In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
d1= pd.read_csv(r"Downloads\ML Data - data (1).csv")
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_4460\2590878669.py:1: DtypeWarning: Columns (21,32,33,60) have mixed types. Specify
dtype option on import or set low_memory=False.
  d1= pd.read_csv(r"Downloads\ML Data - data (1).csv")
```

In [3]:
```
d1.head()
```

Out[3]:

	appointmentId	inspectionStartTime	year	month	engineTransmission_battery_value	engineTransmission_battery_cc_value_0	engineTransmission_batter
0	aj_01	02/03/2019 15:43	2008	8	No	Weak	
1	aj_02	1/16/19 13:02	2007	5	Yes	NaN	
2	aj_03	02/09/2019 13:31	2012	5	Yes	NaN	
3	aj_04	1/18/19 11:02	2013	1	Yes	NaN	
4	aj_05	1/27/19 12:12	2011	7	Yes	NaN	

5 rows × 73 columns

In [4]:
```
d1.shape
```

Out[4]:
```
(26307, 73)
```

In [5]:
```
# finding null values by using pandas
d1.isnull().sum()
```

Out[5]:

```
appointmentId                        0
inspectionStartTime                  0
year                                 0
month                                0
engineTransmission_battery_value     0
                                   ...
engineTransmission_comments_value_3    26248
engineTransmission_comments_value_4    26293
fuel_type                            0
odometer_reading                     0
rating_engineTransmission            0
Length: 73, dtype: int64
```

In [6]:

```
d1.dropna(how='all')
```

Out[6]:

	appointmentId	inspectionStartTime	year	month	engineTransmission_battery_value	engineTransmission_battery_cc_value_0	engineTransmission_b
0	aj_01	02/03/2019 15:43	2008	8	No	Weak	
1	aj_02	1/16/19 13:02	2007	5	Yes	NaN	
2	aj_03	02/09/2019 13:31	2012	5	Yes	NaN	
3	aj_04	1/18/19 11:02	2013	1	Yes	NaN	
4	aj_05	1/27/19 12:12	2011	7	Yes	NaN	
...	...	...	...	...	...	...	
26302	aj_26303	03/10/2019 13:08	2013	3	Yes	NaN	
26303	aj_26304	04/12/2019 13:59	2007	8	No	Weak	
26304	aj_26305	2/28/19 10:42	2004	7	Yes	NaN	
26305	aj_26306	04/02/2019 12:21	2010	12	Yes	NaN	
26306	aj_26307	04/06/2019 13:09	2015	11	Yes	NaN	

26307 rows × 73 columns

In [7]:

```
d1['engineTransmission_gearShifting_cc_value_2'].value_counts()
```

Out[7]:

```
Abnormal Noise                                63
Automatic Transmission not working properly    3
Name: engineTransmission_gearShifting_cc_value_2, dtype: int64
```

In [8]:

```
# drop column which contain more than 80% null values
null_var=d1.isnull().sum()/d1.shape[0]*100
drop_column=null_var[null_var>80].keys()
drop_column
```

```
Out[8]: Index(['engineTransmission_battery_cc_value_0',
               'engineTransmission_battery_cc_value_1',
               'engineTransmission_battery_cc_value_2',
               'engineTransmission_battery_cc_value_3',
               'engineTransmission_battery_cc_value_4',
               'engineTransmission_engineOilLevelDipstick_cc_value_0',
               'engineTransmission_engineOil_cc_value_3',
               'engineTransmission_engineOil_cc_value_4',
               'engineTransmission_engineOil_cc_value_5',
               'engineTransmission_engineOil_cc_value_6',
               'engineTransmission_engineOil_cc_value_7',
               'engineTransmission_engineOil_cc_value_8',
               'engineTransmission_engineOil_cc_value_9',
               'engineTransmission_engine_cc_value_1',
               'engineTransmission_engine_cc_value_2',
               'engineTransmission_engine_cc_value_3',
               'engineTransmission_engine_cc_value_4',
               'engineTransmission_engine_cc_value_5',
               'engineTransmission_engine_cc_value_6',
               'engineTransmission_engine_cc_value_7',
               'engineTransmission_engine_cc_value_8',
               'engineTransmission_engine_cc_value_9',
               'engineTransmission_engine_cc_value_10',
               'engineTransmission_coolant_cc_value_1',
               'engineTransmission_coolant_cc_value_2',
               'engineTransmission_coolant_cc_value_3',
               'engineTransmission_engineSound_cc_value_3',
               'engineTransmission_engineSound_cc_value_4',
               'engineTransmission_engineSound_cc_value_5',
               'engineTransmission_clutch_cc_value_1',
               'engineTransmission_clutch_cc_value_2',
               'engineTransmission_clutch_cc_value_3',
               'engineTransmission_clutch_cc_value_4',
               'engineTransmission_clutch_cc_value_5',
               'engineTransmission_clutch_cc_value_6',
               'engineTransmission_gearShifting_cc_value_0',
               'engineTransmission_gearShifting_cc_value_1',
               'engineTransmission_gearShifting_cc_value_2',
               'engineTransmission_comments_value_0',
               'engineTransmission_comments_value_1',
               'engineTransmission_comments_value_2',
               'engineTransmission_comments_value_3',
               'engineTransmission_comments_value_4'],
              dtype='object')
```

In [9]:

