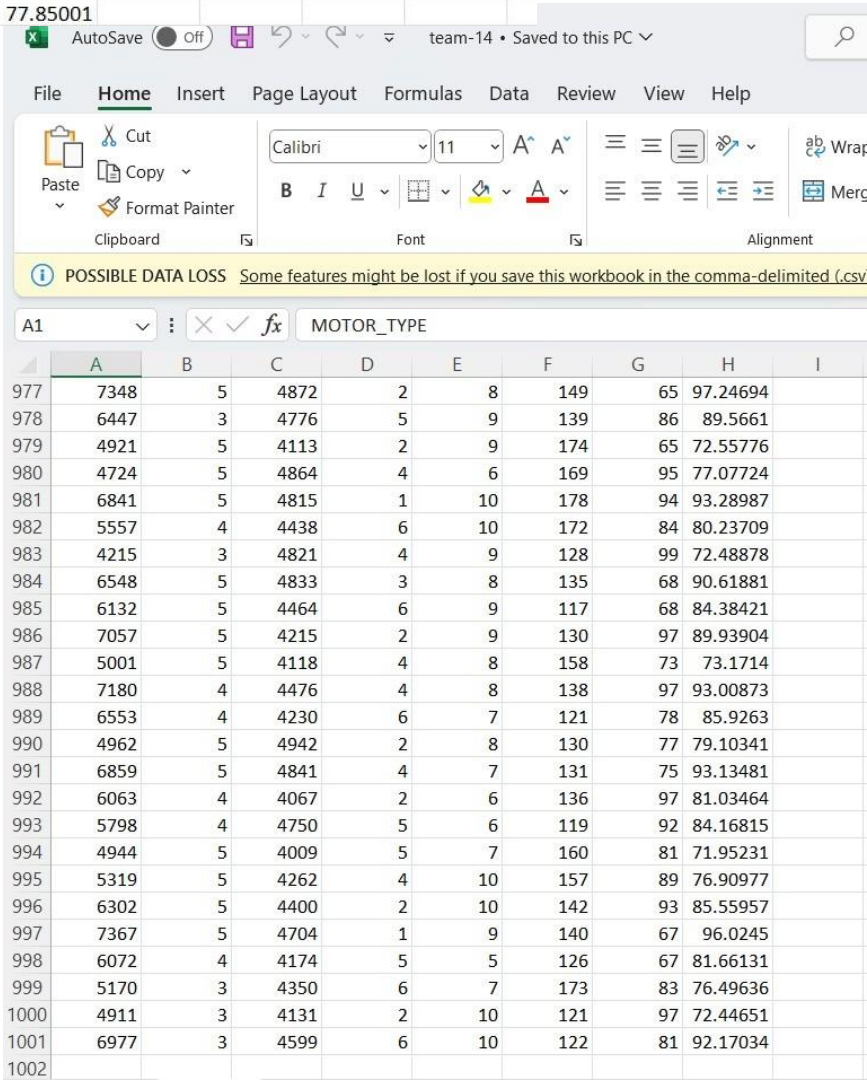
Step 3: Extraction

Code:

```
import random
import pandas as pd
[w1,w2,w3,w4,w5,w6,w7] =
[0.00781,0.00682,0.00782,0.01042,0.00782,0.00773,0.00755]
vals = []
for i in range(1000):
    x1 = random.randint(1, 10)
    x2 = random.randint(1, 5)
    x3 = random.randint(0, 100)
    x4 = random.randint(0, 70)
    x5 = random.randint(1, 2)
    x6 = random.randint(1, 10)
    x7 = random.randint(1, 2)
    eq = w1*x1+w2*x2+w3*x3+w4*x4+w5*x5+w6*x6+w7*x7
    vals.append([x1,x2,x3,x4,x5,x6,x7,eq])
df =
pd.DataFrame(vals,columns=['MOTOR_TYPE','BATTERY_CAPACIT
Y','BATTERY_POWER','RANGE','ACCELERATION','LOAD_CAPACITY'
,'SATISFACTION_RATE','PERFORMANCE_RATE'])
df.to_csv('team-14.csv',index=False
```



	A	B	C	D	E	F	G	H	I	J	K	L
1	MOTOR_T	BATTERY_	BATTERY_	RANGE	ACCELERA	LOAD_CAF	SATISFACT	PERFORMANCE_RATE				
2	5603	3	4367	5	9	165	78	79.91666				
3	6476	5	4403	2	5	126	81	86.68859				
4	6940	3	4829	4	5	168	90	94.04356				
5	5066	4	4434	6	8	111	74	75.80843				
6	6597	5	4729	2	7	140	70	90.22373				
7	6489	5	4542	4	7	142	67	87.93156				
8	5526	5	4223	5	10	121	66	77.77995				
9	4339	5	4235	1	7	162	84	68.99101				
10	5628	5	4089	5	10	122	67	77.54397				
11	4085	3	4405	2	5	149	86	68.23242				
12	6206	3	4290	2	8	131	91	83.8202				
13	5138	5	4390	2	6	136	87	76.26757				
14	6105	3	4315	4	10	177	68	83.4453				
15	5949	5	4881	2	10	155	86	86.6117				
16	5088	4	4240	1	8	107	68	74.33485				
17	5654	4	4399	5	7	172	73	80.57275				
18	4013	4	4900	2	7	110	72	71.15629				
19	6642	3	4323	2	10	119	93	87.4214				
20	5491	3	4364	2	5	126	69	78.58652				
21	7379	3	4800	4	9	125	86	96.91406				
22	4393	4	4638	1	9	154	66	72.37529				
23	7003	3	4524	4	10	135	95	91.97225				
24	4172	5	4674	3	5	110	79	70.68521				
25	4044	4	4100	2	8	130	65	65.25197				
26	4732	4	4955	3	5	175	92	77.85001				



	A	B	C	D	E	F	G	H	I
977	7348	5	4872	2	8	149	65	97.24694	
978	6447	3	4776	5	9	139	86	89.5661	
979	4921	5	4113	2	9	174	65	72.55776	
980	4724	5	4864	4	6	169	95	77.07724	
981	6841	5	4815	1	10	178	94	93.28987	
982	5557	4	4438	6	10	172	84	80.23709	
983	4215	3	4821	4	9	128	99	72.48878	
984	6548	5	4833	3	8	135	68	90.61881	
985	6132	5	4464	6	9	117	68	84.38421	
986	7057	5	4215	2	9	130	97	89.93904	
987	5001	5	4118	4	8	158	73	73.1714	
988	7180	4	4476	4	8	138	97	93.00873	
989	6553	4	4230	6	7	121	78	85.9263	
990	4962	5	4942	2	8	130	77	79.10341	
991	6859	5	4841	4	7	131	75	93.13481	
992	6063	4	4067	2	6	136	97	81.03464	
993	5798	4	4750	5	6	119	92	84.16815	
994	4944	5	4009	5	7	160	81	71.95231	
995	5319	5	4262	4	10	157	89	76.90977	
996	6302	5	4400	2	10	142	93	85.55957	
997	7367	5	4704	1	9	140	67	96.0245	
998	6072	4	4174	5	5	126	67	81.66131	
999	5170	3	4350	6	7	173	83	76.49636	
1000	4911	3	4131	2	10	121	97	72.44651	
1001	6977	3	4599	6	10	122	81	92.17034	
1002									

Level 3: (Data framing and Github)

Github:

main · 1 branch · 0 tags · Go to file · Add file · Code

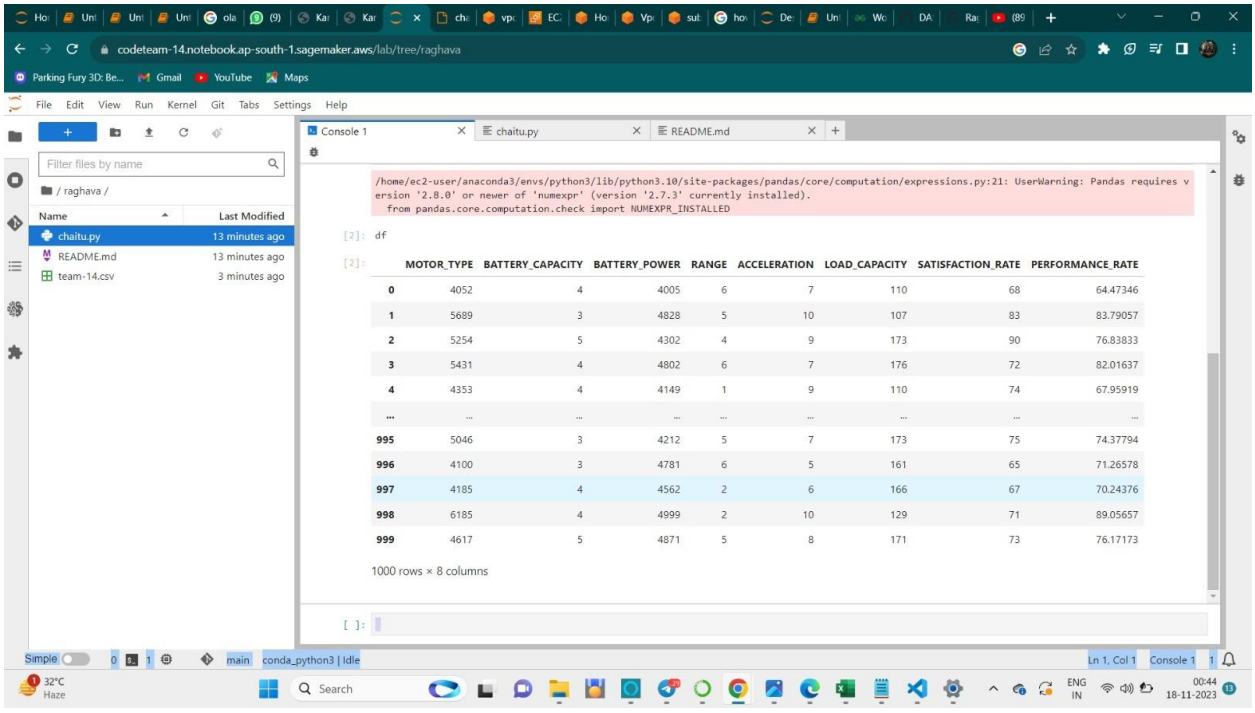
DASIREDDYCHAITANYA Add files via upload — abebb0e 2 minutes ago — 2 commits

README.md	Initial commit	2 weeks ago
cal.html	Initial commit	2 weeks ago
chaitu.css	Initial commit	2 weeks ago
chaitu.py	Add files via upload	2 minutes ago
index.js	Initial commit	2 weeks ago
port.css	Initial commit	2 weeks ago
portfolio.html	Initial commit	2 weeks ago
style.css	Initial commit	2 weeks ago
viganan.html	Initial commit	2 weeks ago

README.md

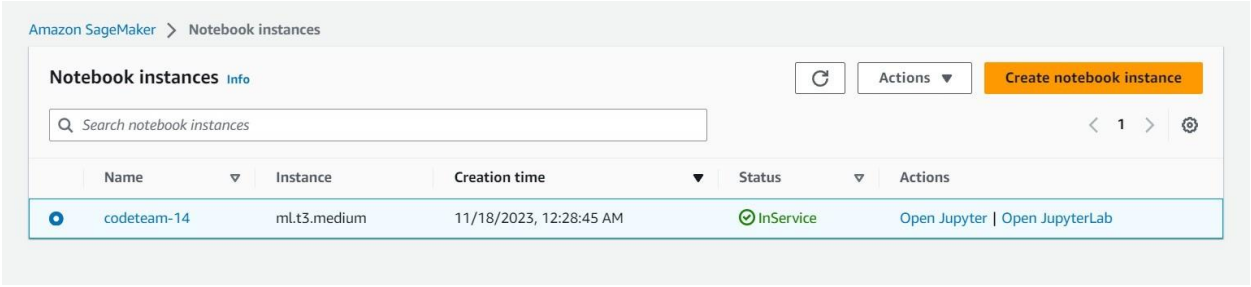
my-first-intern-hacktoberfest

Data framing:



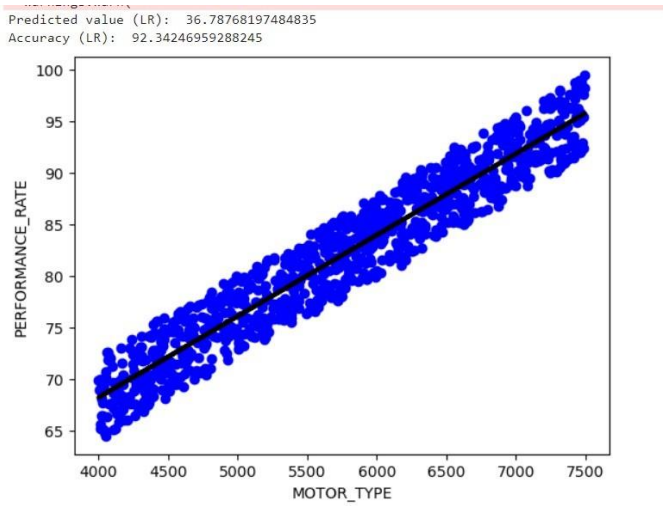
Level 4: (Notebook instance and ML model)

Notebook instance:

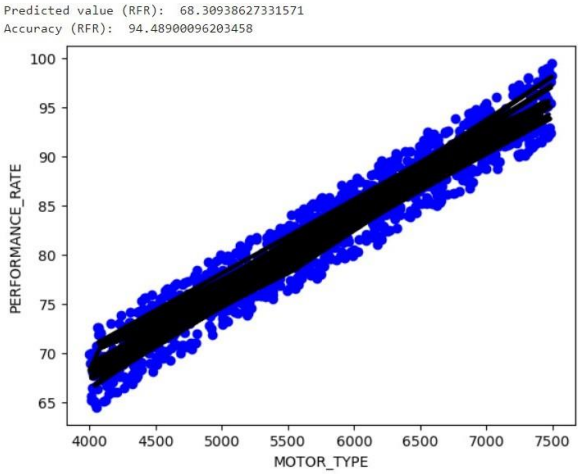


ML model:

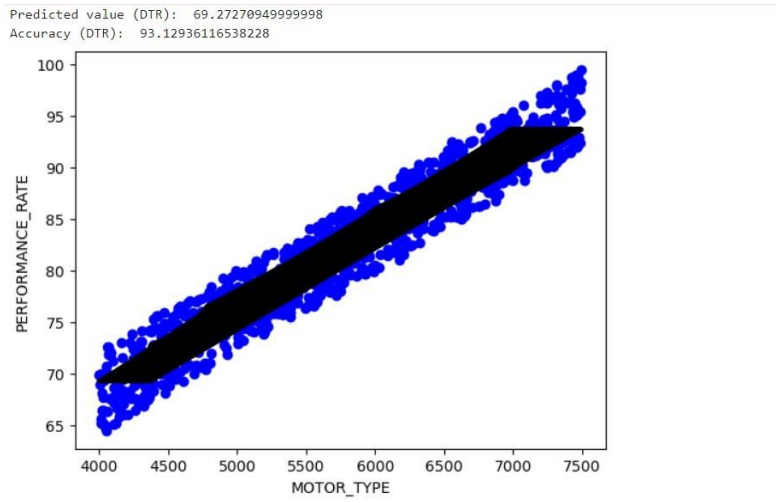
Linear regression-



Random forest regressor:



Decision tree regressor-



Evaluation:

Linear regression-

```
[21]: from sklearn.metrics import mean_absolute_error
      from sklearn.linear_model import LinearRegression
      regressor=LinearRegression()
      regressor.fit(x_train,y_train)
      score=regressor.score(x_train,y_train)
      y_predL=regressor.predict(x_test)
      print("Accuracy of Linear Regression",score)
      y_predL=regressor.predict(x_test)
      from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
      mse=mean_squared_error(y_test,y_pred)
      r2=r2_score(y_test,y_pred)
      mae=mean_absolute_error(y_test,y_pred)
      print("Mean Square Error of Linear Regression",mse)
      print("R2 score of Linear Regression",r2)
      print("Mean absolute Error of Linear Regression",mae)

      Accuracy of Linear Regression 1.0
      Mean Square Error of Linear Regression 0.14466155531357702
      R2 score of Linear Regression 0.9981407156391697
      Mean absolute Error of Linear Regression 0.30650008000000767
```

Random forest regressor:

```
[19]: from sklearn.ensemble import RandomForestRegressor
      rf_model=RandomForestRegressor(n_estimators=100, random_state=42)
      rf_model.fit(x_train, y_train)
      y_pred=rf_model.predict(x_test)

      from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
      mse=mean_squared_error(y_test,y_pred)
      r2=r2_score(y_test,y_pred)
      mae=mean_absolute_error(y_test,y_pred)
      print("Mean Square Error of RandomForest Regression",mse)
      print("R2 score of RandomForest Regression",r2)
      print("Mean absolute Error of RandomForest Regression",mae)

      Mean Square Error of RandomForest Regression 0.14466155531357702
      R2 score of RandomForest Regression 0.9981407156391697
      Mean absolute Error of RandomForest Regression 0.30650008000000767
```

Conclusion:

Using more efficiency materials and checking the vehicle more times during manufacturing