```
missing_value_clm_gre_20=['engineTransmission_battery_cc_value_0',
       'engineTransmission_battery_cc_value_1',
       'engineTransmission_battery_cc_value_2',
       'engineTransmission_battery_cc_value_3',
       'engineTransmission_battery_cc_value_4',
       'engineTransmission_engineOilLevelDipstick_cc_value_0',
       'engineTransmission_engineOil_cc_value_3',
       'engineTransmission_engineOil_cc_value_4',
       'engineTransmission_engineOil_cc_value_5',
       'engineTransmission_engineOil_cc_value_6',
       'engineTransmission_engineOil_cc_value_7',
       'engineTransmission_engineOil_cc_value_8',
       'engineTransmission_engineOil_cc_value_9',
       'engineTransmission_engine_cc_value_1',
       'engineTransmission_engine_cc_value_2',
       'engineTransmission_engine_cc_value_3',
       'engineTransmission_engine_cc_value_4',
       'engineTransmission_engine_cc_value_5',
       'engineTransmission_engine_cc_value_6',
       'engineTransmission_engine_cc_value_7',
       'engineTransmission_engine_cc_value_8',
       'engineTransmission_engine_cc_value_9',
       'engineTransmission_engine_cc_value_10',
       'engineTransmission_coolant_cc_value_1',
       'engineTransmission_coolant_cc_value_2',
       'engineTransmission_coolant_cc_value_3',
       'engineTransmission_engineSound_cc_value_3',
       'engineTransmission_engineSound_cc_value_4',
       'engineTransmission_engineSound_cc_value_5',
       'engineTransmission_clutch_cc_value_1',
       'engineTransmission_clutch_cc_value_2',
       'engineTransmission_clutch_cc_value_3',
       'engineTransmission_clutch_cc_value_4',
       'engineTransmission_clutch_cc_value_5',
       'engineTransmission_clutch_cc_value_6',
       'engineTransmission_gearShifting_cc_value_0',
       'engineTransmission_gearShifting_cc_value_1',
       'engineTransmission_gearShifting_cc_value_2',
       'engineTransmission_comments_value_0',
       'engineTransmission_comments_value_1',
       'engineTransmission_comments_value_2',
       'engineTransmission_comments_value_3',
            'engineTransmission_comments_value_4']
df2=d1.drop(columns=missing_value_clm_gre_20)
```

In [10]:

```
df2.shape
```

Out[10]:

```
(26307, 30)
```

In [11]:

```
df2.isnull().keys()
```

Out[11]:

```
Index(['appointmentId', 'inspectionStartTime', 'year', 'month',
       'engineTransmission_battery_value',
       'engineTransmission_engineoilLevelDipstick_value',
       'engineTransmission_engineOil',
       'engineTransmission_engineOil_cc_value_0',
       'engineTransmission_engineOil_cc_value_1',
       'engineTransmission_engineOil_cc_value_2',
       'engineTransmission_engine_value',
       'engineTransmission_engine_cc_value_0',
       'engineTransmission_coolant_value',
       'engineTransmission_coolant_cc_value_0',
       'engineTransmission_engineMounting_value',
       'engineTransmission_engineMounting_cc_value_0',
       'engineTransmission_engineSound_value',
       'engineTransmission_engineSound_cc_value_0',
       'engineTransmission_engineSound_cc_value_1',
       'engineTransmission_engineSound_cc_value_2',
       'engineTransmission_exhaustSmoke_value',
       'engineTransmission_exhaustSmoke_cc_value_0',
       'engineTransmission_engineBlowByBackCompression_value',
       'engineTransmission_engineBlowByBackCompression_cc_value_0',
       'engineTransmission_clutch_value',
       'engineTransmission_clutch_cc_value_0',
       'engineTransmission_gearShifting_value', 'fuel_type',
       'odometer_reading', 'rating_engineTransmission'],
      dtype='object')
```

In [12]:

```
# finding keys which contain null values
isnull_per=df2.isnull().mean()*100
miss_vars=isnull_per[isnull_per>0].keys()
miss_vars
```

Out[12]:

```
Index(['engineTransmission_engineOil_cc_value_0',
       'engineTransmission_engineOil_cc_value_1',
       'engineTransmission_engineOil_cc_value_2',
       'engineTransmission_engine_cc_value_0',
       'engineTransmission_coolant_cc_value_0',
       'engineTransmission_engineMounting_cc_value_0',
       'engineTransmission_engineSound_cc_value_0',
       'engineTransmission_engineSound_cc_value_1',
       'engineTransmission_engineSound_cc_value_2',
       'engineTransmission_exhaustSmoke_cc_value_0',
       'engineTransmission_clutch_cc_value_0'],
      dtype='object')
```

In [13]:

```python
# fill null values by using mode because of categorical values
for i in miss_vars:
    df2[i]=df2[i].fillna(df2[i].mode()[0])
df2.isnull().sum()
```

Out[13]:

```
appointmentId                                                0
inspectionStartTime                                          0
year                                                         0
month                                                        0
engineTransmission_battery_value                             0
engineTransmission_engineoilLevelDipstick_value              0
engineTransmission_engineOil                                 0
engineTransmission_engineOil_cc_value_0                      0
engineTransmission_engineOil_cc_value_1                      0
engineTransmission_engineOil_cc_value_2                      0
engineTransmission_engine_value                              0
engineTransmission_engine_cc_value_0                         0
engineTransmission_coolant_value                             0
engineTransmission_coolant_cc_value_0                        0
engineTransmission_engineMounting_value                      0
engineTransmission_engineMounting_cc_value_0                 0
engineTransmission_engineSound_value                         0
engineTransmission_engineSound_cc_value_0                    0
engineTransmission_engineSound_cc_value_1                    0
engineTransmission_engineSound_cc_value_2                    0
engineTransmission_exhaustSmoke_value                        0
engineTransmission_exhaustSmoke_cc_value_0                   0
engineTransmission_engineBlowByBackCompression_value         0
engineTransmission_engineBlowByBackCompression_cc_value_0    0
engineTransmission_clutch_value                              0
engineTransmission_clutch_cc_value_0                         0
engineTransmission_gearShifting_value                        0
fuel_type                                                    0
odometer_reading                                             0
rating_engineTransmission                                    0
dtype: int64
```

In [14]:

```
df2.keys()
```

Out[14]:

```
Index(['appointmentId', 'inspectionStartTime', 'year', 'month',
       'engineTransmission_battery_value',
       'engineTransmission_engineoilLevelDipstick_value',
       'engineTransmission_engineOil',
       'engineTransmission_engineOil_cc_value_0',
       'engineTransmission_engineOil_cc_value_1',
       'engineTransmission_engineOil_cc_value_2',
       'engineTransmission_engine_value',
       'engineTransmission_engine_cc_value_0',
       'engineTransmission_coolant_value',
       'engineTransmission_coolant_cc_value_0',
       'engineTransmission_engineMounting_value',
       'engineTransmission_engineMounting_cc_value_0',
       'engineTransmission_engineSound_value',
       'engineTransmission_engineSound_cc_value_0',
       'engineTransmission_engineSound_cc_value_1',
       'engineTransmission_engineSound_cc_value_2',
       'engineTransmission_exhaustSmoke_value',
       'engineTransmission_exhaustSmoke_cc_value_0',
       'engineTransmission_engineBlowByBackCompression_value',
       'engineTransmission_engineBlowByBackCompression_cc_value_0',
       'engineTransmission_clutch_value',
       'engineTransmission_clutch_cc_value_0',
       'engineTransmission_gearShifting_value', 'fuel_type',
       'odometer_reading', 'rating_engineTransmission'],
      dtype='object')
```

In [15]:

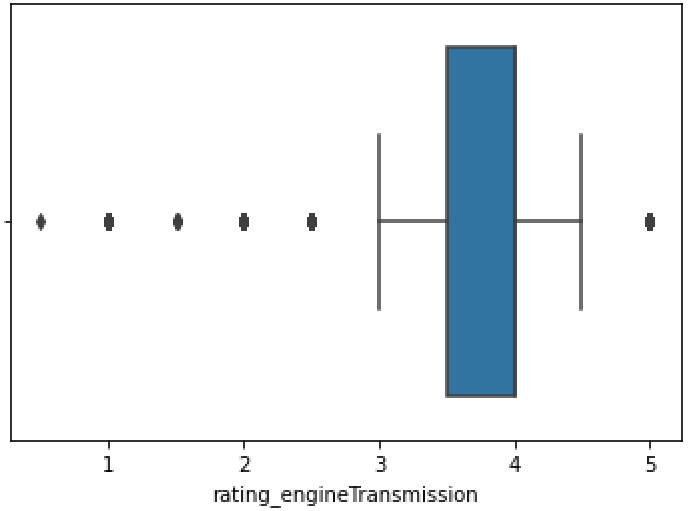
```
df2.shape
```

Out[15]:

```
(26307, 30)
```

In [16]:

```
import seaborn as sns
sns.boxplot(x='rating_engineTransmission',data=df2)
plt.show()
```



In [31]:
```
def remove_outlier(df_in, col_name):
    q1 = df_in[col_name].quantile(0.25)
    q3 = df_in[col_name].quantile(0.75)
    iqr = q3-q1 #Interquartile range
    fence_low  = q1-1.5*iqr
    fence_high = q3+1.5*iqr
    df_out = df_in.loc[(df_in[col_name] > fence_low) & (df_in[col_name] < fence_high)]
    return df_out
remove_outlier(df2,'rating_engineTransmission')
df2.shape
```

Out[31]:
```
(26307, 30)
```

In [21]:
```
df3=['appointmentId', 'inspectionStartTime',
       'engineTransmission_battery_value',
       'engineTransmission_engineoilLevelDipstick_value',
       'engineTransmission_engineOil',
       'engineTransmission_engineOil_cc_value_0',
       'engineTransmission_engineOil_cc_value_1',
       'engineTransmission_engineOil_cc_value_2',
       'engineTransmission_engine_value',
       'engineTransmission_engine_cc_value_0',
       'engineTransmission_coolant_value',
       'engineTransmission_coolant_cc_value_0',
       'engineTransmission_engineMounting_value',
       'engineTransmission_engineMounting_cc_value_0',
       'engineTransmission_engineSound_value',
       'engineTransmission_engineSound_cc_value_0',
       'engineTransmission_engineSound_cc_value_1',
       'engineTransmission_engineSound_cc_value_2',
       'engineTransmission_exhaustSmoke_value',
       'engineTransmission_exhaustSmoke_cc_value_0',
       'engineTransmission_engineBlowByBackCompression_value',
       'engineTransmission_engineBlowByBackCompression_cc_value_0',
       'engineTransmission_clutch_value',
       'engineTransmission_clutch_cc_value_0',
       'engineTransmission_gearShifting_value', 'fuel_type','year'
       ]
```

In [37]:

```
for i in df3:
    df2[i] = pd.Categorical(df2[i]).codes

df3=df2
df3.shape
```

Out[37]:

```
(26307, 30)
```

In [23]:

```
# Data Preprocessing & Model building
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [24]:

```
x=df3.drop(columns=["rating_engineTransmission"])
y=df3["rating_engineTransmission"]
x_train, x_test,y_train, y_test=train_test_split(x,y,test_size=0.2, random_state=1)
```

In [25]:

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR

scaler=StandardScaler()
linear=LinearRegression()
knn=KNeighborsRegressor()
rand=RandomForestRegressor()
svr=SVR()
```

In [38]:

```
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

```
pipe1=make_pipeline(scaler,linear)
pipe1.fit(x_train,y_train)
y_pred_lr=pipe1.predict(x_test)
r2_score(y_test,y_pred_lr)
```

```
Out[38]: 0.41267320996523893
```

```
In [39]: pipe2=make_pipeline(scaler,knn)
         pipe2.fit(x_train,y_train)
         y_pred_knn=pipe2.predict(x_test)
         r2_score(y_test,y_pred_knn)
```

```
Out[39]: 0.6349499184025618
```

```
In [40]: pipe3=make_pipeline(scaler,rand)
         pipe3.fit(x_train,y_train)
         y_pred_ran=pipe3.predict(x_test)
         r2_score(y_test,y_pred_ran)
```

```
Out[40]: 0.7094538464894233
```

```
In [43]: pipe4=make_pipeline(scaler,svr)
         pipe4.fit(x_train,y_train)
         y_pred_svr=pipe4.predict(x_test)
         r2_score(y_test,y_pred_svr)
```

```
Out[43]: 0.6569660261250128
```

```
In [44]: from sklearn.metrics import mean_squared_error
         mean_squared_error(y_test,y_pred_ran)
```

```
Out[44]: 0.20577737077156977
```

```
In [45]: df3.describe()
```

Out[45]:

	appointmentId	inspectionStartTime	year	month	engineTransmission_battery_value	engineTransmission_engineoilLevelDipstick_valu
count	26307.000000	26307.000000	26307.000000	26307.000000	26307.000000	26307.00000
mean	13153.000000	10176.066408	18.856768	4.462006	0.869312	0.98437
std	7594.321102	5874.428394	3.765236	3.583866	0.337065	0.12401
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	6576.500000	5064.500000	16.000000	1.000000	1.000000	1.00000
50%	13153.000000	10172.000000	19.000000	4.000000	1.000000	1.00000
75%	19729.500000	15229.500000	22.000000	8.000000	1.000000	1.00000
max	26306.000000	20319.000000	27.000000	11.000000	1.000000	1.00000

8 rows × 30 columns

In [ ]: