yt8t8wo3g

October 18, 2023

1 IMPORTING REQUIRED LIBRARIES

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
pd.set_option('display.max_columns',25)
pd.set_option('display.max_rows',300)
plt.rcParams['axes.facecolor']='#99ffcc'
import warnings
warnings.filterwarnings('ignore')
```

2 IMPORTING DATASETS AND MERGING INTO SINGLE DATAFRAME

[75]:
```
sales=pd.read_csv('/kaggle/input/cognizant-artificial-intelligence/data for␣
 ↪modeling/sales.csv')
#sensor stock levels
ssl=pd.read_csv('/kaggle/input/cognizant-artificial-intelligence/data for␣
 ↪modeling/sensor_stock_levels.csv')
#sensor storage temperature
sst=pd.read_csv('/kaggle/input/cognizant-artificial-intelligence/data for␣
 ↪modeling/sensor_storage_temperature.csv')
```

[76]:
```
sales.sample(4)
```

[76]:

	Unnamed: 0	transaction_id	timestamp	product_id	category	customer_type
7316	7316	4d39b17e-465a-46f4-8c80-39344eb2a02e	2022-03-05 10:15:21	cb8ce677-5883-46e8-9c85-bf1c88841ef9	kitchen	gold
847	847	80d1bfdf-57e5-4bd5-bbb2-f098cfd0df0e	2022-03-06 18:42:11	0ddc2379-adba-4fb0-aa97-19fcafc738a1	fruit	non-member
1033	1033	476eebf8-f531-4de6-a0c5-d3dfc79959ac	2022-03-01 17:37:17	abc7ecb4-9cbf-4276-8d2d-7bdac2ae5e9f	vegetables	non-member
1976	1976	b5d13427-1a28-454f-9cad-c2224599256f	2022-03-03 18:43:07			

```
1976  16e6119d-2e17-49b8-82e5-07f363b5cb08  refrigerated items      standard

      unit_price  quantity  total payment_type
7316       18.99         1  18.99     e-wallet
847         3.99         1   3.99  credit card
1033        4.99         1   4.99   debit card
1976        4.49         3  13.47  credit card
```

[77]:
```
ssl.head()
#['id', 'timestamp', 'product_id', 'estimated_stock_pct']
```

[77]:
```
   Unnamed: 0                                    id            timestamp  \
0           0  4220e505-c247-478d-9831-6b9f87a4488a  2022-03-07 12:13:02
1           1  f2612b26-fc82-49ea-8940-0751fdd4d9ef  2022-03-07 16:39:46
2           2  989a287f-67e6-4478-aa49-c3a35dac0e2e  2022-03-01 18:17:43
3           3  af8e5683-d247-46ac-9909-1a77bdebefb2  2022-03-02 14:29:09
4           4  08a32247-3f44-4002-85fb-c198434dd4bb  2022-03-02 13:46:18

                             product_id  estimated_stock_pct
0  f658605e-75f3-4fed-a655-c0903f344427                 0.75
1  de06083a-f5c0-451d-b2f4-9ab88b52609d                 0.48
2  ce8f3a04-d1a4-43b1-a7c2-fa1b8e7674c8                 0.58
3  c21e3ba9-92a3-4745-92c2-6faef73223f7                 0.79
4  7f478817-aa5b-44e9-9059-8045228c9eb0                 0.22
```

[78]:
```
sst.head()
#['id', 'timestamp', 'temperature']
```

[78]:
```
   Unnamed: 0                                    id            timestamp  \
0           0  d1ca1ef8-0eac-42fc-af80-97106efc7b13  2022-03-07 15:55:20
1           1  4b8a66c4-0f3a-4f16-826f-8cf9397e9d18  2022-03-01 09:18:22
2           2  3d47a0c7-1e72-4512-812f-b6b5d8428cf3  2022-03-04 15:12:26
3           3  9500357b-ce15-424a-837a-7677b386f471  2022-03-02 12:30:42
4           4  c4b61fec-99c2-4c6d-8e5d-4edd8c9632fa  2022-03-05 09:09:33

   temperature
0         2.96
1         1.88
2         1.78
3         2.18
4         1.38
```

[79]:
```
m1=pd.merge(sales,ssl,on=['product_id'])
m1.sample(5)
#['id', 'timestamp', 'product_id', 'estimated_stock_pct']
```

[79]:
```
        Unnamed: 0_x                        transaction_id  \
325208          5735  673aab90-301c-427c-9f8a-e19a461a6e73
162367          2556  710b9d6c-2a8e-4c2c-baa6-15217148c61c
33038            279  4183cc2d-0232-4926-9e51-5db399d5897f
281907          4827  d31da7ea-3c3e-49ee-97d5-74abba3342c9
94881           1045  779697cd-eb7e-4a71-acfd-5ce0ad14779d

                timestamp_x                            product_id  \
325208  2022-03-03 10:28:58  693ecb17-5479-474e-8ff9-c2f93de898c0
162367  2022-03-07 18:02:34  81e2fcc6-1e50-4eff-975f-751f18e76444
33038   2022-03-02 16:39:36  7f5e86e6-f06f-45f6-bf44-27b095c9ad1d
281907  2022-03-02 14:11:28  0cc2986e-934c-447e-8f7b-a57b2ceb8693
94881   2022-03-04 13:39:56  bfb96a0d-e52e-4a88-8aba-ecb79e39441d

                category customer_type  unit_price  quantity  total  \
325208       baked goods      standard       10.19         4  40.76
162367  spices and herbs          gold        0.19         3   0.57
33038              fruit    non-member        4.49         4  17.96
281907              meat         basic       16.19         3  48.57
94881         vegetables          gold        1.99         4   7.96

       payment_type  Unnamed: 0_y                                    id  \
325208         cash         10199  f896f049-32ca-4a66-8c39-14232c2610ba
162367     e-wallet          6541  2528753b-3776-40b1-9c56-b624778819a3
33038   credit card          8490  283e66dc-7d8c-4751-9489-df1165ea84cd
281907  credit card          6405  ced06979-0c3d-40b1-8de9-679630d4bc40
94881          cash         14722  739e7db7-21a9-48a8-a7f1-33c05d90f5d3

                timestamp_y  estimated_stock_pct
325208  2022-03-07 14:06:48                 0.34
162367  2022-03-04 09:33:58                 0.09
33038   2022-03-03 09:19:41                 0.07
281907  2022-03-05 17:12:32                 0.66
94881   2022-03-07 15:21:40                 0.89
```

[80]:
```
sst.columns=['Unnamed: 0', 'id', 'timestamp_x', 'temperature']
```

[81]:
```
fdf=pd.merge(m1,sst,on='timestamp_x')
fdf.head()
```

[81]:
```
   Unnamed: 0_x                        transaction_id          timestamp_x  \
0             5  b5b3c8b9-f496-484d-aa30-4f2efb5ed56c  2022-03-07 17:59:47
1             5  b5b3c8b9-f496-484d-aa30-4f2efb5ed56c  2022-03-07 17:59:47
2             5  b5b3c8b9-f496-484d-aa30-4f2efb5ed56c  2022-03-07 17:59:47
3             5  b5b3c8b9-f496-484d-aa30-4f2efb5ed56c  2022-03-07 17:59:47
4             5  b5b3c8b9-f496-484d-aa30-4f2efb5ed56c  2022-03-07 17:59:47
```

```
                             product_id category customer_type  unit_price  \
0  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit      standard        3.99
1  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit      standard        3.99
2  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit      standard        3.99
3  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit      standard        3.99
4  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit      standard        3.99

   quantity  total payment_type  Unnamed: 0_y  \
0         4  15.96         cash           164
1         4  15.96         cash           172
2         4  15.96         cash           242
3         4  15.96         cash           343
4         4  15.96         cash           465

                                  id_x          timestamp_y  \
0  5540da96-885e-4d17-aa33-5720ca9b57d5  2022-03-03 09:16:02
1  e65514e8-2935-4921-baf8-2a9cf849a7f2  2022-03-04 17:52:19
2  1faae309-bd2c-412c-b988-42e3086e2e86  2022-03-07 12:44:46
3  d383beb7-6dd0-46b3-aaa3-a1141e0c94dd  2022-03-02 17:36:38
4  9c0d4cbe-c00d-42c2-8489-263188ae0b17  2022-03-07 09:44:25

   estimated_stock_pct  Unnamed: 0                                  id_y  \
0                 0.25       15854  38808b9d-712c-4703-a5ae-920817960fa6
1                 0.23       15854  38808b9d-712c-4703-a5ae-920817960fa6
2                 0.80       15854  38808b9d-712c-4703-a5ae-920817960fa6
3                 0.79       15854  38808b9d-712c-4703-a5ae-920817960fa6
4                 0.86       15854  38808b9d-712c-4703-a5ae-920817960fa6

   temperature
0        -2.84
1        -2.84
2        -2.84
3        -2.84
4        -2.84
```

[82]:
```
dr=['transaction_id','Unnamed: 0_x', 'id_x','Unnamed: 0_y',␣
 ↪'id_y','timestamp_y','Unnamed: 0']
fdf.drop(columns=dr,inplace=True,axis=1)
```

[83]:
```
fdf.head()
```

[83]:
```
            timestamp_x                             product_id category  \
0  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
1  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
2  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
3  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
4  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
```

```
  customer_type  unit_price  quantity  total payment_type  \
0      standard        3.99         4  15.96         cash
1      standard        3.99         4  15.96         cash
2      standard        3.99         4  15.96         cash
3      standard        3.99         4  15.96         cash
4      standard        3.99         4  15.96         cash

   estimated_stock_pct  temperature
0                 0.25        -2.84
1                 0.23        -2.84
2                 0.80        -2.84
3                 0.79        -2.84
4                 0.86        -2.84
```

[84]:
```
fdf.columns=['timestamp', 'prd_id', 'category', 'customer_type', 'unit_price',
       'quantity', 'total', 'payment_type', 'avg_stk_prc',
       'temperature']
```

[85]:
```
fdf.head()
```

[85]:
```
             timestamp                                prd_id category  \
0  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
1  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
2  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
3  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit
4  2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713    fruit

  customer_type  unit_price  quantity  total payment_type  avg_stk_prc  \
0      standard        3.99         4  15.96         cash         0.25
1      standard        3.99         4  15.96         cash         0.23
2      standard        3.99         4  15.96         cash         0.80
3      standard        3.99         4  15.96         cash         0.79
4      standard        3.99         4  15.96         cash         0.86

   temperature
0        -2.84
1        -2.84
2        -2.84
3        -2.84
4        -2.84
```

3 DATA PREPROCESSING

[86]:
```
fdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38942 entries, 0 to 38941
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   timestamp      38942 non-null  object
 1   prd_id         38942 non-null  object
 2   category       38942 non-null  object
 3   customer_type  38942 non-null  object
 4   unit_price     38942 non-null  float64
 5   quantity       38942 non-null  int64
 6   total          38942 non-null  float64
 7   payment_type   38942 non-null  object
 8   avg_stk_prc    38942 non-null  float64
 9   temperature    38942 non-null  float64
dtypes: float64(4), int64(1), object(5)
memory usage: 3.0+ MB
```

[87]:
```
fdf['timestamp']=pd.to_datetime(fdf['timestamp'])
fdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38942 entries, 0 to 38941
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   timestamp      38942 non-null  datetime64[ns]
 1   prd_id         38942 non-null  object
 2   category       38942 non-null  object
 3   customer_type  38942 non-null  object
 4   unit_price     38942 non-null  float64
 5   quantity       38942 non-null  int64
 6   total          38942 non-null  float64
 7   payment_type   38942 non-null  object
 8   avg_stk_prc    38942 non-null  float64
 9   temperature    38942 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(1), object(4)
memory usage: 3.0+ MB
```

[88]:
```
fdf['week']=fdf['timestamp'].dt.dayofweek
fdf['day']=fdf['timestamp'].dt.day
fdf['hour']=fdf['timestamp'].dt.hour
```

[89]:
```
sum(fdf.duplicated())
```

[89]: 9237

[90]:
```
dups=fdf.duplicated()
ndf=fdf[~dups]
ndf
```

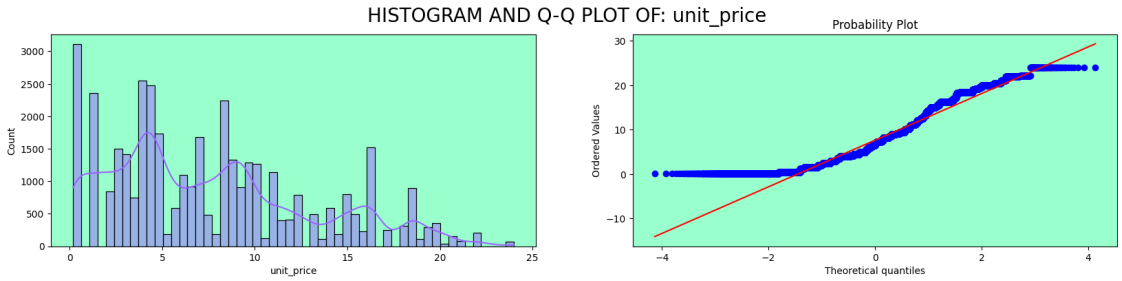
[90]:

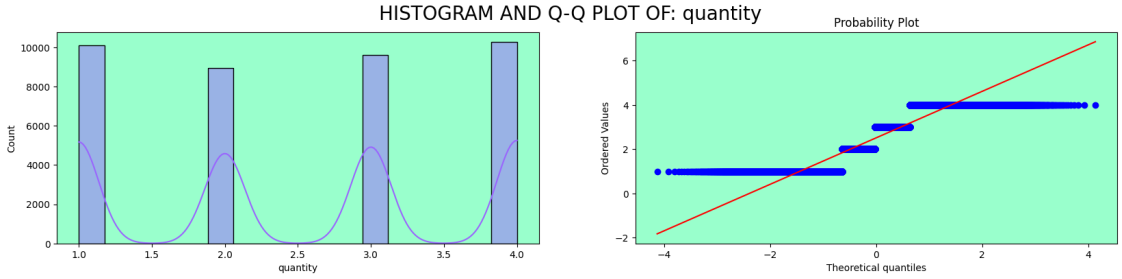
```
                timestamp                                prd_id  \
0     2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713
1     2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713
2     2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713
3     2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713
4     2022-03-07 17:59:47  3bc6c1ea-0198-46de-9ffd-514ae3338713
…                     …                                    …
38936 2022-03-07 11:08:47  bc6187a9-d508-482b-9ca6-590d1cc7524f
38938 2022-03-07 11:08:47  bc6187a9-d508-482b-9ca6-590d1cc7524f
38939 2022-03-07 11:08:47  bc6187a9-d508-482b-9ca6-590d1cc7524f
38940 2022-03-07 11:08:47  bc6187a9-d508-482b-9ca6-590d1cc7524f
38941 2022-03-07 11:08:47  bc6187a9-d508-482b-9ca6-590d1cc7524f

                category customer_type  unit_price  quantity  total  \
0                   fruit      standard        3.99         4  15.96
1                   fruit      standard        3.99         4  15.96
2                   fruit      standard        3.99         4  15.96
3                   fruit      standard        3.99         4  15.96
4                   fruit      standard        3.99         4  15.96
…                       …             …         …        …    …
38936  cleaning products         basic       14.19         2  28.38
38938  cleaning products         basic       14.19         2  28.38
38939  cleaning products         basic       14.19         2  28.38
38940  cleaning products         basic       14.19         2  28.38
38941  cleaning products         basic       14.19         2  28.38

      payment_type  avg_stk_prc  temperature  week  day  hour
0             cash         0.25        -2.84     0    7    17
1             cash         0.23        -2.84     0    7    17
2             cash         0.80        -2.84     0    7    17
3             cash         0.79        -2.84     0    7    17
4             cash         0.86        -2.84     0    7    17
…              …            …            …    …  …   …
38936   debit card         0.68        -1.33     0    7    11
38938   debit card         0.67        -1.33     0    7    11
38939   debit card         0.92        -1.33     0    7    11
38940   debit card         0.21        -1.33     0    7    11
38941   debit card         0.14        -1.33     0    7    11

[29705 rows x 13 columns]
```

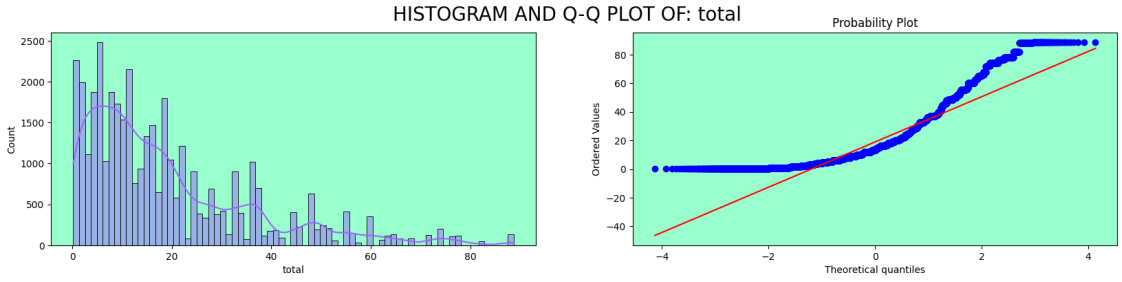
[91]:
```
from scipy.stats import probplot
```

[92]:
```
ifc=['unit_price', 'quantity', 'total', 'avg_stk_prc', 'temperature', 'week',
       'day', 'hour']


def histandbox(a, b):
    fig, axs = plt.subplots(1, 2, figsize=(20, 4))
    sns.histplot(data=b, x=a, ax=axs[0], kde=True, color='#9966ff')
    probplot(b[a],plot=axs[1])
    fig.suptitle(f"HISTOGRAM AND Q-Q PLOT OF: {a}", fontsize=20)
    plt.show()
```

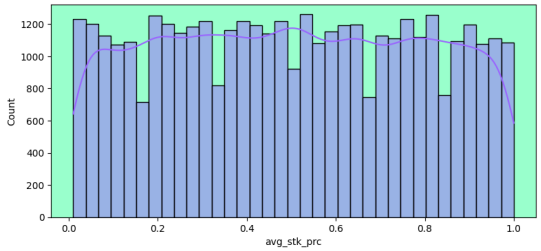
[93]:
```
for i in ifc:
    histandbox(i,fdf)
```

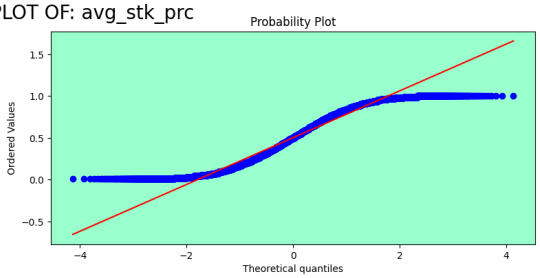




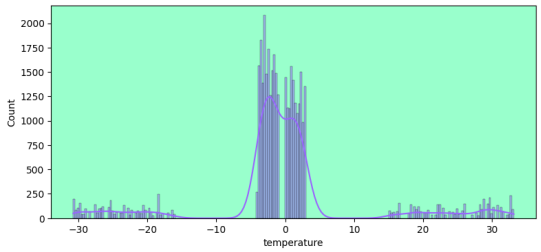


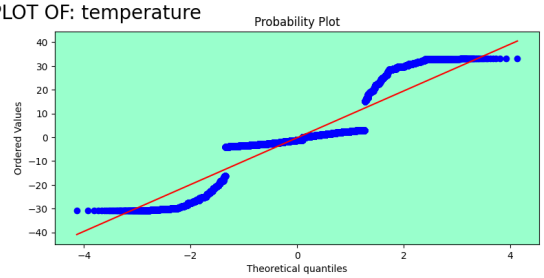
HISTOGRAM AND Q-Q PLOT OF: avg_stk_prc



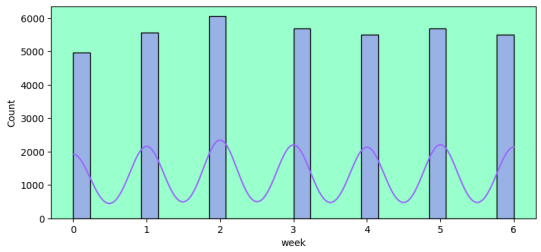


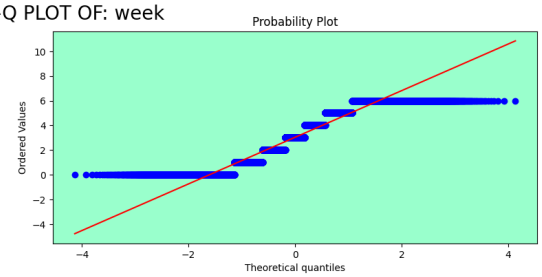
HISTOGRAM AND Q-Q PLOT OF: temperature



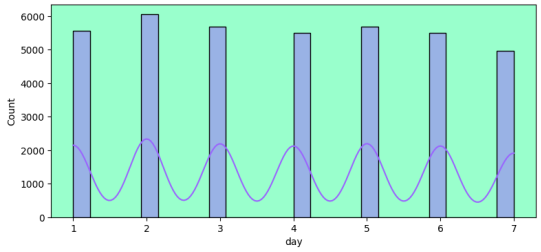


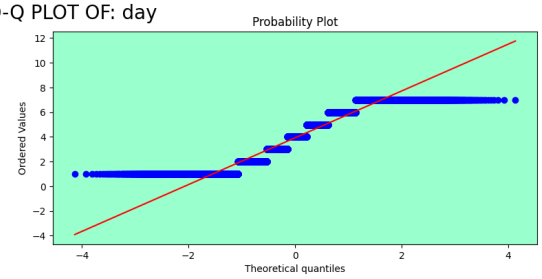
HISTOGRAM AND Q-Q PLOT OF: week

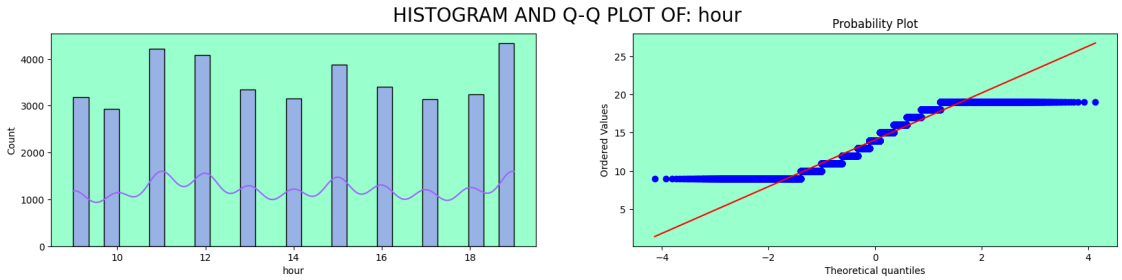




HISTOGRAM AND Q-Q PLOT OF: day







4 OUTLIER DETECTION AND REMOVAL OF INTER-QUARTILE RANGE METHOD

coefficient calculation towards sigma value

- = Q1 - 1.722 * IQR
- = Q1 - 1.722 * (Q3 - Q1)
- = -0.675 - 1.722 * (0.675 - [-0.675])
- = -0.675 - 1.722* 1.35
- = -2.99
- ~=3sigma

[94]:
```
#outliers
c=fdf.select_dtypes(include=['int','float']).columns
q1=fdf[c].quantile(.25)
q3=fdf[c].quantile(.75)
iqr=q3-q1
ll=q1-1.722*iqr
ul=q3+1.722*iqr
mask=fdf[~(fdf[c]>ul)|(fdf[c]<ll)].all(axis=1)
cdf=fdf[mask]
cdf.head(3)
```

[94]:

	timestamp	prd_id	category	customer_type	unit_price	quantity	total	payment_type	avg_stk_prc	temperature	week	day	hour
69	2022-03-01 14:58:07	3bc6c1ea-0198-46de-9ffd-514ae3338713	fruit	premium	3.99	4	15.96	e-wallet	0.25	29.73	1	1	14
70	2022-03-01 14:58:07	3bc6c1ea-0198-46de-9ffd-514ae3338713	fruit	premium	3.99	4	15.96	e-wallet	0.23	29.73	1	1	14
71	2022-03-01 14:58:07	3bc6c1ea-0198-46de-9ffd-514ae3338713	fruit	premium	3.99	4	15.96	e-wallet	0.80				

```
71         29.73      1     1     14
```

[95]:
```
cdf.drop(columns=['timestamp', 'prd_id'],axis=1,inplace=True)
```

[96]:
```
cdf.select_dtypes(include=['int','float']).columns
```

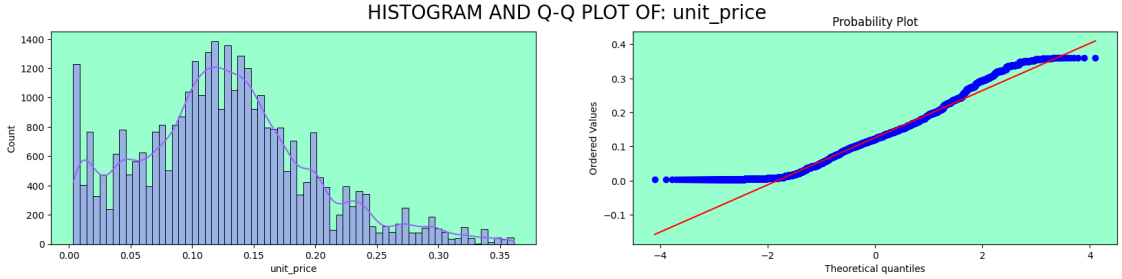
[96]:
```
Index(['unit_price', 'quantity', 'total', 'avg_stk_prc', 'temperature', 'week',
       'day', 'hour'],
      dtype='object')
```

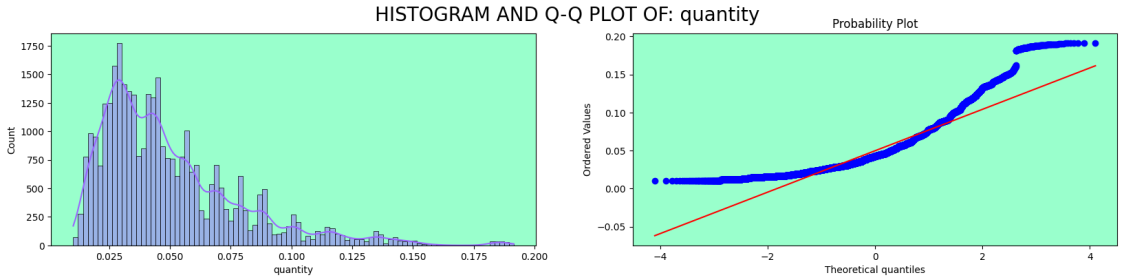
[97]:
```
from sklearn.preprocessing import PowerTransformer,QuantileTransformer

f=['unit_price', 'quantity', 'total', 'avg_stk_prc', 'temperature', 'week',
       'day', 'hour']
```

5 DISTRIBUTION – FEATURE TRANSOFORMATION

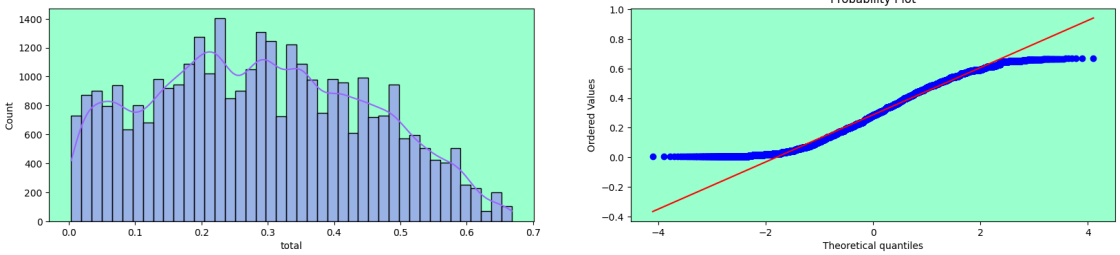
[98]:
```
from sklearn.preprocessing import MaxAbsScaler,Normalizer
```

[99]:
```
n=Normalizer(norm='l1',)
cdf[f]=n.fit_transform(cdf[f])

for i in f:
    histandbox(i,cdf)
```

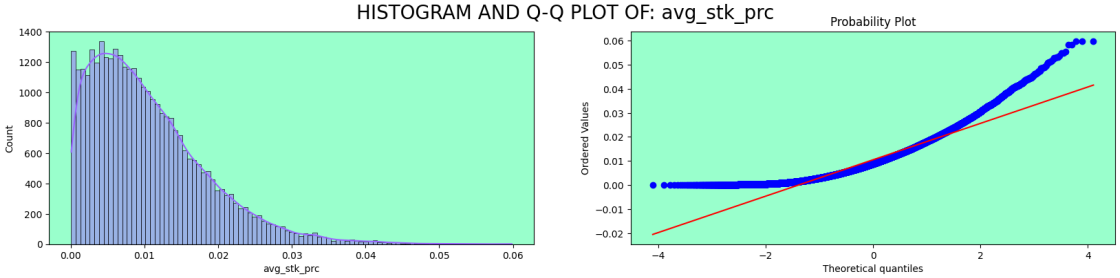




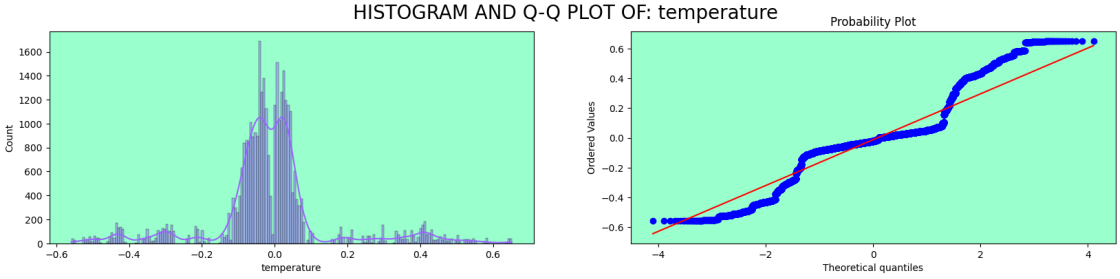
HISTOGRAM AND Q-Q PLOT OF: total



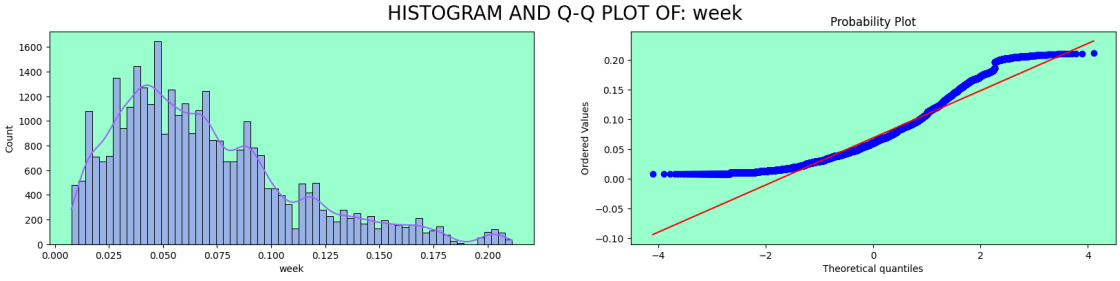
HISTOGRAM AND Q-Q PLOT OF: avg_stk_prc

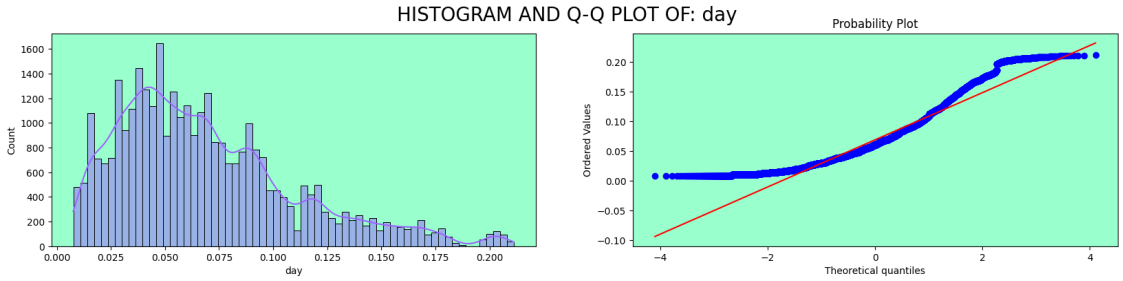


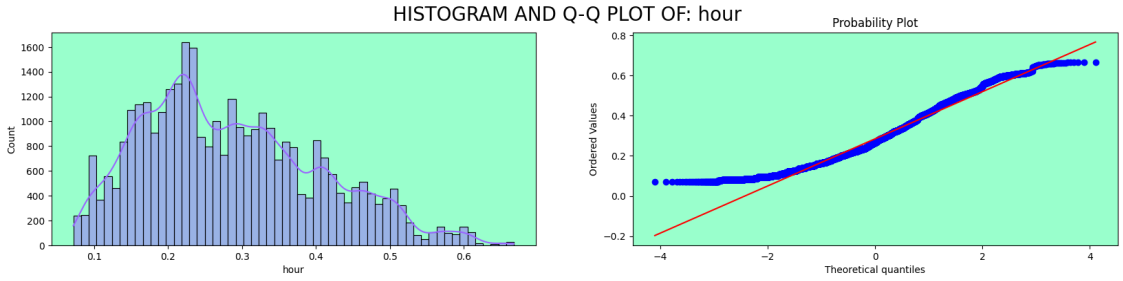
HISTOGRAM AND Q-Q PLOT OF: temperature



HISTOGRAM AND Q-Q PLOT OF: week







I have tried the l2 normalization and max normalization which does not yielded expected result. BUt L1 normalization worked really well for the data. But still some are too skewed and have long tailed distributions.

[100]:
```
cat=cdf.select_dtypes(include='object').columns
cat
```

[100]:
```
Index(['category', 'customer_type', 'payment_type'], dtype='object')
```

[101]:
```
for i in cat:
    print("**************************")
    print(f'Nuniques of :         {i}',cdf[i].nunique())
    print()
    print(cdf[i].value_counts().sort_values(ascending=False))
```

```
**************************
Nuniques of :         category 22

category
fruit                   6081
vegetables              4287
dairy                   2503
meat                    2066
canned foods            2016
refrigerated items      1873
packaged foods          1480
```

```
kitchen                  1372
seafood                  1305
cheese                   1195
baked goods              1165
baby products            1156
medicine                 1132
pets                     1001
beverages                 915
cleaning products         790
personal care             772
baking                    748
snacks                    710
frozen                    677
condiments and sauces     388
spices and herbs          292
Name: count, dtype: int64
**************************
Nuniques of :         customer_type 5

customer_type
non-member    7801
premium       7273
gold          7046
basic         6000
standard      5804
Name: count, dtype: int64
**************************
Nuniques of :         payment_type 4

payment_type
cash           9112
credit card    8787
e-wallet       8640
debit card     7385
Name: count, dtype: int64
```

6 CATEGORICAL FEATURES ENCODING: LABELENCODER & ONEHOTENCODER

[102]:
```
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
le=LabelEncoder()
cdf['category']=le.fit_transform(cdf['category'])
```

[103]:
```
for i in ['customer_type','payment_type']:
    cdf[i]=cdf[i].astype('category')
```

[104]:
```
cdf=pd.get_dummies(cdf,columns=['customer_type','payment_type'])
cdf
```

[104]:

	category	unit_price	quantity	total	avg_stk_prc	temperature	\
69	10	0.057057	0.057200	0.228228	0.003575	0.425139	
70	10	0.057073	0.057216	0.228294	0.003290	0.425261	
71	10	0.056612	0.056754	0.226447	0.011351	0.421822	
72	10	0.056620	0.056762	0.226479	0.011210	0.421882	
73	10	0.056564	0.056705	0.226255	0.012192	0.421463	
…	…	…	…	…	…	…	
38891	6	0.224350	0.026410	0.448699	0.004490	-0.031956	
38892	6	0.224379	0.026413	0.448759	0.004358	-0.031960	
38893	6	0.224320	0.026406	0.448640	0.004621	-0.031951	
38894	6	0.224587	0.026438	0.449174	0.003437	-0.031989	
38895	6	0.222586	0.026202	0.445172	0.012315	-0.031704	

	week	day	hour	customer_type_basic	customer_type_gold	\
69	0.014300	0.014300	0.200200	False	False	
70	0.014304	0.014304	0.200257	False	False	
71	0.014188	0.014188	0.198638	False	False	
72	0.014190	0.014190	0.198666	False	False	
73	0.014176	0.014176	0.198469	False	False	
…	…	…	…	…	…	
38891	0.052819	0.052819	0.158458	False	False	
38892	0.052826	0.052826	0.158479	False	False	
38893	0.052812	0.052812	0.158437	False	False	
38894	0.052875	0.052875	0.158625	False	False	
38895	0.052404	0.052404	0.157212	False	False	

	customer_type_non-member	customer_type_premium	\
69	False	True	
70	False	True	
71	False	True	
72	False	True	
73	False	True	
…	…	…	
38891	False	True	
38892	False	True	
38893	False	True	
38894	False	True	
38895	False	True	

	customer_type_standard	payment_type_cash	payment_type_credit card	\
69	False	False	False	
70	False	False	False	
71	False	False	False	
72	False	False	False	

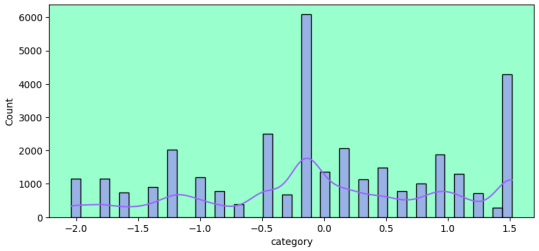
```
73                    False              False                    False
…                     …                  …                        …
38891                 False              True                     False
38892                 False              True                     False
38893                 False              True                     False
38894                 False              True                     False
38895                 False              True                     False

      payment_type_debit card  payment_type_e-wallet
69                      False                   True
70                      False                   True
71                      False                   True
72                      False                   True
73                      False                   True
…                       …                       …
38891                   False                  False
38892                   False                  False
38893                   False                  False
38894                   False                  False
38895                   False                  False

[33924 rows x 18 columns]
```

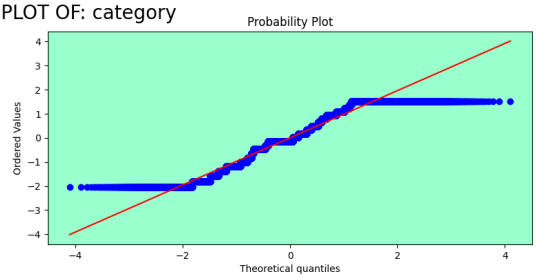
[105]:
```
cdf.columns
```

[105]:
```
Index(['category', 'unit_price', 'quantity', 'total', 'avg_stk_prc',
       'temperature', 'week', 'day', 'hour', 'customer_type_basic',
       'customer_type_gold', 'customer_type_non-member',
       'customer_type_premium', 'customer_type_standard', 'payment_type_cash',
       'payment_type_credit card', 'payment_type_debit card',
       'payment_type_e-wallet'],
      dtype='object')
```

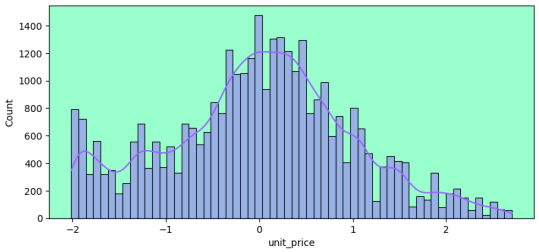
[106]:
```
cols=['category', 'unit_price', 'quantity', 'total', 'avg_stk_prc',
       'temperature', 'week', 'day', 'hour', 'customer_type_basic',
       'customer_type_gold', 'customer_type_non-member',
       'customer_type_premium', 'customer_type_standard', 'payment_type_cash',
       'payment_type_credit card', 'payment_type_debit card',
       'payment_type_e-wallet']
ss=PowerTransformer()
cdf[cols]=ss.fit_transform(cdf[cols])

for i in cols:
    histandbox(i,cdf)
```

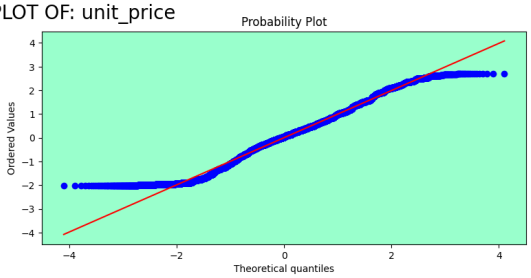
HISTOGRAM AND Q-Q PLOT OF: category



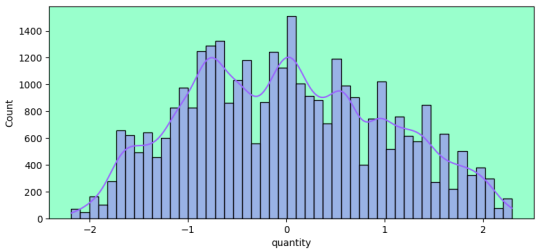


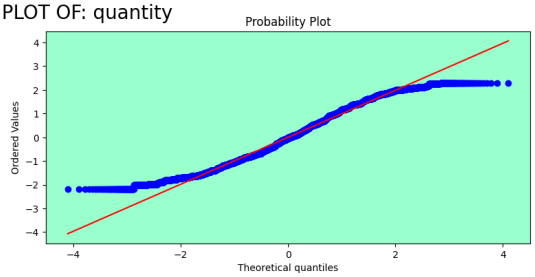
HISTOGRAM AND Q-Q PLOT OF: unit_price



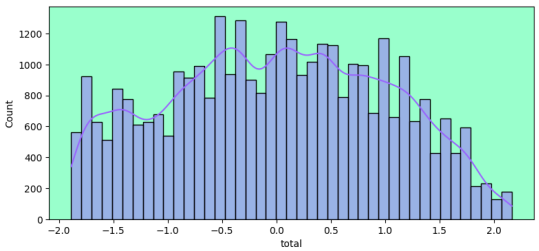


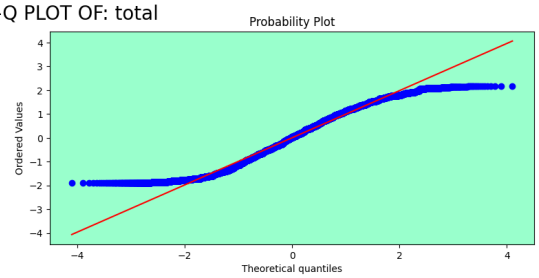
HISTOGRAM AND Q-Q PLOT OF: quantity



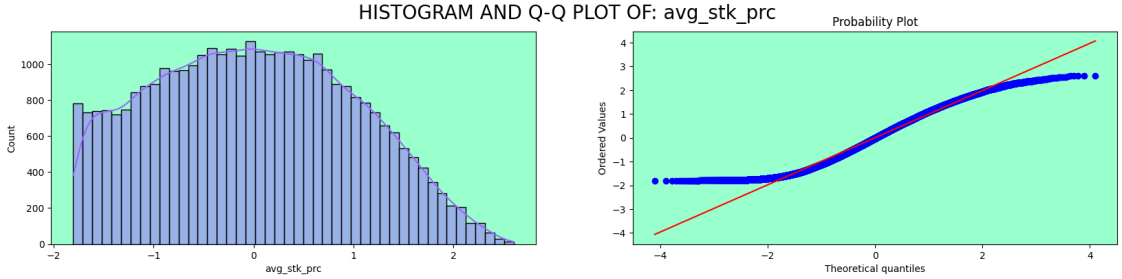


HISTOGRAM AND Q-Q PLOT OF: total

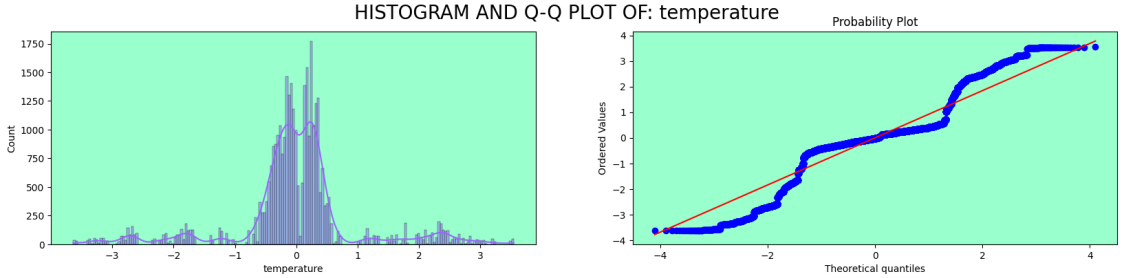




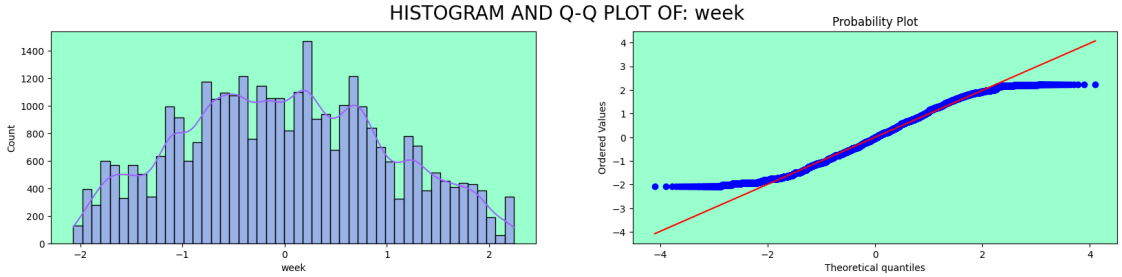
HISTOGRAM AND Q-Q PLOT OF: avg_stk_prc



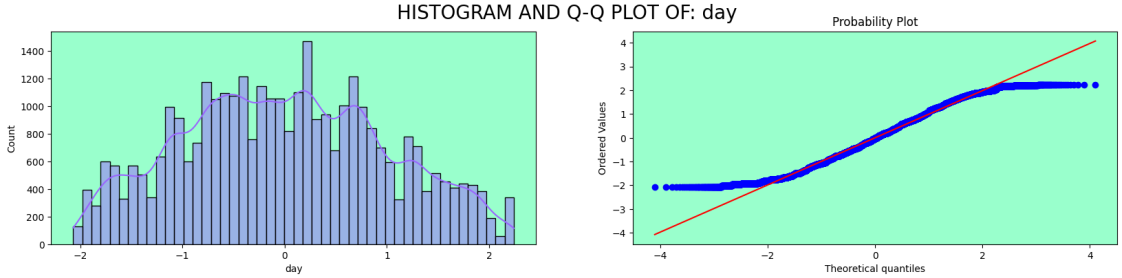
HISTOGRAM AND Q-Q PLOT OF: temperature



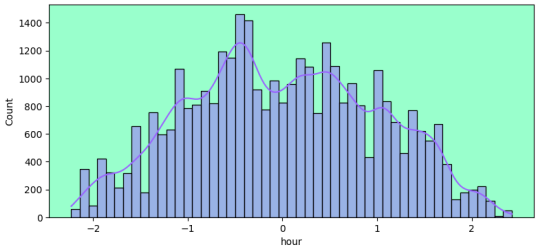
HISTOGRAM AND Q-Q PLOT OF: week

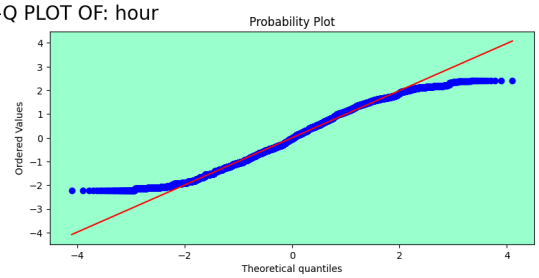


HISTOGRAM AND Q-Q PLOT OF: day

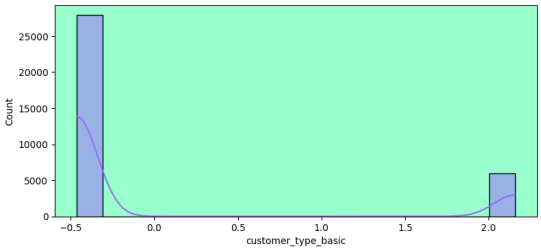


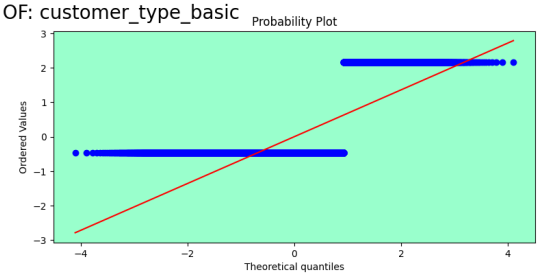
HISTOGRAM AND Q-Q PLOT OF: hour



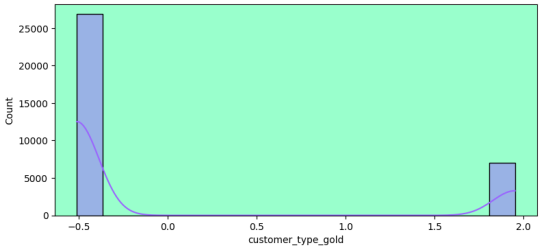


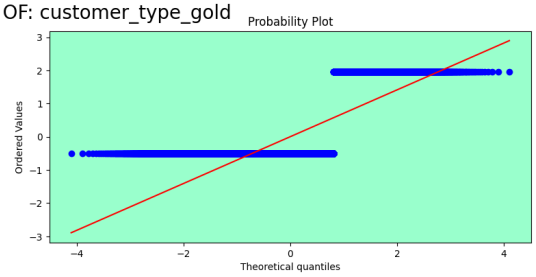
HISTOGRAM AND Q-Q PLOT OF: customer_type_basic



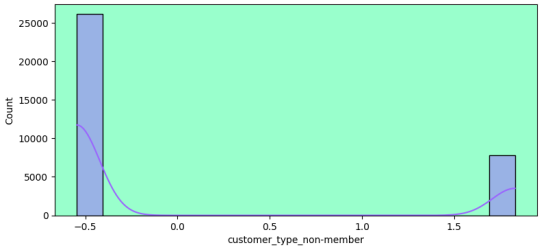


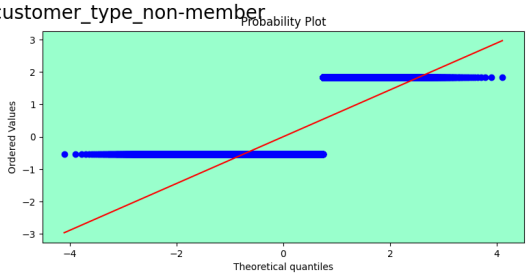
HISTOGRAM AND Q-Q PLOT OF: customer_type_gold



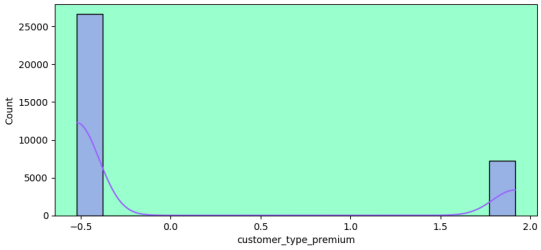


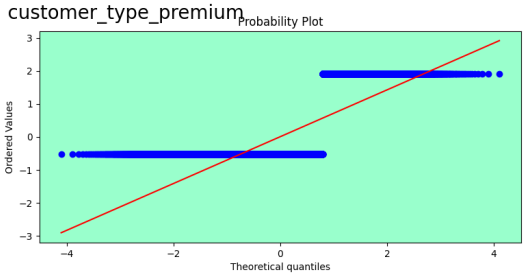
HISTOGRAM AND Q-Q PLOT OF: customer_type_non-member



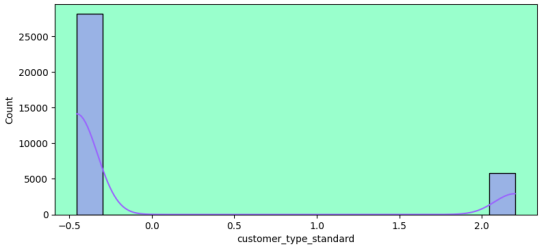


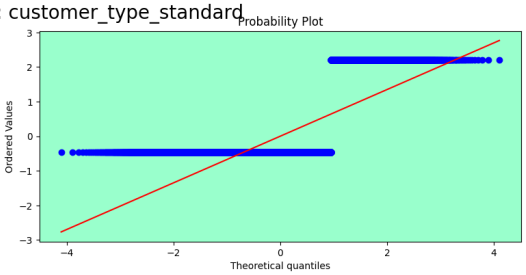
HISTOGRAM AND Q-Q PLOT OF: customer_type_premium



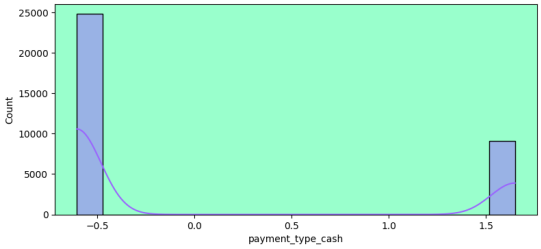


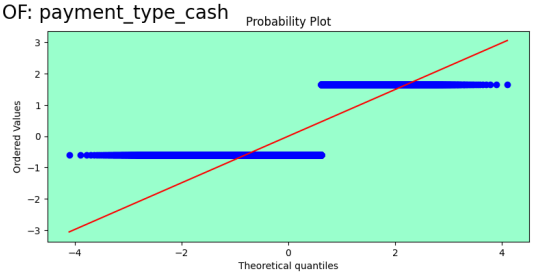
HISTOGRAM AND Q-Q PLOT OF: customer_type_standard



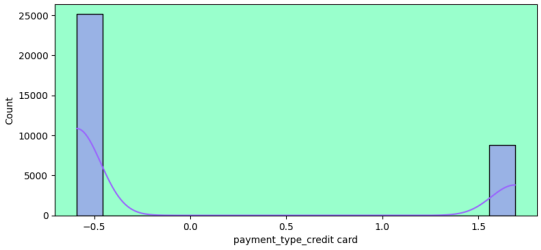


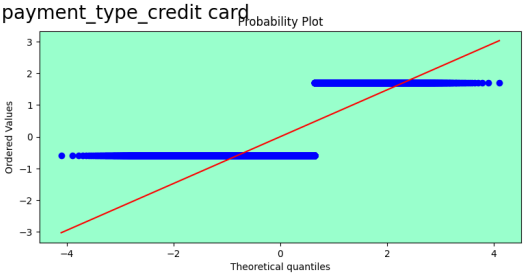
HISTOGRAM AND Q-Q PLOT OF: payment_type_cash

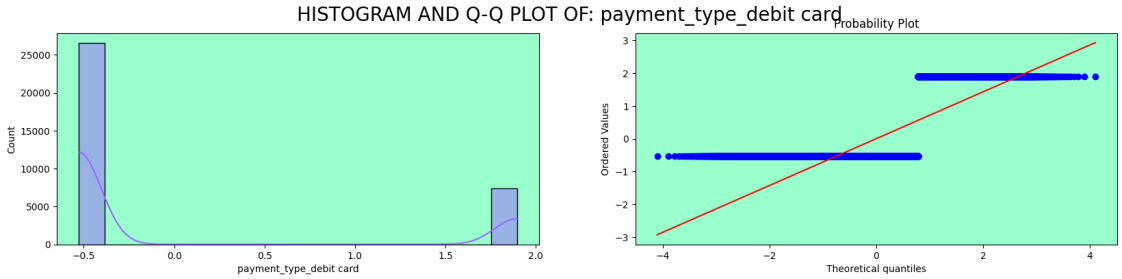


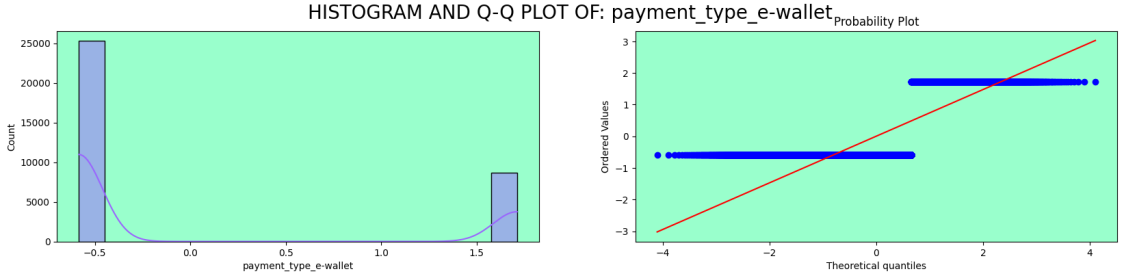


HISTOGRAM AND Q-Q PLOT OF: payment_type_credit card









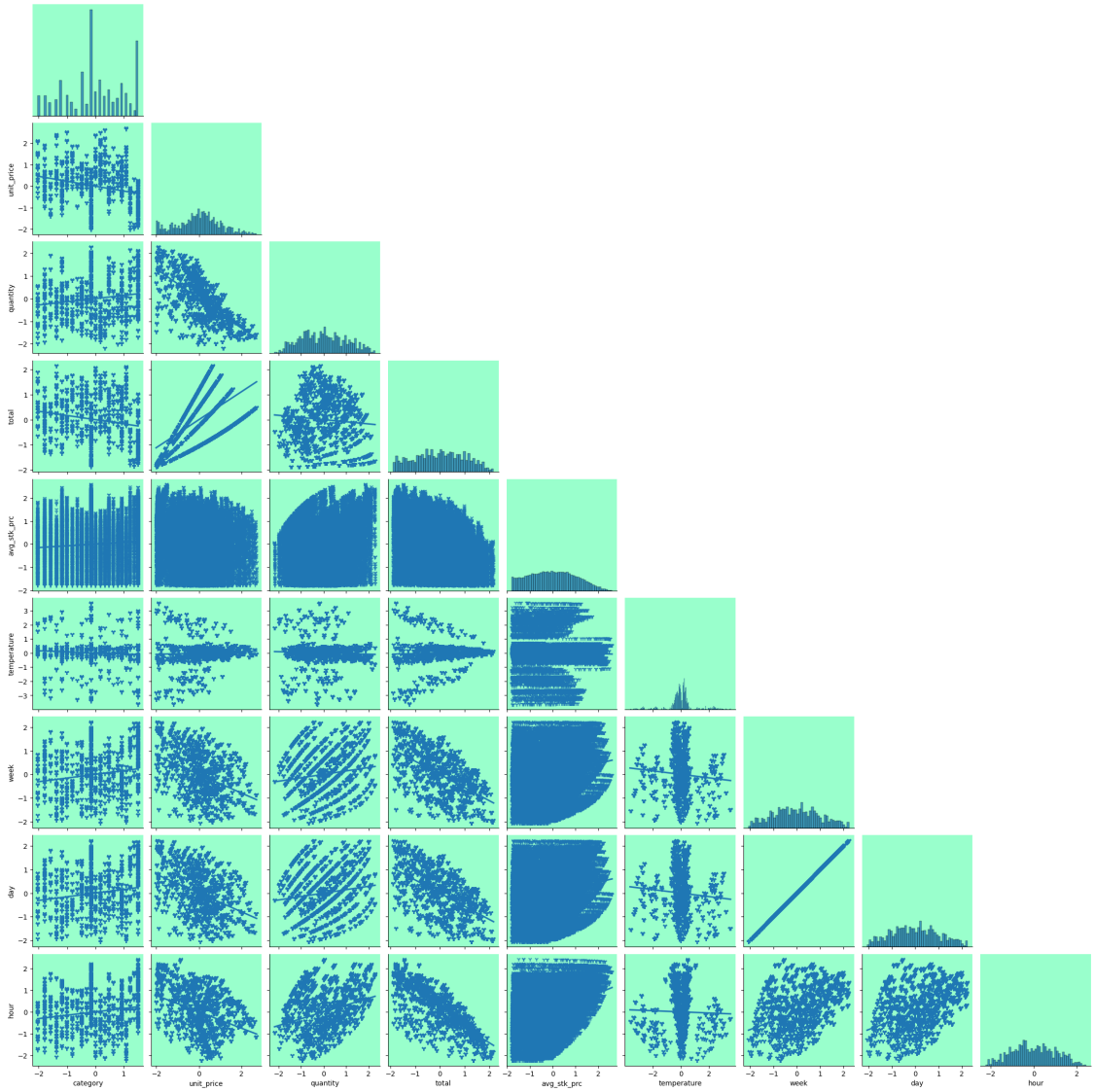
Most of the features got transformed into approximately Normal Distributio using Yeo-Johnson Transformation by powertransformer

[ ]:
```
cc=['customer_type_basic',
       'customer_type_gold', 'customer_type_non-member',
       'customer_type_premium', 'customer_type_standard', 'payment_type_cash',
       'payment_type_credit card', 'payment_type_debit card',
       'payment_type_e-wallet']
```

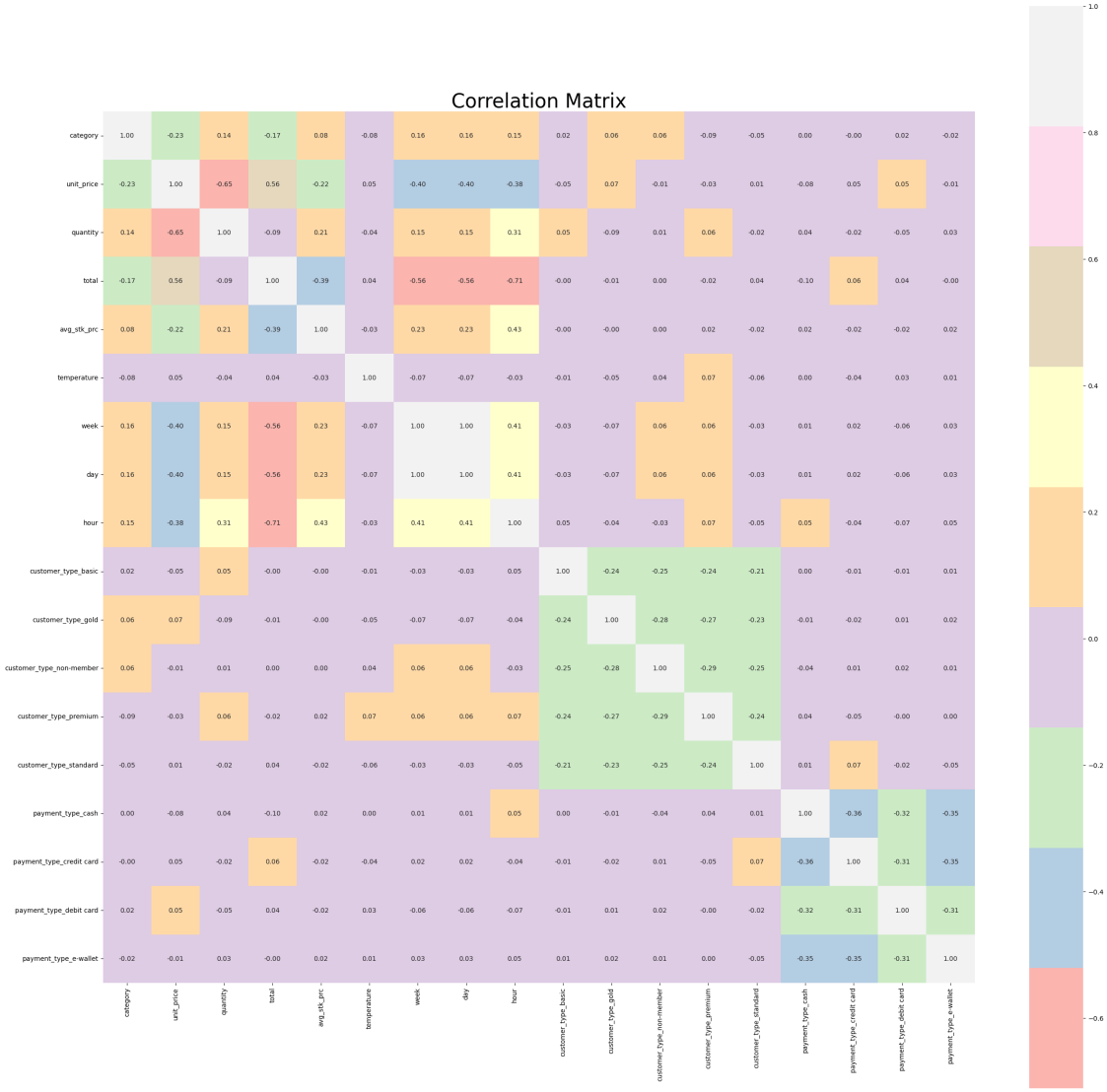
[123]:
```
a=sns.pairplot(cdf,palette='#FFFACD',kind='reg',corner=True,markers="1")
a.fig.suptitle('Pairplot', y=1.08,fontsize=30)
```

[123]:
```
Text(0.5, 1.08, 'Pairplot')
```

Pairplot



[115]:
```
plt.figure(figsize=(30,30))
sns.heatmap(cdf.corr(method='pearson'),annot=True,fmt='.
  ↪2f',cmap='Pastel1',square=True)
plt.title('Correlation Matrix',fontsize=30)
plt.show()
```



There is lot of very low-inter correlation between the independent variables and few have good amount of correlation with the independent variables

[108]:
```
for i in cc:
    cdf[cc]=cdf[cc].astype('category')
```

[36]:
```
cdf[cc].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33924 entries, 69 to 38895
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   customer_type_basic     33924 non-null  category
```

```
 1   customer_type_gold         33924 non-null  category
 2   customer_type_non-member   33924 non-null  category
 3   customer_type_premium      33924 non-null  category
 4   customer_type_standard     33924 non-null  category
 5   payment_type_cash          33924 non-null  category
 6   payment_type_credit card   33924 non-null  category
 7   payment_type_debit card    33924 non-null  category
 8   payment_type_e-wallet      33924 non-null  category
dtypes: category(9)
memory usage: 564.3 KB
```

7 FEATURE IMPORTANCE: MUTUAL INFORMATION & F-STATISTIC for REGRESSION

[37]:
```
y=cdf.avg_stk_prc
x=cdf.drop(columns='avg_stk_prc')
```

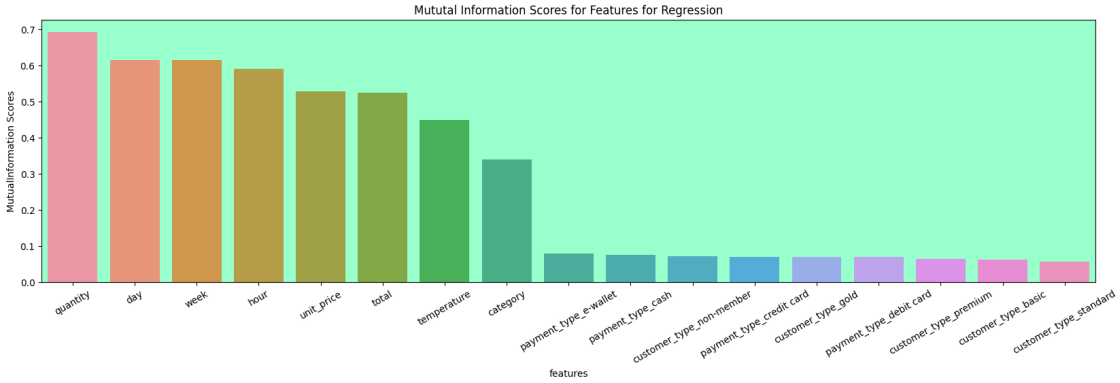
[229]:
```
from sklearn.feature_selection import mutual_info_regression
d=['cus_gold', 'cus_non-member',
       'cus_premium', 'cus_standard', 'pymode_credit card',
       'pymode_debit card', 'pymode_e-wallet']
mi=mutual_info_regression(x,y,discrete_features='auto')
```

[230]:
```
miscores=mi.tolist()
ms=pd.DataFrame({'features':x.columns,'MutualInformation Scores':miscores})
ms
```

[230]:

	features	MutualInformation Scores
0	category	0.338898
1	unit_price	0.528565
2	quantity	0.693032
3	total	0.523817
4	temperature	0.448207
5	week	0.615157
6	day	0.615265
7	hour	0.591473
8	customer_type_basic	0.061260
9	customer_type_gold	0.068757
10	customer_type_non-member	0.071831
11	customer_type_premium	0.064185
12	customer_type_standard	0.055076
13	payment_type_cash	0.075531
14	payment_type_credit card	0.069140
15	payment_type_debit card	0.068745
16	payment_type_e-wallet	0.077961

[231]:
```
ms=ms.sort_values(by='MutualInformation Scores',ascending=False)
plt.figure(figsize=(20,5))
sns.barplot(data=ms,x='features',y='MutualInformation Scores')
plt.xticks(rotation=30)
plt.title('Mututal Information Scores for Features for Regression')
plt.show()
```

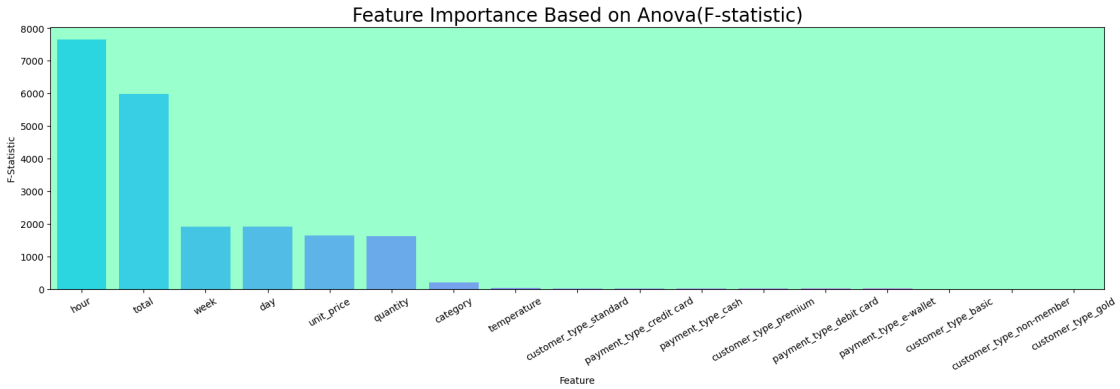


[232]:
```
from sklearn.feature_selection import f_regression
f,p=f_regression(x,y)
anova=pd.DataFrame({'Feature':x.columns,
                    'F-Statistic':f,
                    'P value':p})
anova=anova.sort_values(by='F-Statistic',ascending=False)
anova.round(2)
```

[232]:

	Feature	F-Statistic	P value
7	hour	7653.04	0.00
3	total	5970.87	0.00
5	week	1915.42	0.00
6	day	1915.42	0.00
1	unit_price	1649.66	0.00
2	quantity	1620.00	0.00
0	category	196.28	0.00
4	temperature	30.03	0.00
12	customer_type_standard	19.75	0.00
14	payment_type_credit card	19.55	0.00
13	payment_type_cash	19.18	0.00
11	customer_type_premium	18.99	0.00
15	payment_type_debit card	11.29	0.00
16	payment_type_e-wallet	10.07	0.00
8	customer_type_basic	0.20	0.66
10	customer_type_non-member	0.18	0.68
9	customer_type_gold	0.09	0.76

[233]:
```
plt.figure(figsize=(20,5))
sns.barplot(data=anova,x='Feature',y='F-Statistic',palette='cool')
plt.xticks(rotation=30)
plt.title('Feature Importance Based on Anova(F-statistic)',fontsize=20)
plt.show()
```



8 TRAIN-TEST SPLIT

[88]:
```
from sklearn.model_selection import train_test_split
```

[89]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,shuffle=True,train_size=.80)
print(f"xtrain: {xtrain.shape} and xtest: {xtest.shape}")
print(f"ytrain: {ytrain.shape} and ytest: {ytest.shape}")
```

```
xtrain: (27139, 17) and xtest: (6785, 17)
ytrain: (27139,) and ytest: (6785,)
```

9 PREDICTIVE MODEL BUILDING AND VALIDATION

[90]:
```
from sklearn.metrics import␣
↪mean_squared_error,mean_absolute_error,r2_score,median_absolute_error
from sklearn.model_selection import KFold,cross_val_score,RepeatedKFold
from sklearn.linear_model import LinearRegression
import pickle as pk
```

[237]:
```
from sklearn.model_selection import RandomizedSearchCV
```

[238]:
```
def p(g,model,name):
    print(model)
    print('')
  ␣
↪pt=RandomizedSearchCV(estimator=model,param_distributions=g,n_jobs=-1,cv=7,random_state=344
```

```
    pt.fit(xtrain,ytrain)
    print('Best Parameters',pt.best_params_)
    print('Best Score',round(pt.best_score_*100,2),"%")
    print('')
    best=pt.best_estimator_

    best.fit(xtrain,ytrain)
    ypred=best.predict(xtest)

    pk.dump(best, open(f'"{name}.pkl"', 'wb'))

    #Dataframe
    b={'Metric_Name':['R.
↪sq(%)','Mean-Squared-Error','Rooted-Mean-Squared-Error','Mean-Absoulute-Error','Median-Abso
      'Values':
↪[round(r2_score(ytest,ypred)*100,2),round(mean_squared_error(ytest,ypred),3),
                round(np.
↪sqrt(mean_absolute_error(ytest,ypred)),3),round(mean_absolute_error(ytest,ypred),3),
                round(median_absolute_error(ytest,ypred),3)]}

    d=pd.DataFrame(b)
    return d
```

10 * LINEAR REGRESSION *

[47]:
```
lr=LinearRegression(n_jobs=-1,)
g={'positive':[True,False],'fit_intercept':[True,False]
  }
p(g,lr,'linearreg')
```

```
LinearRegression(n_jobs=-1)

Best Parameters {'positive': False, 'fit_intercept': False}
Best Score 22.65 %
```

[47]:

	Metric_Name	Values
0	R.sq(%)	23.110
1	Mean-Squared-Error	0.757
2	Rooted-Mean-Squared-Error	0.850
3	Mean-Absoulute-Error	0.723
4	Median-Absoulte-Error	0.666

11 * HIST-GRADIENT-BOOSTING-REGRESSOR *

[48]:
```
from sklearn.ensemble import AdaBoostRegressor,HistGradientBoostingRegressor
from sklearn.tree import DecisionTreeRegressor
import warnings
warnings.filterwarnings('ignore')
```

[49]:
```
h=HistGradientBoostingRegressor(random_state=34563,max_bins=244,max_depth=30)
g={'learning_rate':[0.1,0.01],
  'max_iter':[100,200,500,600,800,900],
  'max_leaf_nodes':[20,30],
 'l2_regularization':[1,0.01],'tol':[1e-7,1e-8]}
p(g,h,'histgrdbstreg')
```

```
HistGradientBoostingRegressor(max_bins=244, max_depth=30, random_state=34563)

Best Parameters {'tol': 1e-08, 'max_leaf_nodes': 30, 'max_iter': 800,
'learning_rate': 0.1, 'l2_regularization': 1}
Best Score 94.08 %
```

[49]:

	Metric_Name	Values
0	R.sq(%)	94.810
1	Mean-Squared-Error	0.051
2	Rooted-Mean-Squared-Error	0.408
3	Mean-Absoulute-Error	0.166
4	Median-Absoulte-Error	0.126

12 * ADAPTIVE-BOOSTING-REGRESSOR *

[50]:
```
ada=AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=16))
grid={'n_estimators':[7,8,10],
         'learning_rate':[1.2,1.6,2],
         'loss':['linear', 'square', 'exponential']}
p(grid,ada,'adabstreg')
```

```
AdaBoostRegressor(estimator=DecisionTreeRegressor(max_depth=16))

Best Parameters {'n_estimators': 8, 'loss': 'linear', 'learning_rate': 1.6}
Best Score 97.39 %
```

[50]:

	Metric_Name	Values
0	R.sq(%)	96.790
1	Mean-Squared-Error	0.032
2	Rooted-Mean-Squared-Error	0.338
3	Mean-Absoulute-Error	0.114

```
4      Median-Absoulte-Error   0.069
```

13 *BAGGINGREGRESSOR*

[51]:
```
from sklearn.ensemble import BaggingRegressor
```

[52]:
```
bag=BaggingRegressor(estimator=DecisionTreeRegressor(max_depth=17),oob_score=False,n_jobs=-1)
bgrid={'n_estimators':[10,13,16]}
p(bgrid,bag,'bagreg')
```

```
BaggingRegressor(estimator=DecisionTreeRegressor(max_depth=17), n_jobs=-1)

Best Parameters {'n_estimators': 16}
Best Score 94.52 %
```

[52]:

	Metric_Name	Values
0	R.sq(%)	94.540
1	Mean-Squared-Error	0.054
2	Rooted-Mean-Squared-Error	0.384
3	Mean-Absoulute-Error	0.148
4	Median-Absoulte-Error	0.083

14 *RANDOMFORESTREGESSOR*

[53]:
```
from sklearn.ensemble import RandomForestRegressor
```

[54]:
```
r=RandomForestRegressor(n_jobs=-1,oob_score=True)
rgrid={'max_depth':[170,190,200,210],
       'max_features':['sqrt', 'log2'],
       'max_samples':[30,100,150,200],
       'max_leaf_nodes':[20,40,60,100]}

p(rgrid,r,'randfrstreg')
```

```
RandomForestRegressor(n_jobs=-1, oob_score=True)

Best Parameters {'max_samples': 200, 'max_leaf_nodes': 40, 'max_features':
'log2', 'max_depth': 170}
Best Score 25.39 %
```

[54]:

	Metric_Name	Values
0	R.sq(%)	25.350
1	Mean-Squared-Error	0.735
2	Rooted-Mean-Squared-Error	0.844

3	Mean-Absoulute-Error	0.713
4	Median-Absoulte-Error	0.662

15 *RADIUS NEIGHBOR REGRESSOR*

[55]:
```
from sklearn.neighbors import RadiusNeighborsRegressor
```

[56]:
```
rnr=RadiusNeighborsRegressor(algorithm='auto',n_jobs=-1)
rng={'weights':['uniform', 'distance'],
    'p':[1,2], 'leaf_size':[30,40,50]}
p(rng,rnr,'Radiusneighreg')
```

```
RadiusNeighborsRegressor(n_jobs=-1)

Best Parameters {'weights': 'distance', 'p': 1, 'leaf_size': 30}
Best Score 93.77 %
```

[56]:

	Metric_Name	Values
0	R.sq(%)	94.760
1	Mean-Squared-Error	0.052
2	Rooted-Mean-Squared-Error	0.362
3	Mean-Absoulute-Error	0.131
4	Median-Absoulte-Error	0.064

16 *STOCHASTIC-GRADIENT-DESCENT-REGRESSOR*

[57]:
```
from sklearn.linear_model import SGDRegressor
```

[58]:
```
sgd=SGDRegressor()
sgdg={'penalty':['l2', 'l1', 'elasticnet', None],
     'max_iter':[100,400,800], 'tol':[1e-3,1e-5,1e-8],'alpha':[0.1,.001,0.
 ↪0001,1],'learning_rate':['constant','optimal','invscaling','adaptive']
     }
p(sgdg,sgd,'sgdreg')
```

```
SGDRegressor()

Best Parameters {'tol': 0.001, 'penalty': 'l1', 'max_iter': 100,
'learning_rate': 'adaptive', 'alpha': 0.0001}
Best Score 22.65 %
```

[58]:

	Metric_Name	Values
0	R.sq(%)	23.110
1	Mean-Squared-Error	0.757

```
2  Rooted-Mean-Squared-Error   0.850
3       Mean-Absoulute-Error   0.723
4     Median-Absoulte-Error    0.666
```

17 *ARTIFICIAL NEURAL NETWORK MODEL-MULTIPLE LINEAR REGRESSION*

[98]:
```
import tensorflow as tf
from tensorflow import keras
from keras.layers import Dense
from tensorflow.keras.losses import MeanSquaredError
from keras.optimizers import SGD,Adam
```

[99]:
```
regNN=tf.keras.Sequential([Dense(units=256,input_dim=xtrain.
 ↪shape[1],activation='relu',name='l1'),
                           Dense(units=128,activation='relu',name='l2'),
                           Dense(units=64,activation='relu',name='l3'),
                           Dense(units=32,activation='relu',name='l4'),
                           Dense(units=16,activation='relu',name='l5'),
                           Dense(units=8,activation='relu',name='l6'),#2
                           Dense(units=4,activation='relu',name='l7'),
                           Dense(units=1,activation='linear',name='l8')
                          ])
```

[100]:
```
ms=tf.keras.losses.MeanSquaredError()
sgd=tf.keras.optimizers.SGD(learning_rate=0.001,nesterov=True,momentum=0.
 ↪001,ema_momentum=1)#epsilon=1e-03
regNN.compile(optimizer=sgd,loss=ms,metrics=['mean_absolute_error',␣
 ↪'mean_squared_error'],)
```

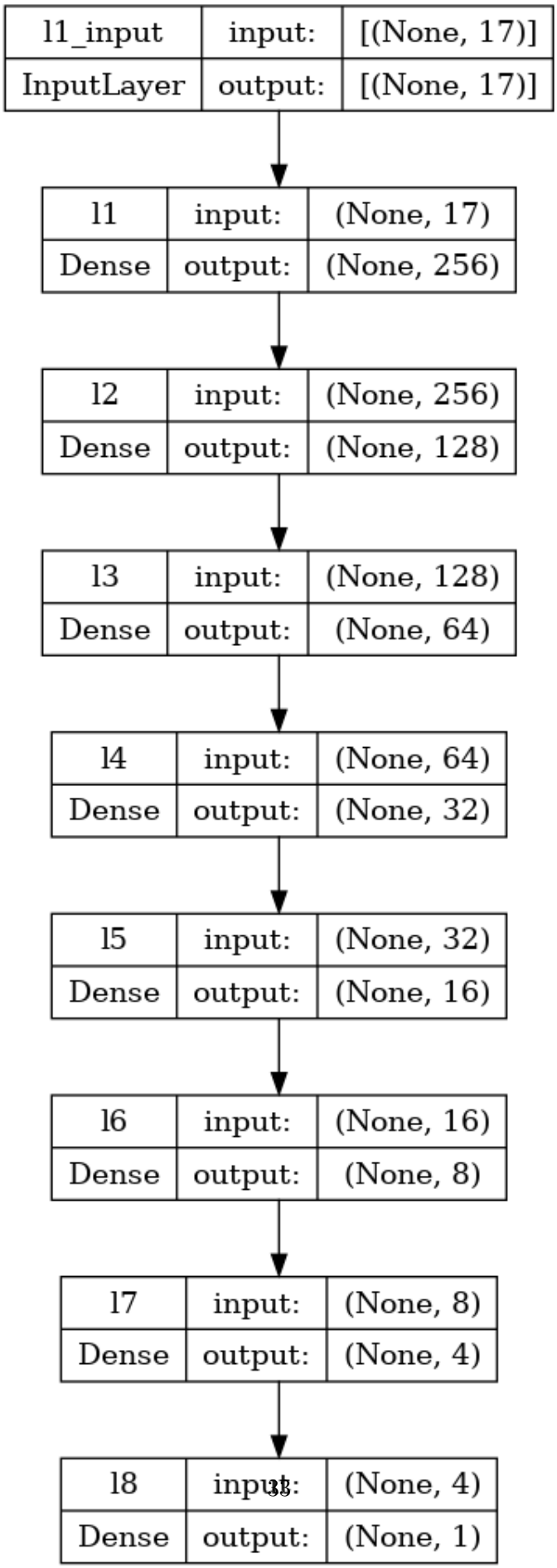
[101]:
```
regNN.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 l1 (Dense)                  (None, 256)               4608

 l2 (Dense)                  (None, 128)               32896

 l3 (Dense)                  (None, 64)                8256

 l4 (Dense)                  (None, 32)                2080

 l5 (Dense)                  (None, 16)                528
```

```
 l6 (Dense)                  (None, 8)                 136

 l7 (Dense)                  (None, 4)                 36

 l8 (Dense)                  (None, 1)                 5

=================================================================
Total params: 48,545
Trainable params: 48,545
Non-trainable params: 0
_________________________________________________________________
```

[102]:
```
from tensorflow.keras.utils import plot_model
plot_model(regNN,show_shapes=True,to_file='regNN.png')
```

[102]:


l1_input
input:
[(None, 17)]
InputLayer
output:
[(None, 17)]
l1
input:
(None, 17)
Dense
output:
(None, 256)
l2
input:
(None, 256)
Dense
output:
(None, 128)
l3
input:
(None, 128)
Dense
output:
(None, 64)
l4
input:
(None, 64)
Dense
output:
(None, 32)
l5
input:
(None, 32)
Dense
output:
(None, 16)
l6
input:
(None, 16)
Dense
output:
(None, 8)
l7
input:
(None, 8)
Dense
output:
(None, 4)
l8
input:
(None, 4)
Dense
output:
(None, 1)

[103]:
```
history=regNN.fit(xtrain,ytrain,shuffle=True,epochs=350,validation_split=.
↪2,steps_per_epoch=250)
his=history.history
```

```
Epoch 1/350
250/250 [==============================] - 2s 5ms/step - loss: 0.9670 -
mean_absolute_error: 0.8262 - mean_squared_error: 0.9670 - val_loss: 0.9567 -
val_mean_absolute_error: 0.8247 - val_mean_squared_error: 0.9567
Epoch 2/350
250/250 [==============================] - 1s 4ms/step - loss: 0.9177 -
mean_absolute_error: 0.8041 - mean_squared_error: 0.9177 - val_loss: 0.9176 -
val_mean_absolute_error: 0.8065 - val_mean_squared_error: 0.9176
Epoch 3/350
250/250 [==============================] - 1s 3ms/step - loss: 0.8843 -
mean_absolute_error: 0.7879 - mean_squared_error: 0.8843 - val_loss: 0.8887 -
val_mean_absolute_error: 0.7919 - val_mean_squared_error: 0.8887
Epoch 4/350
250/250 [==============================] - 1s 4ms/step - loss: 0.8601 -
mean_absolute_error: 0.7755 - mean_squared_error: 0.8601 - val_loss: 0.8689 -
val_mean_absolute_error: 0.7826 - val_mean_squared_error: 0.8689
Epoch 5/350
250/250 [==============================] - 1s 3ms/step - loss: 0.8418 -
mean_absolute_error: 0.7663 - mean_squared_error: 0.8418 - val_loss: 0.8534 -
val_mean_absolute_error: 0.7751 - val_mean_squared_error: 0.8534
Epoch 6/350
250/250 [==============================] - 1s 3ms/step - loss: 0.8276 -
mean_absolute_error: 0.7594 - mean_squared_error: 0.8276 - val_loss: 0.8404 -
val_mean_absolute_error: 0.7674 - val_mean_squared_error: 0.8404
Epoch 7/350
250/250 [==============================] - 1s 3ms/step - loss: 0.8166 -
mean_absolute_error: 0.7535 - mean_squared_error: 0.8166 - val_loss: 0.8314 -
val_mean_absolute_error: 0.7643 - val_mean_squared_error: 0.8314
Epoch 8/350
250/250 [==============================] - 1s 3ms/step - loss: 0.8080 -
mean_absolute_error: 0.7493 - mean_squared_error: 0.8080 - val_loss: 0.8239 -
val_mean_absolute_error: 0.7613 - val_mean_squared_error: 0.8239
Epoch 9/350
250/250 [==============================] - 1s 4ms/step - loss: 0.8009 -
mean_absolute_error: 0.7461 - mean_squared_error: 0.8009 - val_loss: 0.8163 -
val_mean_absolute_error: 0.7572 - val_mean_squared_error: 0.8163
Epoch 10/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7947 -
mean_absolute_error: 0.7431 - mean_squared_error: 0.7947 - val_loss: 0.8105 -
val_mean_absolute_error: 0.7549 - val_mean_squared_error: 0.8105
Epoch 11/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.7894 -
mean_absolute_error: 0.7404 - mean_squared_error: 0.7894 - val_loss: 0.8067 -
val_mean_absolute_error: 0.7544 - val_mean_squared_error: 0.8067
Epoch 12/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7852 -
mean_absolute_error: 0.7388 - mean_squared_error: 0.7852 - val_loss: 0.8019 -
val_mean_absolute_error: 0.7513 - val_mean_squared_error: 0.8019
Epoch 13/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7813 -
mean_absolute_error: 0.7365 - mean_squared_error: 0.7813 - val_loss: 0.7994 -
val_mean_absolute_error: 0.7517 - val_mean_squared_error: 0.7994
Epoch 14/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7784 -
mean_absolute_error: 0.7356 - mean_squared_error: 0.7784 - val_loss: 0.7953 -
val_mean_absolute_error: 0.7482 - val_mean_squared_error: 0.7953
Epoch 15/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7756 -
mean_absolute_error: 0.7340 - mean_squared_error: 0.7756 - val_loss: 0.7924 -
val_mean_absolute_error: 0.7461 - val_mean_squared_error: 0.7924
Epoch 16/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7731 -
mean_absolute_error: 0.7329 - mean_squared_error: 0.7731 - val_loss: 0.7903 -
val_mean_absolute_error: 0.7442 - val_mean_squared_error: 0.7903
Epoch 17/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7709 -
mean_absolute_error: 0.7315 - mean_squared_error: 0.7709 - val_loss: 0.7890 -
val_mean_absolute_error: 0.7461 - val_mean_squared_error: 0.7890
Epoch 18/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7688 -
mean_absolute_error: 0.7308 - mean_squared_error: 0.7688 - val_loss: 0.7876 -
val_mean_absolute_error: 0.7458 - val_mean_squared_error: 0.7876
Epoch 19/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7670 -
mean_absolute_error: 0.7299 - mean_squared_error: 0.7670 - val_loss: 0.7855 -
val_mean_absolute_error: 0.7445 - val_mean_squared_error: 0.7855
Epoch 20/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7652 -
mean_absolute_error: 0.7292 - mean_squared_error: 0.7652 - val_loss: 0.7837 -
val_mean_absolute_error: 0.7427 - val_mean_squared_error: 0.7837
Epoch 21/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7637 -
mean_absolute_error: 0.7282 - mean_squared_error: 0.7637 - val_loss: 0.7826 -
val_mean_absolute_error: 0.7428 - val_mean_squared_error: 0.7826
Epoch 22/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7620 -
mean_absolute_error: 0.7276 - mean_squared_error: 0.7620 - val_loss: 0.7808 -
val_mean_absolute_error: 0.7409 - val_mean_squared_error: 0.7808
Epoch 23/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.7606 -
mean_absolute_error: 0.7267 - mean_squared_error: 0.7606 - val_loss: 0.7797 -
val_mean_absolute_error: 0.7416 - val_mean_squared_error: 0.7797
Epoch 24/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7591 -
mean_absolute_error: 0.7261 - mean_squared_error: 0.7591 - val_loss: 0.7782 -
val_mean_absolute_error: 0.7405 - val_mean_squared_error: 0.7782
Epoch 25/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7577 -
mean_absolute_error: 0.7253 - mean_squared_error: 0.7577 - val_loss: 0.7772 -
val_mean_absolute_error: 0.7403 - val_mean_squared_error: 0.7772
Epoch 26/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7564 -
mean_absolute_error: 0.7250 - mean_squared_error: 0.7564 - val_loss: 0.7756 -
val_mean_absolute_error: 0.7382 - val_mean_squared_error: 0.7756
Epoch 27/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7551 -
mean_absolute_error: 0.7239 - mean_squared_error: 0.7551 - val_loss: 0.7756 -
val_mean_absolute_error: 0.7407 - val_mean_squared_error: 0.7756
Epoch 28/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7537 -
mean_absolute_error: 0.7239 - mean_squared_error: 0.7537 - val_loss: 0.7734 -
val_mean_absolute_error: 0.7368 - val_mean_squared_error: 0.7734
Epoch 29/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7527 -
mean_absolute_error: 0.7231 - mean_squared_error: 0.7527 - val_loss: 0.7723 -
val_mean_absolute_error: 0.7371 - val_mean_squared_error: 0.7723
Epoch 30/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7512 -
mean_absolute_error: 0.7224 - mean_squared_error: 0.7512 - val_loss: 0.7723 -
val_mean_absolute_error: 0.7393 - val_mean_squared_error: 0.7723
Epoch 31/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7501 -
mean_absolute_error: 0.7220 - mean_squared_error: 0.7501 - val_loss: 0.7718 -
val_mean_absolute_error: 0.7393 - val_mean_squared_error: 0.7718
Epoch 32/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7490 -
mean_absolute_error: 0.7214 - mean_squared_error: 0.7490 - val_loss: 0.7692 -
val_mean_absolute_error: 0.7365 - val_mean_squared_error: 0.7692
Epoch 33/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7478 -
mean_absolute_error: 0.7209 - mean_squared_error: 0.7478 - val_loss: 0.7681 -
val_mean_absolute_error: 0.7351 - val_mean_squared_error: 0.7681
Epoch 34/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7465 -
mean_absolute_error: 0.7201 - mean_squared_error: 0.7465 - val_loss: 0.7675 -
val_mean_absolute_error: 0.7365 - val_mean_squared_error: 0.7675
Epoch 35/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.7456 -
mean_absolute_error: 0.7204 - mean_squared_error: 0.7456 - val_loss: 0.7661 -
val_mean_absolute_error: 0.7331 - val_mean_squared_error: 0.7661
Epoch 36/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7445 -
mean_absolute_error: 0.7193 - mean_squared_error: 0.7445 - val_loss: 0.7658 -
val_mean_absolute_error: 0.7358 - val_mean_squared_error: 0.7658
Epoch 37/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7434 -
mean_absolute_error: 0.7191 - mean_squared_error: 0.7434 - val_loss: 0.7643 -
val_mean_absolute_error: 0.7338 - val_mean_squared_error: 0.7643
Epoch 38/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7424 -
mean_absolute_error: 0.7184 - mean_squared_error: 0.7424 - val_loss: 0.7633 -
val_mean_absolute_error: 0.7328 - val_mean_squared_error: 0.7633
Epoch 39/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7412 -
mean_absolute_error: 0.7177 - mean_squared_error: 0.7412 - val_loss: 0.7632 -
val_mean_absolute_error: 0.7351 - val_mean_squared_error: 0.7632
Epoch 40/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7399 -
mean_absolute_error: 0.7174 - mean_squared_error: 0.7399 - val_loss: 0.7619 -
val_mean_absolute_error: 0.7340 - val_mean_squared_error: 0.7619
Epoch 41/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7392 -
mean_absolute_error: 0.7171 - mean_squared_error: 0.7392 - val_loss: 0.7605 -
val_mean_absolute_error: 0.7317 - val_mean_squared_error: 0.7605
Epoch 42/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7380 -
mean_absolute_error: 0.7163 - mean_squared_error: 0.7380 - val_loss: 0.7596 -
val_mean_absolute_error: 0.7314 - val_mean_squared_error: 0.7596
Epoch 43/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7370 -
mean_absolute_error: 0.7158 - mean_squared_error: 0.7370 - val_loss: 0.7600 -
val_mean_absolute_error: 0.7343 - val_mean_squared_error: 0.7600
Epoch 44/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7359 -
mean_absolute_error: 0.7155 - mean_squared_error: 0.7359 - val_loss: 0.7579 -
val_mean_absolute_error: 0.7309 - val_mean_squared_error: 0.7579
Epoch 45/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7348 -
mean_absolute_error: 0.7149 - mean_squared_error: 0.7348 - val_loss: 0.7568 -
val_mean_absolute_error: 0.7305 - val_mean_squared_error: 0.7568
Epoch 46/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7338 -
mean_absolute_error: 0.7145 - mean_squared_error: 0.7338 - val_loss: 0.7561 -
val_mean_absolute_error: 0.7303 - val_mean_squared_error: 0.7561
Epoch 47/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.7329 -
mean_absolute_error: 0.7139 - mean_squared_error: 0.7329 - val_loss: 0.7557 -
val_mean_absolute_error: 0.7307 - val_mean_squared_error: 0.7557
Epoch 48/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7318 -
mean_absolute_error: 0.7136 - mean_squared_error: 0.7318 - val_loss: 0.7545 -
val_mean_absolute_error: 0.7282 - val_mean_squared_error: 0.7545
Epoch 49/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7306 -
mean_absolute_error: 0.7127 - mean_squared_error: 0.7306 - val_loss: 0.7542 -
val_mean_absolute_error: 0.7303 - val_mean_squared_error: 0.7542
Epoch 50/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7299 -
mean_absolute_error: 0.7125 - mean_squared_error: 0.7299 - val_loss: 0.7561 -
val_mean_absolute_error: 0.7343 - val_mean_squared_error: 0.7561
Epoch 51/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7291 -
mean_absolute_error: 0.7122 - mean_squared_error: 0.7291 - val_loss: 0.7525 -
val_mean_absolute_error: 0.7292 - val_mean_squared_error: 0.7525
Epoch 52/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7279 -
mean_absolute_error: 0.7115 - mean_squared_error: 0.7279 - val_loss: 0.7520 -
val_mean_absolute_error: 0.7300 - val_mean_squared_error: 0.7520
Epoch 53/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7268 -
mean_absolute_error: 0.7111 - mean_squared_error: 0.7268 - val_loss: 0.7530 -
val_mean_absolute_error: 0.7330 - val_mean_squared_error: 0.7530
Epoch 54/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7259 -
mean_absolute_error: 0.7104 - mean_squared_error: 0.7259 - val_loss: 0.7515 -
val_mean_absolute_error: 0.7319 - val_mean_squared_error: 0.7515
Epoch 55/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7247 -
mean_absolute_error: 0.7103 - mean_squared_error: 0.7247 - val_loss: 0.7503 -
val_mean_absolute_error: 0.7307 - val_mean_squared_error: 0.7503
Epoch 56/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7238 -
mean_absolute_error: 0.7096 - mean_squared_error: 0.7238 - val_loss: 0.7478 -
val_mean_absolute_error: 0.7277 - val_mean_squared_error: 0.7478
Epoch 57/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7229 -
mean_absolute_error: 0.7093 - mean_squared_error: 0.7229 - val_loss: 0.7470 -
val_mean_absolute_error: 0.7260 - val_mean_squared_error: 0.7470
Epoch 58/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7218 -
mean_absolute_error: 0.7084 - mean_squared_error: 0.7218 - val_loss: 0.7465 -
val_mean_absolute_error: 0.7278 - val_mean_squared_error: 0.7465
Epoch 59/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.7209 -
mean_absolute_error: 0.7082 - mean_squared_error: 0.7209 - val_loss: 0.7462 -
val_mean_absolute_error: 0.7272 - val_mean_squared_error: 0.7462
Epoch 60/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7200 -
mean_absolute_error: 0.7078 - mean_squared_error: 0.7200 - val_loss: 0.7456 -
val_mean_absolute_error: 0.7234 - val_mean_squared_error: 0.7456
Epoch 61/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7188 -
mean_absolute_error: 0.7069 - mean_squared_error: 0.7188 - val_loss: 0.7463 -
val_mean_absolute_error: 0.7296 - val_mean_squared_error: 0.7463
Epoch 62/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7180 -
mean_absolute_error: 0.7067 - mean_squared_error: 0.7180 - val_loss: 0.7429 -
val_mean_absolute_error: 0.7236 - val_mean_squared_error: 0.7429
Epoch 63/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7172 -
mean_absolute_error: 0.7065 - mean_squared_error: 0.7172 - val_loss: 0.7434 -
val_mean_absolute_error: 0.7203 - val_mean_squared_error: 0.7434
Epoch 64/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7162 -
mean_absolute_error: 0.7057 - mean_squared_error: 0.7162 - val_loss: 0.7418 -
val_mean_absolute_error: 0.7213 - val_mean_squared_error: 0.7418
Epoch 65/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7149 -
mean_absolute_error: 0.7050 - mean_squared_error: 0.7149 - val_loss: 0.7413 -
val_mean_absolute_error: 0.7249 - val_mean_squared_error: 0.7413
Epoch 66/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7142 -
mean_absolute_error: 0.7048 - mean_squared_error: 0.7142 - val_loss: 0.7399 -
val_mean_absolute_error: 0.7211 - val_mean_squared_error: 0.7399
Epoch 67/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7133 -
mean_absolute_error: 0.7043 - mean_squared_error: 0.7133 - val_loss: 0.7398 -
val_mean_absolute_error: 0.7210 - val_mean_squared_error: 0.7398
Epoch 68/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7127 -
mean_absolute_error: 0.7041 - mean_squared_error: 0.7127 - val_loss: 0.7389 -
val_mean_absolute_error: 0.7228 - val_mean_squared_error: 0.7389
Epoch 69/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7113 -
mean_absolute_error: 0.7032 - mean_squared_error: 0.7113 - val_loss: 0.7377 -
val_mean_absolute_error: 0.7205 - val_mean_squared_error: 0.7377
Epoch 70/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7104 -
mean_absolute_error: 0.7031 - mean_squared_error: 0.7104 - val_loss: 0.7370 -
val_mean_absolute_error: 0.7188 - val_mean_squared_error: 0.7370
Epoch 71/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.7093 -
mean_absolute_error: 0.7024 - mean_squared_error: 0.7093 - val_loss: 0.7365 -
val_mean_absolute_error: 0.7209 - val_mean_squared_error: 0.7365
Epoch 72/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7083 -
mean_absolute_error: 0.7018 - mean_squared_error: 0.7083 - val_loss: 0.7355 -
val_mean_absolute_error: 0.7192 - val_mean_squared_error: 0.7355
Epoch 73/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7074 -
mean_absolute_error: 0.7015 - mean_squared_error: 0.7074 - val_loss: 0.7348 -
val_mean_absolute_error: 0.7208 - val_mean_squared_error: 0.7348
Epoch 74/350
250/250 [==============================] - 1s 4ms/step - loss: 0.7061 -
mean_absolute_error: 0.7005 - mean_squared_error: 0.7061 - val_loss: 0.7334 -
val_mean_absolute_error: 0.7198 - val_mean_squared_error: 0.7334
Epoch 75/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7052 -
mean_absolute_error: 0.7004 - mean_squared_error: 0.7052 - val_loss: 0.7333 -
val_mean_absolute_error: 0.7205 - val_mean_squared_error: 0.7333
Epoch 76/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7040 -
mean_absolute_error: 0.6994 - mean_squared_error: 0.7040 - val_loss: 0.7388 -
val_mean_absolute_error: 0.7287 - val_mean_squared_error: 0.7388
Epoch 77/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7028 -
mean_absolute_error: 0.6993 - mean_squared_error: 0.7028 - val_loss: 0.7309 -
val_mean_absolute_error: 0.7164 - val_mean_squared_error: 0.7309
Epoch 78/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7019 -
mean_absolute_error: 0.6984 - mean_squared_error: 0.7019 - val_loss: 0.7308 -
val_mean_absolute_error: 0.7149 - val_mean_squared_error: 0.7308
Epoch 79/350
250/250 [==============================] - 1s 3ms/step - loss: 0.7008 -
mean_absolute_error: 0.6979 - mean_squared_error: 0.7008 - val_loss: 0.7299 -
val_mean_absolute_error: 0.7187 - val_mean_squared_error: 0.7299
Epoch 80/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6997 -
mean_absolute_error: 0.6973 - mean_squared_error: 0.6997 - val_loss: 0.7289 -
val_mean_absolute_error: 0.7168 - val_mean_squared_error: 0.7289
Epoch 81/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6981 -
mean_absolute_error: 0.6966 - mean_squared_error: 0.6981 - val_loss: 0.7303 -
val_mean_absolute_error: 0.7219 - val_mean_squared_error: 0.7303
Epoch 82/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6970 -
mean_absolute_error: 0.6958 - mean_squared_error: 0.6970 - val_loss: 0.7309 -
val_mean_absolute_error: 0.7237 - val_mean_squared_error: 0.7309
Epoch 83/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.6957 -
mean_absolute_error: 0.6955 - mean_squared_error: 0.6957 - val_loss: 0.7257 -
val_mean_absolute_error: 0.7141 - val_mean_squared_error: 0.7257
Epoch 84/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6944 -
mean_absolute_error: 0.6945 - mean_squared_error: 0.6944 - val_loss: 0.7241 -
val_mean_absolute_error: 0.7161 - val_mean_squared_error: 0.7241
Epoch 85/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6932 -
mean_absolute_error: 0.6938 - mean_squared_error: 0.6932 - val_loss: 0.7254 -
val_mean_absolute_error: 0.7203 - val_mean_squared_error: 0.7254
Epoch 86/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6916 -
mean_absolute_error: 0.6931 - mean_squared_error: 0.6916 - val_loss: 0.7213 -
val_mean_absolute_error: 0.7126 - val_mean_squared_error: 0.7213
Epoch 87/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6909 -
mean_absolute_error: 0.6926 - mean_squared_error: 0.6909 - val_loss: 0.7219 -
val_mean_absolute_error: 0.7171 - val_mean_squared_error: 0.7219
Epoch 88/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6891 -
mean_absolute_error: 0.6916 - mean_squared_error: 0.6891 - val_loss: 0.7236 -
val_mean_absolute_error: 0.7190 - val_mean_squared_error: 0.7236
Epoch 89/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6873 -
mean_absolute_error: 0.6913 - mean_squared_error: 0.6873 - val_loss: 0.7198 -
val_mean_absolute_error: 0.7071 - val_mean_squared_error: 0.7198
Epoch 90/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6853 -
mean_absolute_error: 0.6893 - mean_squared_error: 0.6853 - val_loss: 0.7293 -
val_mean_absolute_error: 0.7263 - val_mean_squared_error: 0.7293
Epoch 91/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6844 -
mean_absolute_error: 0.6893 - mean_squared_error: 0.6844 - val_loss: 0.7163 -
val_mean_absolute_error: 0.7066 - val_mean_squared_error: 0.7163
Epoch 92/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6828 -
mean_absolute_error: 0.6877 - mean_squared_error: 0.6828 - val_loss: 0.7126 -
val_mean_absolute_error: 0.7086 - val_mean_squared_error: 0.7126
Epoch 93/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6804 -
mean_absolute_error: 0.6868 - mean_squared_error: 0.6804 - val_loss: 0.7187 -
val_mean_absolute_error: 0.7195 - val_mean_squared_error: 0.7187
Epoch 94/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6796 -
mean_absolute_error: 0.6868 - mean_squared_error: 0.6796 - val_loss: 0.7198 -
val_mean_absolute_error: 0.7205 - val_mean_squared_error: 0.7198
Epoch 95/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.6774 -
mean_absolute_error: 0.6856 - mean_squared_error: 0.6774 - val_loss: 0.7085 -
val_mean_absolute_error: 0.7022 - val_mean_squared_error: 0.7085
Epoch 96/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6750 -
mean_absolute_error: 0.6845 - mean_squared_error: 0.6750 - val_loss: 0.7083 -
val_mean_absolute_error: 0.7015 - val_mean_squared_error: 0.7083
Epoch 97/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6735 -
mean_absolute_error: 0.6837 - mean_squared_error: 0.6735 - val_loss: 0.7093 -
val_mean_absolute_error: 0.6973 - val_mean_squared_error: 0.7093
Epoch 98/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6711 -
mean_absolute_error: 0.6820 - mean_squared_error: 0.6711 - val_loss: 0.7015 -
val_mean_absolute_error: 0.7035 - val_mean_squared_error: 0.7015
Epoch 99/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6685 -
mean_absolute_error: 0.6810 - mean_squared_error: 0.6685 - val_loss: 0.7013 -
val_mean_absolute_error: 0.6968 - val_mean_squared_error: 0.7013
Epoch 100/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6667 -
mean_absolute_error: 0.6797 - mean_squared_error: 0.6667 - val_loss: 0.7095 -
val_mean_absolute_error: 0.6925 - val_mean_squared_error: 0.7095
Epoch 101/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6645 -
mean_absolute_error: 0.6784 - mean_squared_error: 0.6645 - val_loss: 0.6959 -
val_mean_absolute_error: 0.6929 - val_mean_squared_error: 0.6959
Epoch 102/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6616 -
mean_absolute_error: 0.6769 - mean_squared_error: 0.6616 - val_loss: 0.6944 -
val_mean_absolute_error: 0.6905 - val_mean_squared_error: 0.6944
Epoch 103/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6587 -
mean_absolute_error: 0.6756 - mean_squared_error: 0.6587 - val_loss: 0.6886 -
val_mean_absolute_error: 0.6974 - val_mean_squared_error: 0.6886
Epoch 104/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6556 -
mean_absolute_error: 0.6734 - mean_squared_error: 0.6556 - val_loss: 0.6893 -
val_mean_absolute_error: 0.6893 - val_mean_squared_error: 0.6893
Epoch 105/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6522 -
mean_absolute_error: 0.6714 - mean_squared_error: 0.6522 - val_loss: 0.6842 -
val_mean_absolute_error: 0.6988 - val_mean_squared_error: 0.6842
Epoch 106/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6487 -
mean_absolute_error: 0.6697 - mean_squared_error: 0.6487 - val_loss: 0.6782 -
val_mean_absolute_error: 0.6941 - val_mean_squared_error: 0.6782
Epoch 107/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.6461 -
mean_absolute_error: 0.6689 - mean_squared_error: 0.6461 - val_loss: 0.6745 -
val_mean_absolute_error: 0.6863 - val_mean_squared_error: 0.6745
Epoch 108/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6414 -
mean_absolute_error: 0.6663 - mean_squared_error: 0.6414 - val_loss: 0.6768 -
val_mean_absolute_error: 0.6969 - val_mean_squared_error: 0.6768
Epoch 109/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6385 -
mean_absolute_error: 0.6643 - mean_squared_error: 0.6385 - val_loss: 0.6659 -
val_mean_absolute_error: 0.6773 - val_mean_squared_error: 0.6659
Epoch 110/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6332 -
mean_absolute_error: 0.6618 - mean_squared_error: 0.6332 - val_loss: 0.6646 -
val_mean_absolute_error: 0.6784 - val_mean_squared_error: 0.6646
Epoch 111/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6285 -
mean_absolute_error: 0.6591 - mean_squared_error: 0.6285 - val_loss: 0.6714 -
val_mean_absolute_error: 0.6666 - val_mean_squared_error: 0.6714
Epoch 112/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6235 -
mean_absolute_error: 0.6560 - mean_squared_error: 0.6235 - val_loss: 0.6526 -
val_mean_absolute_error: 0.6808 - val_mean_squared_error: 0.6526
Epoch 113/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6187 -
mean_absolute_error: 0.6535 - mean_squared_error: 0.6187 - val_loss: 0.6736 -
val_mean_absolute_error: 0.7012 - val_mean_squared_error: 0.6736
Epoch 114/350
250/250 [==============================] - 1s 4ms/step - loss: 0.6123 -
mean_absolute_error: 0.6495 - mean_squared_error: 0.6123 - val_loss: 0.6438 -
val_mean_absolute_error: 0.6596 - val_mean_squared_error: 0.6438
Epoch 115/350
250/250 [==============================] - 1s 3ms/step - loss: 0.6043 -
mean_absolute_error: 0.6454 - mean_squared_error: 0.6043 - val_loss: 0.6241 -
val_mean_absolute_error: 0.6596 - val_mean_squared_error: 0.6241
Epoch 116/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5958 -
mean_absolute_error: 0.6398 - mean_squared_error: 0.5958 - val_loss: 0.6746 -
val_mean_absolute_error: 0.7031 - val_mean_squared_error: 0.6746
Epoch 117/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5908 -
mean_absolute_error: 0.6372 - mean_squared_error: 0.5908 - val_loss: 0.6154 -
val_mean_absolute_error: 0.6458 - val_mean_squared_error: 0.6154
Epoch 118/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5835 -
mean_absolute_error: 0.6328 - mean_squared_error: 0.5835 - val_loss: 0.6061 -
val_mean_absolute_error: 0.6453 - val_mean_squared_error: 0.6061
Epoch 119/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.5728 -
mean_absolute_error: 0.6274 - mean_squared_error: 0.5728 - val_loss: 0.6388 -
val_mean_absolute_error: 0.6836 - val_mean_squared_error: 0.6388
Epoch 120/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5638 -
mean_absolute_error: 0.6213 - mean_squared_error: 0.5638 - val_loss: 0.5731 -
val_mean_absolute_error: 0.6252 - val_mean_squared_error: 0.5731
Epoch 121/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5621 -
mean_absolute_error: 0.6195 - mean_squared_error: 0.5621 - val_loss: 0.5678 -
val_mean_absolute_error: 0.6169 - val_mean_squared_error: 0.5678
Epoch 122/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5498 -
mean_absolute_error: 0.6118 - mean_squared_error: 0.5498 - val_loss: 0.5575 -
val_mean_absolute_error: 0.6091 - val_mean_squared_error: 0.5575
Epoch 123/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5457 -
mean_absolute_error: 0.6085 - mean_squared_error: 0.5457 - val_loss: 0.5408 -
val_mean_absolute_error: 0.6077 - val_mean_squared_error: 0.5408
Epoch 124/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5526 -
mean_absolute_error: 0.6109 - mean_squared_error: 0.5526 - val_loss: 0.5551 -
val_mean_absolute_error: 0.6379 - val_mean_squared_error: 0.5551
Epoch 125/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5415 -
mean_absolute_error: 0.6030 - mean_squared_error: 0.5415 - val_loss: 0.5478 -
val_mean_absolute_error: 0.5986 - val_mean_squared_error: 0.5478
Epoch 126/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5470 -
mean_absolute_error: 0.6038 - mean_squared_error: 0.5470 - val_loss: 0.5041 -
val_mean_absolute_error: 0.5858 - val_mean_squared_error: 0.5041
Epoch 127/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5412 -
mean_absolute_error: 0.6010 - mean_squared_error: 0.5412 - val_loss: 0.5158 -
val_mean_absolute_error: 0.6073 - val_mean_squared_error: 0.5158
Epoch 128/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5414 -
mean_absolute_error: 0.6004 - mean_squared_error: 0.5414 - val_loss: 0.5738 -
val_mean_absolute_error: 0.6514 - val_mean_squared_error: 0.5738
Epoch 129/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5254 -
mean_absolute_error: 0.5925 - mean_squared_error: 0.5254 - val_loss: 0.5754 -
val_mean_absolute_error: 0.6516 - val_mean_squared_error: 0.5754
Epoch 130/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5404 -
mean_absolute_error: 0.6003 - mean_squared_error: 0.5404 - val_loss: 0.5209 -
val_mean_absolute_error: 0.6209 - val_mean_squared_error: 0.5209
Epoch 131/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.5364 -
mean_absolute_error: 0.5962 - mean_squared_error: 0.5364 - val_loss: 0.5760 -
val_mean_absolute_error: 0.5867 - val_mean_squared_error: 0.5760
Epoch 132/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5118 -
mean_absolute_error: 0.5841 - mean_squared_error: 0.5118 - val_loss: 0.4600 -
val_mean_absolute_error: 0.5785 - val_mean_squared_error: 0.4600
Epoch 133/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5281 -
mean_absolute_error: 0.5907 - mean_squared_error: 0.5281 - val_loss: 0.6208 -
val_mean_absolute_error: 0.5962 - val_mean_squared_error: 0.6208
Epoch 134/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5166 -
mean_absolute_error: 0.5851 - mean_squared_error: 0.5166 - val_loss: 0.7309 -
val_mean_absolute_error: 0.6485 - val_mean_squared_error: 0.7309
Epoch 135/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5077 -
mean_absolute_error: 0.5781 - mean_squared_error: 0.5077 - val_loss: 0.4248 -
val_mean_absolute_error: 0.5409 - val_mean_squared_error: 0.4248
Epoch 136/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5152 -
mean_absolute_error: 0.5823 - mean_squared_error: 0.5152 - val_loss: 0.8753 -
val_mean_absolute_error: 0.7162 - val_mean_squared_error: 0.8753
Epoch 137/350
250/250 [==============================] - 1s 4ms/step - loss: 0.5016 -
mean_absolute_error: 0.5734 - mean_squared_error: 0.5016 - val_loss: 0.5821 -
val_mean_absolute_error: 0.6546 - val_mean_squared_error: 0.5821
Epoch 138/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4741 -
mean_absolute_error: 0.5599 - mean_squared_error: 0.4741 - val_loss: 0.7032 -
val_mean_absolute_error: 0.6346 - val_mean_squared_error: 0.7032
Epoch 139/350
250/250 [==============================] - 1s 3ms/step - loss: 0.5079 -
mean_absolute_error: 0.5766 - mean_squared_error: 0.5079 - val_loss: 0.5110 -
val_mean_absolute_error: 0.5471 - val_mean_squared_error: 0.5110
Epoch 140/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4893 -
mean_absolute_error: 0.5650 - mean_squared_error: 0.4893 - val_loss: 0.7646 -
val_mean_absolute_error: 0.6708 - val_mean_squared_error: 0.7646
Epoch 141/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4652 -
mean_absolute_error: 0.5504 - mean_squared_error: 0.4652 - val_loss: 0.3903 -
val_mean_absolute_error: 0.5070 - val_mean_squared_error: 0.3903
Epoch 142/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4946 -
mean_absolute_error: 0.5686 - mean_squared_error: 0.4946 - val_loss: 0.4126 -
val_mean_absolute_error: 0.5104 - val_mean_squared_error: 0.4126
Epoch 143/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.4699 -
mean_absolute_error: 0.5538 - mean_squared_error: 0.4699 - val_loss: 0.3922 -
val_mean_absolute_error: 0.4941 - val_mean_squared_error: 0.3922
Epoch 144/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4675 -
mean_absolute_error: 0.5502 - mean_squared_error: 0.4675 - val_loss: 0.3979 -
val_mean_absolute_error: 0.4913 - val_mean_squared_error: 0.3979
Epoch 145/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4741 -
mean_absolute_error: 0.5533 - mean_squared_error: 0.4741 - val_loss: 0.4688 -
val_mean_absolute_error: 0.5813 - val_mean_squared_error: 0.4688
Epoch 146/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4605 -
mean_absolute_error: 0.5483 - mean_squared_error: 0.4605 - val_loss: 0.5311 -
val_mean_absolute_error: 0.5537 - val_mean_squared_error: 0.5311
Epoch 147/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4435 -
mean_absolute_error: 0.5344 - mean_squared_error: 0.4435 - val_loss: 0.3713 -
val_mean_absolute_error: 0.5009 - val_mean_squared_error: 0.3713
Epoch 148/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4521 -
mean_absolute_error: 0.5417 - mean_squared_error: 0.4521 - val_loss: 0.3375 -
val_mean_absolute_error: 0.4819 - val_mean_squared_error: 0.3375
Epoch 149/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4433 -
mean_absolute_error: 0.5375 - mean_squared_error: 0.4433 - val_loss: 0.4895 -
val_mean_absolute_error: 0.5290 - val_mean_squared_error: 0.4895
Epoch 150/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4502 -
mean_absolute_error: 0.5400 - mean_squared_error: 0.4502 - val_loss: 0.3861 -
val_mean_absolute_error: 0.5324 - val_mean_squared_error: 0.3861
Epoch 151/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4246 -
mean_absolute_error: 0.5234 - mean_squared_error: 0.4246 - val_loss: 0.3988 -
val_mean_absolute_error: 0.4829 - val_mean_squared_error: 0.3988
Epoch 152/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4432 -
mean_absolute_error: 0.5362 - mean_squared_error: 0.4432 - val_loss: 0.3141 -
val_mean_absolute_error: 0.4696 - val_mean_squared_error: 0.3141
Epoch 153/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4139 -
mean_absolute_error: 0.5181 - mean_squared_error: 0.4139 - val_loss: 0.4216 -
val_mean_absolute_error: 0.5586 - val_mean_squared_error: 0.4216
Epoch 154/350
250/250 [==============================] - 1s 4ms/step - loss: 0.4202 -
mean_absolute_error: 0.5211 - mean_squared_error: 0.4202 - val_loss: 0.3185 -
val_mean_absolute_error: 0.4351 - val_mean_squared_error: 0.3185
Epoch 155/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.4174 -
mean_absolute_error: 0.5180 - mean_squared_error: 0.4174 - val_loss: 0.5576 -
val_mean_absolute_error: 0.5777 - val_mean_squared_error: 0.5576
Epoch 156/350
250/250 [==============================] - 1s 3ms/step - loss: 0.4138 -
mean_absolute_error: 0.5178 - mean_squared_error: 0.4138 - val_loss: 0.3668 -
val_mean_absolute_error: 0.5201 - val_mean_squared_error: 0.3668
Epoch 157/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3931 -
mean_absolute_error: 0.5045 - mean_squared_error: 0.3931 - val_loss: 0.4985 -
val_mean_absolute_error: 0.5406 - val_mean_squared_error: 0.4985
Epoch 158/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3899 -
mean_absolute_error: 0.5017 - mean_squared_error: 0.3899 - val_loss: 0.4606 -
val_mean_absolute_error: 0.5944 - val_mean_squared_error: 0.4606
Epoch 159/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3983 -
mean_absolute_error: 0.5060 - mean_squared_error: 0.3983 - val_loss: 0.5730 -
val_mean_absolute_error: 0.6559 - val_mean_squared_error: 0.5730
Epoch 160/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3889 -
mean_absolute_error: 0.5012 - mean_squared_error: 0.3889 - val_loss: 0.2995 -
val_mean_absolute_error: 0.4177 - val_mean_squared_error: 0.2995
Epoch 161/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3802 -
mean_absolute_error: 0.4944 - mean_squared_error: 0.3802 - val_loss: 0.3699 -
val_mean_absolute_error: 0.4668 - val_mean_squared_error: 0.3699
Epoch 162/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3711 -
mean_absolute_error: 0.4869 - mean_squared_error: 0.3711 - val_loss: 0.2473 -
val_mean_absolute_error: 0.3938 - val_mean_squared_error: 0.2473
Epoch 163/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3702 -
mean_absolute_error: 0.4870 - mean_squared_error: 0.3702 - val_loss: 0.2505 -
val_mean_absolute_error: 0.3998 - val_mean_squared_error: 0.2505
Epoch 164/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3536 -
mean_absolute_error: 0.4744 - mean_squared_error: 0.3536 - val_loss: 0.3640 -
val_mean_absolute_error: 0.5108 - val_mean_squared_error: 0.3640
Epoch 165/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3570 -
mean_absolute_error: 0.4772 - mean_squared_error: 0.3570 - val_loss: 0.4506 -
val_mean_absolute_error: 0.5773 - val_mean_squared_error: 0.4506
Epoch 166/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3578 -
mean_absolute_error: 0.4797 - mean_squared_error: 0.3578 - val_loss: 0.2967 -
val_mean_absolute_error: 0.4161 - val_mean_squared_error: 0.2967
Epoch 167/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.3524 -
mean_absolute_error: 0.4756 - mean_squared_error: 0.3524 - val_loss: 0.2721 -
val_mean_absolute_error: 0.4311 - val_mean_squared_error: 0.2721
Epoch 168/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3627 -
mean_absolute_error: 0.4843 - mean_squared_error: 0.3627 - val_loss: 0.2927 -
val_mean_absolute_error: 0.4508 - val_mean_squared_error: 0.2927
Epoch 169/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3602 -
mean_absolute_error: 0.4797 - mean_squared_error: 0.3602 - val_loss: 0.4613 -
val_mean_absolute_error: 0.5920 - val_mean_squared_error: 0.4613
Epoch 170/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3443 -
mean_absolute_error: 0.4679 - mean_squared_error: 0.3443 - val_loss: 0.2683 -
val_mean_absolute_error: 0.3915 - val_mean_squared_error: 0.2683
Epoch 171/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3303 -
mean_absolute_error: 0.4586 - mean_squared_error: 0.3303 - val_loss: 0.2326 -
val_mean_absolute_error: 0.3846 - val_mean_squared_error: 0.2326
Epoch 172/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3381 -
mean_absolute_error: 0.4662 - mean_squared_error: 0.3381 - val_loss: 0.3395 -
val_mean_absolute_error: 0.4425 - val_mean_squared_error: 0.3395
Epoch 173/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3378 -
mean_absolute_error: 0.4643 - mean_squared_error: 0.3378 - val_loss: 0.4840 -
val_mean_absolute_error: 0.5869 - val_mean_squared_error: 0.4840
Epoch 174/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3179 -
mean_absolute_error: 0.4494 - mean_squared_error: 0.3179 - val_loss: 0.6881 -
val_mean_absolute_error: 0.6746 - val_mean_squared_error: 0.6881
Epoch 175/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3224 -
mean_absolute_error: 0.4526 - mean_squared_error: 0.3224 - val_loss: 0.2447 -
val_mean_absolute_error: 0.3711 - val_mean_squared_error: 0.2447
Epoch 176/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3145 -
mean_absolute_error: 0.4462 - mean_squared_error: 0.3145 - val_loss: 0.3081 -
val_mean_absolute_error: 0.4766 - val_mean_squared_error: 0.3081
Epoch 177/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3094 -
mean_absolute_error: 0.4424 - mean_squared_error: 0.3094 - val_loss: 0.3432 -
val_mean_absolute_error: 0.5139 - val_mean_squared_error: 0.3432
Epoch 178/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3150 -
mean_absolute_error: 0.4494 - mean_squared_error: 0.3150 - val_loss: 0.2402 -
val_mean_absolute_error: 0.4139 - val_mean_squared_error: 0.2402
Epoch 179/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.3073 -
mean_absolute_error: 0.4388 - mean_squared_error: 0.3073 - val_loss: 0.3543 -
val_mean_absolute_error: 0.5110 - val_mean_squared_error: 0.3543
Epoch 180/350
250/250 [==============================] - 1s 3ms/step - loss: 0.3048 -
mean_absolute_error: 0.4407 - mean_squared_error: 0.3048 - val_loss: 0.2938 -
val_mean_absolute_error: 0.4554 - val_mean_squared_error: 0.2938
Epoch 181/350
250/250 [==============================] - 1s 4ms/step - loss: 0.3113 -
mean_absolute_error: 0.4469 - mean_squared_error: 0.3113 - val_loss: 0.2732 -
val_mean_absolute_error: 0.4519 - val_mean_squared_error: 0.2732
Epoch 182/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2857 -
mean_absolute_error: 0.4235 - mean_squared_error: 0.2857 - val_loss: 0.2033 -
val_mean_absolute_error: 0.3676 - val_mean_squared_error: 0.2033
Epoch 183/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2878 -
mean_absolute_error: 0.4284 - mean_squared_error: 0.2878 - val_loss: 0.2495 -
val_mean_absolute_error: 0.4199 - val_mean_squared_error: 0.2495
Epoch 184/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2930 -
mean_absolute_error: 0.4307 - mean_squared_error: 0.2930 - val_loss: 0.3249 -
val_mean_absolute_error: 0.4924 - val_mean_squared_error: 0.3249
Epoch 185/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2806 -
mean_absolute_error: 0.4210 - mean_squared_error: 0.2806 - val_loss: 0.4564 -
val_mean_absolute_error: 0.5852 - val_mean_squared_error: 0.4564
Epoch 186/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2854 -
mean_absolute_error: 0.4255 - mean_squared_error: 0.2854 - val_loss: 0.3296 -
val_mean_absolute_error: 0.4538 - val_mean_squared_error: 0.3296
Epoch 187/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2749 -
mean_absolute_error: 0.4184 - mean_squared_error: 0.2749 - val_loss: 0.2299 -
val_mean_absolute_error: 0.3597 - val_mean_squared_error: 0.2299
Epoch 188/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2629 -
mean_absolute_error: 0.4057 - mean_squared_error: 0.2629 - val_loss: 0.1608 -
val_mean_absolute_error: 0.3315 - val_mean_squared_error: 0.1608
Epoch 189/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2780 -
mean_absolute_error: 0.4218 - mean_squared_error: 0.2780 - val_loss: 0.1715 -
val_mean_absolute_error: 0.3387 - val_mean_squared_error: 0.1715
Epoch 190/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2636 -
mean_absolute_error: 0.4081 - mean_squared_error: 0.2636 - val_loss: 0.1988 -
val_mean_absolute_error: 0.3277 - val_mean_squared_error: 0.1988
Epoch 191/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.2715 -
mean_absolute_error: 0.4162 - mean_squared_error: 0.2715 - val_loss: 0.6950 -
val_mean_absolute_error: 0.7179 - val_mean_squared_error: 0.6950
Epoch 192/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2684 -
mean_absolute_error: 0.4114 - mean_squared_error: 0.2684 - val_loss: 0.6306 -
val_mean_absolute_error: 0.6833 - val_mean_squared_error: 0.6306
Epoch 193/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2611 -
mean_absolute_error: 0.4048 - mean_squared_error: 0.2611 - val_loss: 0.3486 -
val_mean_absolute_error: 0.4886 - val_mean_squared_error: 0.3486
Epoch 194/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2530 -
mean_absolute_error: 0.4003 - mean_squared_error: 0.2530 - val_loss: 0.2427 -
val_mean_absolute_error: 0.4205 - val_mean_squared_error: 0.2427
Epoch 195/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2591 -
mean_absolute_error: 0.4053 - mean_squared_error: 0.2591 - val_loss: 0.4544 -
val_mean_absolute_error: 0.5458 - val_mean_squared_error: 0.4544
Epoch 196/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2474 -
mean_absolute_error: 0.3948 - mean_squared_error: 0.2474 - val_loss: 0.6565 -
val_mean_absolute_error: 0.6710 - val_mean_squared_error: 0.6565
Epoch 197/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2478 -
mean_absolute_error: 0.3962 - mean_squared_error: 0.2478 - val_loss: 0.2508 -
val_mean_absolute_error: 0.4280 - val_mean_squared_error: 0.2508
Epoch 198/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2347 -
mean_absolute_error: 0.3819 - mean_squared_error: 0.2347 - val_loss: 0.2688 -
val_mean_absolute_error: 0.3965 - val_mean_squared_error: 0.2688
Epoch 199/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2428 -
mean_absolute_error: 0.3915 - mean_squared_error: 0.2428 - val_loss: 0.4642 -
val_mean_absolute_error: 0.5827 - val_mean_squared_error: 0.4642
Epoch 200/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2444 -
mean_absolute_error: 0.3945 - mean_squared_error: 0.2444 - val_loss: 0.2132 -
val_mean_absolute_error: 0.3511 - val_mean_squared_error: 0.2132
Epoch 201/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2192 -
mean_absolute_error: 0.3695 - mean_squared_error: 0.2192 - val_loss: 0.2057 -
val_mean_absolute_error: 0.3553 - val_mean_squared_error: 0.2057
Epoch 202/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2466 -
mean_absolute_error: 0.3938 - mean_squared_error: 0.2466 - val_loss: 0.1688 -
val_mean_absolute_error: 0.3393 - val_mean_squared_error: 0.1688
Epoch 203/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.2188 -
mean_absolute_error: 0.3672 - mean_squared_error: 0.2188 - val_loss: 0.2764 -
val_mean_absolute_error: 0.4036 - val_mean_squared_error: 0.2764
Epoch 204/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2366 -
mean_absolute_error: 0.3831 - mean_squared_error: 0.2366 - val_loss: 0.1506 -
val_mean_absolute_error: 0.2813 - val_mean_squared_error: 0.1506
Epoch 205/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2293 -
mean_absolute_error: 0.3765 - mean_squared_error: 0.2293 - val_loss: 0.3052 -
val_mean_absolute_error: 0.4733 - val_mean_squared_error: 0.3052
Epoch 206/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2150 -
mean_absolute_error: 0.3657 - mean_squared_error: 0.2150 - val_loss: 0.1251 -
val_mean_absolute_error: 0.2746 - val_mean_squared_error: 0.1251
Epoch 207/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2255 -
mean_absolute_error: 0.3783 - mean_squared_error: 0.2255 - val_loss: 0.2624 -
val_mean_absolute_error: 0.4477 - val_mean_squared_error: 0.2624
Epoch 208/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2031 -
mean_absolute_error: 0.3544 - mean_squared_error: 0.2031 - val_loss: 0.1129 -
val_mean_absolute_error: 0.2405 - val_mean_squared_error: 0.1129
Epoch 209/350
250/250 [==============================] - 1s 4ms/step - loss: 0.2144 -
mean_absolute_error: 0.3676 - mean_squared_error: 0.2144 - val_loss: 0.1171 -
val_mean_absolute_error: 0.2645 - val_mean_squared_error: 0.1171
Epoch 210/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2143 -
mean_absolute_error: 0.3663 - mean_squared_error: 0.2143 - val_loss: 0.1272 -
val_mean_absolute_error: 0.2811 - val_mean_squared_error: 0.1272
Epoch 211/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2115 -
mean_absolute_error: 0.3603 - mean_squared_error: 0.2115 - val_loss: 0.1679 -
val_mean_absolute_error: 0.3340 - val_mean_squared_error: 0.1679
Epoch 212/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1986 -
mean_absolute_error: 0.3478 - mean_squared_error: 0.1986 - val_loss: 0.1036 -
val_mean_absolute_error: 0.2482 - val_mean_squared_error: 0.1036
Epoch 213/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2013 -
mean_absolute_error: 0.3494 - mean_squared_error: 0.2013 - val_loss: 0.0946 -
val_mean_absolute_error: 0.2228 - val_mean_squared_error: 0.0946
Epoch 214/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1866 -
mean_absolute_error: 0.3384 - mean_squared_error: 0.1866 - val_loss: 0.2614 -
val_mean_absolute_error: 0.4416 - val_mean_squared_error: 0.2614
Epoch 215/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.2027 -
mean_absolute_error: 0.3554 - mean_squared_error: 0.2027 - val_loss: 0.2304 -
val_mean_absolute_error: 0.4191 - val_mean_squared_error: 0.2304
Epoch 216/350
250/250 [==============================] - 1s 3ms/step - loss: 0.2015 -
mean_absolute_error: 0.3563 - mean_squared_error: 0.2015 - val_loss: 0.1266 -
val_mean_absolute_error: 0.2846 - val_mean_squared_error: 0.1266
Epoch 217/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1844 -
mean_absolute_error: 0.3394 - mean_squared_error: 0.1844 - val_loss: 0.1492 -
val_mean_absolute_error: 0.2868 - val_mean_squared_error: 0.1492
Epoch 218/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1936 -
mean_absolute_error: 0.3486 - mean_squared_error: 0.1936 - val_loss: 0.1292 -
val_mean_absolute_error: 0.2966 - val_mean_squared_error: 0.1292
Epoch 219/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1799 -
mean_absolute_error: 0.3343 - mean_squared_error: 0.1799 - val_loss: 0.1445 -
val_mean_absolute_error: 0.3117 - val_mean_squared_error: 0.1445
Epoch 220/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1876 -
mean_absolute_error: 0.3409 - mean_squared_error: 0.1876 - val_loss: 0.1081 -
val_mean_absolute_error: 0.2418 - val_mean_squared_error: 0.1081
Epoch 221/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1745 -
mean_absolute_error: 0.3288 - mean_squared_error: 0.1745 - val_loss: 0.2456 -
val_mean_absolute_error: 0.4309 - val_mean_squared_error: 0.2456
Epoch 222/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1835 -
mean_absolute_error: 0.3374 - mean_squared_error: 0.1835 - val_loss: 0.1234 -
val_mean_absolute_error: 0.2821 - val_mean_squared_error: 0.1234
Epoch 223/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1886 -
mean_absolute_error: 0.3420 - mean_squared_error: 0.1886 - val_loss: 0.0929 -
val_mean_absolute_error: 0.2225 - val_mean_squared_error: 0.0929
Epoch 224/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1839 -
mean_absolute_error: 0.3397 - mean_squared_error: 0.1839 - val_loss: 0.6567 -
val_mean_absolute_error: 0.6478 - val_mean_squared_error: 0.6567
Epoch 225/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1660 -
mean_absolute_error: 0.3166 - mean_squared_error: 0.1660 - val_loss: 0.2630 -
val_mean_absolute_error: 0.4163 - val_mean_squared_error: 0.2630
Epoch 226/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1699 -
mean_absolute_error: 0.3226 - mean_squared_error: 0.1699 - val_loss: 0.3183 -
val_mean_absolute_error: 0.4599 - val_mean_squared_error: 0.3183
Epoch 227/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.1776 -
mean_absolute_error: 0.3339 - mean_squared_error: 0.1776 - val_loss: 0.1970 -
val_mean_absolute_error: 0.3407 - val_mean_squared_error: 0.1970
Epoch 228/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1576 -
mean_absolute_error: 0.3101 - mean_squared_error: 0.1576 - val_loss: 0.0752 -
val_mean_absolute_error: 0.1966 - val_mean_squared_error: 0.0752
Epoch 229/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1503 -
mean_absolute_error: 0.3021 - mean_squared_error: 0.1503 - val_loss: 0.1437 -
val_mean_absolute_error: 0.2878 - val_mean_squared_error: 0.1437
Epoch 230/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1674 -
mean_absolute_error: 0.3181 - mean_squared_error: 0.1674 - val_loss: 0.2741 -
val_mean_absolute_error: 0.4117 - val_mean_squared_error: 0.2741
Epoch 231/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1662 -
mean_absolute_error: 0.3210 - mean_squared_error: 0.1662 - val_loss: 0.1291 -
val_mean_absolute_error: 0.2759 - val_mean_squared_error: 0.1291
Epoch 232/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1585 -
mean_absolute_error: 0.3106 - mean_squared_error: 0.1585 - val_loss: 0.2100 -
val_mean_absolute_error: 0.3970 - val_mean_squared_error: 0.2100
Epoch 233/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1559 -
mean_absolute_error: 0.3086 - mean_squared_error: 0.1559 - val_loss: 0.1185 -
val_mean_absolute_error: 0.2861 - val_mean_squared_error: 0.1185
Epoch 234/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1452 -
mean_absolute_error: 0.2963 - mean_squared_error: 0.1452 - val_loss: 0.2507 -
val_mean_absolute_error: 0.4251 - val_mean_squared_error: 0.2507
Epoch 235/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1641 -
mean_absolute_error: 0.3175 - mean_squared_error: 0.1641 - val_loss: 0.1751 -
val_mean_absolute_error: 0.3228 - val_mean_squared_error: 0.1751
Epoch 236/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1371 -
mean_absolute_error: 0.2816 - mean_squared_error: 0.1371 - val_loss: 0.1149 -
val_mean_absolute_error: 0.2738 - val_mean_squared_error: 0.1149
Epoch 237/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1689 -
mean_absolute_error: 0.3190 - mean_squared_error: 0.1689 - val_loss: 0.0931 -
val_mean_absolute_error: 0.2411 - val_mean_squared_error: 0.0931
Epoch 238/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1386 -
mean_absolute_error: 0.2898 - mean_squared_error: 0.1386 - val_loss: 0.1879 -
val_mean_absolute_error: 0.3274 - val_mean_squared_error: 0.1879
Epoch 239/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.1527 -
mean_absolute_error: 0.3032 - mean_squared_error: 0.1527 - val_loss: 0.2227 -
val_mean_absolute_error: 0.3503 - val_mean_squared_error: 0.2227
Epoch 240/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1657 -
mean_absolute_error: 0.3240 - mean_squared_error: 0.1657 - val_loss: 0.1198 -
val_mean_absolute_error: 0.2975 - val_mean_squared_error: 0.1198
Epoch 241/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1364 -
mean_absolute_error: 0.2877 - mean_squared_error: 0.1364 - val_loss: 0.0950 -
val_mean_absolute_error: 0.2388 - val_mean_squared_error: 0.0950
Epoch 242/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1492 -
mean_absolute_error: 0.2996 - mean_squared_error: 0.1492 - val_loss: 0.1544 -
val_mean_absolute_error: 0.3277 - val_mean_squared_error: 0.1544
Epoch 243/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1275 -
mean_absolute_error: 0.2746 - mean_squared_error: 0.1275 - val_loss: 0.1956 -
val_mean_absolute_error: 0.3775 - val_mean_squared_error: 0.1956
Epoch 244/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1440 -
mean_absolute_error: 0.2942 - mean_squared_error: 0.1440 - val_loss: 0.1743 -
val_mean_absolute_error: 0.3237 - val_mean_squared_error: 0.1743
Epoch 245/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1404 -
mean_absolute_error: 0.2921 - mean_squared_error: 0.1404 - val_loss: 0.1398 -
val_mean_absolute_error: 0.2925 - val_mean_squared_error: 0.1398
Epoch 246/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1477 -
mean_absolute_error: 0.3030 - mean_squared_error: 0.1477 - val_loss: 0.1892 -
val_mean_absolute_error: 0.3752 - val_mean_squared_error: 0.1892
Epoch 247/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1360 -
mean_absolute_error: 0.2861 - mean_squared_error: 0.1360 - val_loss: 0.1938 -
val_mean_absolute_error: 0.3597 - val_mean_squared_error: 0.1938
Epoch 248/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1295 -
mean_absolute_error: 0.2816 - mean_squared_error: 0.1295 - val_loss: 0.1080 -
val_mean_absolute_error: 0.2564 - val_mean_squared_error: 0.1080
Epoch 249/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1242 -
mean_absolute_error: 0.2707 - mean_squared_error: 0.1242 - val_loss: 0.1255 -
val_mean_absolute_error: 0.2954 - val_mean_squared_error: 0.1255
Epoch 250/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1366 -
mean_absolute_error: 0.2845 - mean_squared_error: 0.1366 - val_loss: 0.1284 -
val_mean_absolute_error: 0.3004 - val_mean_squared_error: 0.1284
Epoch 251/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.1276 -
mean_absolute_error: 0.2777 - mean_squared_error: 0.1276 - val_loss: 0.1500 -
val_mean_absolute_error: 0.3047 - val_mean_squared_error: 0.1500
Epoch 252/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1231 -
mean_absolute_error: 0.2703 - mean_squared_error: 0.1231 - val_loss: 0.3724 -
val_mean_absolute_error: 0.5070 - val_mean_squared_error: 0.3724
Epoch 253/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1426 -
mean_absolute_error: 0.2958 - mean_squared_error: 0.1426 - val_loss: 0.0859 -
val_mean_absolute_error: 0.2141 - val_mean_squared_error: 0.0859
Epoch 254/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1195 -
mean_absolute_error: 0.2623 - mean_squared_error: 0.1195 - val_loss: 0.1233 -
val_mean_absolute_error: 0.2849 - val_mean_squared_error: 0.1233
Epoch 255/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1272 -
mean_absolute_error: 0.2796 - mean_squared_error: 0.1272 - val_loss: 0.2445 -
val_mean_absolute_error: 0.4005 - val_mean_squared_error: 0.2445
Epoch 256/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1191 -
mean_absolute_error: 0.2685 - mean_squared_error: 0.1191 - val_loss: 0.0531 -
val_mean_absolute_error: 0.1772 - val_mean_squared_error: 0.0531
Epoch 257/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1115 -
mean_absolute_error: 0.2518 - mean_squared_error: 0.1115 - val_loss: 0.2598 -
val_mean_absolute_error: 0.4370 - val_mean_squared_error: 0.2598
Epoch 258/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1145 -
mean_absolute_error: 0.2588 - mean_squared_error: 0.1145 - val_loss: 0.1184 -
val_mean_absolute_error: 0.2665 - val_mean_squared_error: 0.1184
Epoch 259/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1121 -
mean_absolute_error: 0.2583 - mean_squared_error: 0.1121 - val_loss: 0.0603 -
val_mean_absolute_error: 0.1780 - val_mean_squared_error: 0.0603
Epoch 260/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1179 -
mean_absolute_error: 0.2691 - mean_squared_error: 0.1179 - val_loss: 0.1802 -
val_mean_absolute_error: 0.3401 - val_mean_squared_error: 0.1802
Epoch 261/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1146 -
mean_absolute_error: 0.2612 - mean_squared_error: 0.1146 - val_loss: 0.0658 -
val_mean_absolute_error: 0.1842 - val_mean_squared_error: 0.0658
Epoch 262/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1111 -
mean_absolute_error: 0.2554 - mean_squared_error: 0.1111 - val_loss: 0.0551 -
val_mean_absolute_error: 0.1650 - val_mean_squared_error: 0.0551
Epoch 263/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.1069 -
mean_absolute_error: 0.2513 - mean_squared_error: 0.1069 - val_loss: 0.0438 -
val_mean_absolute_error: 0.1608 - val_mean_squared_error: 0.0438
Epoch 264/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1005 -
mean_absolute_error: 0.2428 - mean_squared_error: 0.1005 - val_loss: 0.0420 -
val_mean_absolute_error: 0.1470 - val_mean_squared_error: 0.0420
Epoch 265/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1141 -
mean_absolute_error: 0.2604 - mean_squared_error: 0.1141 - val_loss: 0.0599 -
val_mean_absolute_error: 0.1783 - val_mean_squared_error: 0.0599
Epoch 266/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1084 -
mean_absolute_error: 0.2551 - mean_squared_error: 0.1084 - val_loss: 0.2232 -
val_mean_absolute_error: 0.4033 - val_mean_squared_error: 0.2232
Epoch 267/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1120 -
mean_absolute_error: 0.2551 - mean_squared_error: 0.1120 - val_loss: 0.2587 -
val_mean_absolute_error: 0.4150 - val_mean_squared_error: 0.2587
Epoch 268/350
250/250 [==============================] - 1s 3ms/step - loss: 0.1105 -
mean_absolute_error: 0.2585 - mean_squared_error: 0.1105 - val_loss: 0.0465 -
val_mean_absolute_error: 0.1636 - val_mean_squared_error: 0.0465
Epoch 269/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0972 -
mean_absolute_error: 0.2385 - mean_squared_error: 0.0972 - val_loss: 0.0545 -
val_mean_absolute_error: 0.1798 - val_mean_squared_error: 0.0545
Epoch 270/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1063 -
mean_absolute_error: 0.2542 - mean_squared_error: 0.1063 - val_loss: 0.1470 -
val_mean_absolute_error: 0.3173 - val_mean_squared_error: 0.1470
Epoch 271/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0872 -
mean_absolute_error: 0.2203 - mean_squared_error: 0.0872 - val_loss: 0.0498 -
val_mean_absolute_error: 0.1760 - val_mean_squared_error: 0.0498
Epoch 272/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0959 -
mean_absolute_error: 0.2379 - mean_squared_error: 0.0959 - val_loss: 0.0563 -
val_mean_absolute_error: 0.1875 - val_mean_squared_error: 0.0563
Epoch 273/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1098 -
mean_absolute_error: 0.2562 - mean_squared_error: 0.1098 - val_loss: 0.0569 -
val_mean_absolute_error: 0.1775 - val_mean_squared_error: 0.0569
Epoch 274/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0865 -
mean_absolute_error: 0.2203 - mean_squared_error: 0.0865 - val_loss: 0.0425 -
val_mean_absolute_error: 0.1590 - val_mean_squared_error: 0.0425
Epoch 275/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.0912 -
mean_absolute_error: 0.2313 - mean_squared_error: 0.0912 - val_loss: 0.2814 -
val_mean_absolute_error: 0.4481 - val_mean_squared_error: 0.2814
Epoch 276/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1050 -
mean_absolute_error: 0.2491 - mean_squared_error: 0.1050 - val_loss: 0.0433 -
val_mean_absolute_error: 0.1594 - val_mean_squared_error: 0.0433
Epoch 277/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1096 -
mean_absolute_error: 0.2550 - mean_squared_error: 0.1096 - val_loss: 0.0581 -
val_mean_absolute_error: 0.2016 - val_mean_squared_error: 0.0581
Epoch 278/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0974 -
mean_absolute_error: 0.2459 - mean_squared_error: 0.0974 - val_loss: 0.3449 -
val_mean_absolute_error: 0.4787 - val_mean_squared_error: 0.3449
Epoch 279/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0896 -
mean_absolute_error: 0.2304 - mean_squared_error: 0.0896 - val_loss: 0.2124 -
val_mean_absolute_error: 0.3703 - val_mean_squared_error: 0.2124
Epoch 280/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0963 -
mean_absolute_error: 0.2378 - mean_squared_error: 0.0963 - val_loss: 0.0453 -
val_mean_absolute_error: 0.1685 - val_mean_squared_error: 0.0453
Epoch 281/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0904 -
mean_absolute_error: 0.2320 - mean_squared_error: 0.0904 - val_loss: 0.1115 -
val_mean_absolute_error: 0.2647 - val_mean_squared_error: 0.1115
Epoch 282/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0845 -
mean_absolute_error: 0.2239 - mean_squared_error: 0.0845 - val_loss: 0.0472 -
val_mean_absolute_error: 0.1593 - val_mean_squared_error: 0.0472
Epoch 283/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0960 -
mean_absolute_error: 0.2401 - mean_squared_error: 0.0960 - val_loss: 0.1075 -
val_mean_absolute_error: 0.2655 - val_mean_squared_error: 0.1075
Epoch 284/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0760 -
mean_absolute_error: 0.2077 - mean_squared_error: 0.0760 - val_loss: 0.0431 -
val_mean_absolute_error: 0.1548 - val_mean_squared_error: 0.0431
Epoch 285/350
250/250 [==============================] - 1s 4ms/step - loss: 0.1033 -
mean_absolute_error: 0.2458 - mean_squared_error: 0.1033 - val_loss: 0.0869 -
val_mean_absolute_error: 0.2470 - val_mean_squared_error: 0.0869
Epoch 286/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0888 -
mean_absolute_error: 0.2298 - mean_squared_error: 0.0888 - val_loss: 0.0880 -
val_mean_absolute_error: 0.2300 - val_mean_squared_error: 0.0880
Epoch 287/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.0962 -
mean_absolute_error: 0.2449 - mean_squared_error: 0.0962 - val_loss: 0.1346 -
val_mean_absolute_error: 0.2971 - val_mean_squared_error: 0.1346
Epoch 288/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0863 -
mean_absolute_error: 0.2237 - mean_squared_error: 0.0863 - val_loss: 0.0297 -
val_mean_absolute_error: 0.1250 - val_mean_squared_error: 0.0297
Epoch 289/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0952 -
mean_absolute_error: 0.2406 - mean_squared_error: 0.0952 - val_loss: 0.0640 -
val_mean_absolute_error: 0.2028 - val_mean_squared_error: 0.0640
Epoch 290/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0748 -
mean_absolute_error: 0.2082 - mean_squared_error: 0.0748 - val_loss: 0.2313 -
val_mean_absolute_error: 0.4067 - val_mean_squared_error: 0.2313
Epoch 291/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0815 -
mean_absolute_error: 0.2175 - mean_squared_error: 0.0815 - val_loss: 0.0373 -
val_mean_absolute_error: 0.1503 - val_mean_squared_error: 0.0373
Epoch 292/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0865 -
mean_absolute_error: 0.2222 - mean_squared_error: 0.0865 - val_loss: 0.0794 -
val_mean_absolute_error: 0.2420 - val_mean_squared_error: 0.0794
Epoch 293/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0697 -
mean_absolute_error: 0.2010 - mean_squared_error: 0.0697 - val_loss: 0.0507 -
val_mean_absolute_error: 0.1675 - val_mean_squared_error: 0.0507
Epoch 294/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0850 -
mean_absolute_error: 0.2252 - mean_squared_error: 0.0850 - val_loss: 0.1937 -
val_mean_absolute_error: 0.3524 - val_mean_squared_error: 0.1937
Epoch 295/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0818 -
mean_absolute_error: 0.2143 - mean_squared_error: 0.0818 - val_loss: 0.0340 -
val_mean_absolute_error: 0.1339 - val_mean_squared_error: 0.0340
Epoch 296/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0686 -
mean_absolute_error: 0.1984 - mean_squared_error: 0.0686 - val_loss: 0.0841 -
val_mean_absolute_error: 0.2282 - val_mean_squared_error: 0.0841
Epoch 297/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0869 -
mean_absolute_error: 0.2269 - mean_squared_error: 0.0869 - val_loss: 0.0326 -
val_mean_absolute_error: 0.1351 - val_mean_squared_error: 0.0326
Epoch 298/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0664 -
mean_absolute_error: 0.1942 - mean_squared_error: 0.0664 - val_loss: 0.1131 -
val_mean_absolute_error: 0.2804 - val_mean_squared_error: 0.1131
Epoch 299/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0744 -
mean_absolute_error: 0.2060 - mean_squared_error: 0.0744 - val_loss: 0.0913 -
val_mean_absolute_error: 0.2479 - val_mean_squared_error: 0.0913
Epoch 300/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0840 -
mean_absolute_error: 0.2267 - mean_squared_error: 0.0840 - val_loss: 0.1185 -
val_mean_absolute_error: 0.2674 - val_mean_squared_error: 0.1185
Epoch 301/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0795 -
mean_absolute_error: 0.2147 - mean_squared_error: 0.0795 - val_loss: 0.1355 -
val_mean_absolute_error: 0.3047 - val_mean_squared_error: 0.1355
Epoch 302/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0708 -
mean_absolute_error: 0.2041 - mean_squared_error: 0.0708 - val_loss: 0.0266 -
val_mean_absolute_error: 0.1179 - val_mean_squared_error: 0.0266
Epoch 303/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0821 -
mean_absolute_error: 0.2208 - mean_squared_error: 0.0821 - val_loss: 0.1060 -
val_mean_absolute_error: 0.2637 - val_mean_squared_error: 0.1060
Epoch 304/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0844 -
mean_absolute_error: 0.2265 - mean_squared_error: 0.0844 - val_loss: 0.1155 -
val_mean_absolute_error: 0.2624 - val_mean_squared_error: 0.1155
Epoch 305/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0783 -
mean_absolute_error: 0.2117 - mean_squared_error: 0.0783 - val_loss: 0.0381 -
val_mean_absolute_error: 0.1559 - val_mean_squared_error: 0.0381
Epoch 306/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0686 -
mean_absolute_error: 0.1956 - mean_squared_error: 0.0686 - val_loss: 0.0485 -
val_mean_absolute_error: 0.1643 - val_mean_squared_error: 0.0485
Epoch 307/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0638 -
mean_absolute_error: 0.1906 - mean_squared_error: 0.0638 - val_loss: 0.1348 -
val_mean_absolute_error: 0.3063 - val_mean_squared_error: 0.1348
Epoch 308/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0734 -
mean_absolute_error: 0.2036 - mean_squared_error: 0.0734 - val_loss: 0.0339 -
val_mean_absolute_error: 0.1335 - val_mean_squared_error: 0.0339
Epoch 309/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0814 -
mean_absolute_error: 0.2188 - mean_squared_error: 0.0814 - val_loss: 0.0358 -
val_mean_absolute_error: 0.1490 - val_mean_squared_error: 0.0358
Epoch 310/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0672 -
mean_absolute_error: 0.1935 - mean_squared_error: 0.0672 - val_loss: 0.2091 -
val_mean_absolute_error: 0.3668 - val_mean_squared_error: 0.2091
Epoch 311/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0578 -
mean_absolute_error: 0.1783 - mean_squared_error: 0.0578 - val_loss: 0.0829 -
val_mean_absolute_error: 0.2214 - val_mean_squared_error: 0.0829
Epoch 312/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0596 -
mean_absolute_error: 0.1807 - mean_squared_error: 0.0596 - val_loss: 0.0738 -
val_mean_absolute_error: 0.2237 - val_mean_squared_error: 0.0738
Epoch 313/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0699 -
mean_absolute_error: 0.1997 - mean_squared_error: 0.0699 - val_loss: 0.0767 -
val_mean_absolute_error: 0.2154 - val_mean_squared_error: 0.0767
Epoch 314/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0645 -
mean_absolute_error: 0.1943 - mean_squared_error: 0.0645 - val_loss: 0.0706 -
val_mean_absolute_error: 0.1975 - val_mean_squared_error: 0.0706
Epoch 315/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0595 -
mean_absolute_error: 0.1797 - mean_squared_error: 0.0595 - val_loss: 0.0390 -
val_mean_absolute_error: 0.1493 - val_mean_squared_error: 0.0390
Epoch 316/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0722 -
mean_absolute_error: 0.1970 - mean_squared_error: 0.0722 - val_loss: 0.0278 -
val_mean_absolute_error: 0.1182 - val_mean_squared_error: 0.0278
Epoch 317/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0653 -
mean_absolute_error: 0.1921 - mean_squared_error: 0.0653 - val_loss: 0.0279 -
val_mean_absolute_error: 0.1203 - val_mean_squared_error: 0.0279
Epoch 318/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0548 -
mean_absolute_error: 0.1774 - mean_squared_error: 0.0548 - val_loss: 0.0829 -
val_mean_absolute_error: 0.2287 - val_mean_squared_error: 0.0829
Epoch 319/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0655 -
mean_absolute_error: 0.1922 - mean_squared_error: 0.0655 - val_loss: 0.0218 -
val_mean_absolute_error: 0.1134 - val_mean_squared_error: 0.0218
Epoch 320/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0531 -
mean_absolute_error: 0.1721 - mean_squared_error: 0.0531 - val_loss: 0.0194 -
val_mean_absolute_error: 0.1003 - val_mean_squared_error: 0.0194
Epoch 321/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0642 -
mean_absolute_error: 0.1963 - mean_squared_error: 0.0642 - val_loss: 0.1481 -
val_mean_absolute_error: 0.3179 - val_mean_squared_error: 0.1481
Epoch 322/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0530 -
mean_absolute_error: 0.1711 - mean_squared_error: 0.0530 - val_loss: 0.0561 -
val_mean_absolute_error: 0.1927 - val_mean_squared_error: 0.0561
Epoch 323/350
```

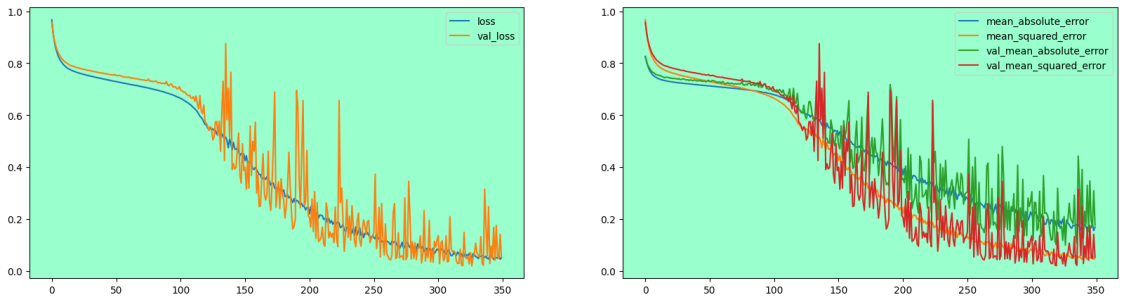
```
250/250 [==============================] - 1s 3ms/step - loss: 0.0622 -
mean_absolute_error: 0.1921 - mean_squared_error: 0.0622 - val_loss: 0.0473 -
val_mean_absolute_error: 0.1634 - val_mean_squared_error: 0.0473
Epoch 324/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0579 -
mean_absolute_error: 0.1805 - mean_squared_error: 0.0579 - val_loss: 0.0557 -
val_mean_absolute_error: 0.1722 - val_mean_squared_error: 0.0557
Epoch 325/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0667 -
mean_absolute_error: 0.1991 - mean_squared_error: 0.0667 - val_loss: 0.0266 -
val_mean_absolute_error: 0.1171 - val_mean_squared_error: 0.0266
Epoch 326/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0572 -
mean_absolute_error: 0.1768 - mean_squared_error: 0.0572 - val_loss: 0.0700 -
val_mean_absolute_error: 0.2087 - val_mean_squared_error: 0.0700
Epoch 327/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0506 -
mean_absolute_error: 0.1600 - mean_squared_error: 0.0506 - val_loss: 0.0184 -
val_mean_absolute_error: 0.0985 - val_mean_squared_error: 0.0184
Epoch 328/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0661 -
mean_absolute_error: 0.1957 - mean_squared_error: 0.0661 - val_loss: 0.0515 -
val_mean_absolute_error: 0.1831 - val_mean_squared_error: 0.0515
Epoch 329/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0612 -
mean_absolute_error: 0.1835 - mean_squared_error: 0.0612 - val_loss: 0.0687 -
val_mean_absolute_error: 0.1984 - val_mean_squared_error: 0.0687
Epoch 330/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0545 -
mean_absolute_error: 0.1777 - mean_squared_error: 0.0545 - val_loss: 0.1063 -
val_mean_absolute_error: 0.2599 - val_mean_squared_error: 0.1063
Epoch 331/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0527 -
mean_absolute_error: 0.1670 - mean_squared_error: 0.0527 - val_loss: 0.0688 -
val_mean_absolute_error: 0.2035 - val_mean_squared_error: 0.0688
Epoch 332/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0472 -
mean_absolute_error: 0.1639 - mean_squared_error: 0.0472 - val_loss: 0.0584 -
val_mean_absolute_error: 0.1909 - val_mean_squared_error: 0.0584
Epoch 333/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0559 -
mean_absolute_error: 0.1761 - mean_squared_error: 0.0559 - val_loss: 0.0485 -
val_mean_absolute_error: 0.1820 - val_mean_squared_error: 0.0485
Epoch 334/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0554 -
mean_absolute_error: 0.1730 - mean_squared_error: 0.0554 - val_loss: 0.1315 -
val_mean_absolute_error: 0.2916 - val_mean_squared_error: 0.1315
Epoch 335/350
```

```
250/250 [==============================] - 1s 4ms/step - loss: 0.0586 -
mean_absolute_error: 0.1743 - mean_squared_error: 0.0586 - val_loss: 0.0292 -
val_mean_absolute_error: 0.1334 - val_mean_squared_error: 0.0292
Epoch 336/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0574 -
mean_absolute_error: 0.1794 - mean_squared_error: 0.0574 - val_loss: 0.0207 -
val_mean_absolute_error: 0.1043 - val_mean_squared_error: 0.0207
Epoch 337/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0513 -
mean_absolute_error: 0.1723 - mean_squared_error: 0.0513 - val_loss: 0.3146 -
val_mean_absolute_error: 0.4416 - val_mean_squared_error: 0.3146
Epoch 338/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0557 -
mean_absolute_error: 0.1798 - mean_squared_error: 0.0557 - val_loss: 0.1674 -
val_mean_absolute_error: 0.3248 - val_mean_squared_error: 0.1674
Epoch 339/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0472 -
mean_absolute_error: 0.1553 - mean_squared_error: 0.0472 - val_loss: 0.0578 -
val_mean_absolute_error: 0.1810 - val_mean_squared_error: 0.0578
Epoch 340/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0453 -
mean_absolute_error: 0.1601 - mean_squared_error: 0.0453 - val_loss: 0.2474 -
val_mean_absolute_error: 0.3908 - val_mean_squared_error: 0.2474
Epoch 341/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0584 -
mean_absolute_error: 0.1850 - mean_squared_error: 0.0584 - val_loss: 0.0276 -
val_mean_absolute_error: 0.1244 - val_mean_squared_error: 0.0276
Epoch 342/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0575 -
mean_absolute_error: 0.1780 - mean_squared_error: 0.0575 - val_loss: 0.0956 -
val_mean_absolute_error: 0.2293 - val_mean_squared_error: 0.0956
Epoch 343/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0514 -
mean_absolute_error: 0.1698 - mean_squared_error: 0.0514 - val_loss: 0.0503 -
val_mean_absolute_error: 0.1794 - val_mean_squared_error: 0.0503
Epoch 344/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0437 -
mean_absolute_error: 0.1538 - mean_squared_error: 0.0437 - val_loss: 0.1672 -
val_mean_absolute_error: 0.3292 - val_mean_squared_error: 0.1672
Epoch 345/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0513 -
mean_absolute_error: 0.1700 - mean_squared_error: 0.0513 - val_loss: 0.0406 -
val_mean_absolute_error: 0.1509 - val_mean_squared_error: 0.0406
Epoch 346/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0506 -
mean_absolute_error: 0.1652 - mean_squared_error: 0.0506 - val_loss: 0.1745 -
val_mean_absolute_error: 0.3466 - val_mean_squared_error: 0.1745
Epoch 347/350
```

```
250/250 [==============================] - 1s 3ms/step - loss: 0.0487 -
mean_absolute_error: 0.1665 - mean_squared_error: 0.0487 - val_loss: 0.0540 -
val_mean_absolute_error: 0.1936 - val_mean_squared_error: 0.0540
Epoch 348/350
250/250 [==============================] - 1s 4ms/step - loss: 0.0599 -
mean_absolute_error: 0.1798 - mean_squared_error: 0.0599 - val_loss: 0.0470 -
val_mean_absolute_error: 0.1720 - val_mean_squared_error: 0.0470
Epoch 349/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0444 -
mean_absolute_error: 0.1546 - mean_squared_error: 0.0444 - val_loss: 0.1405 -
val_mean_absolute_error: 0.3086 - val_mean_squared_error: 0.1405
Epoch 350/350
250/250 [==============================] - 1s 3ms/step - loss: 0.0496 -
mean_absolute_error: 0.1680 - mean_squared_error: 0.0496 - val_loss: 0.0528 -
val_mean_absolute_error: 0.1761 - val_mean_squared_error: 0.0528
```

[104]:
```
fig,axs=plt.subplots(1,2,figsize=(20,5))

axs[0].plot(his['loss'])
axs[0].plot(his['val_loss'])
axs[0].legend(['loss','val_loss'])


axs[1].plot(his['mean_absolute_error'])
axs[1].plot(his['mean_squared_error'])
axs[1].plot(his['val_mean_absolute_error'])
axs[1].plot(his['val_mean_squared_error'])
axs[1].
 ↪legend(['mean_absolute_error','mean_squared_error','val_mean_absolute_error','val_mean_squa
```

[104]:
```
<matplotlib.legend.Legend at 0x7c5e8923f190>
```



[105]:
```
ypred=regNN.predict(xtest)
print(f"MSE: {mean_squared_error(ytest,ypred)}")
print(f"MAE: {mean_absolute_error(ytest,ypred)}")
print(f"R^2: {round(r2_score(ytest,ypred)*100,2)}%")
```

```
213/213 [==============================] - 0s 1ms/step
MSE: 0.05225817538849632
MAE: 0.1759325793284671
R^2: 94.75%
```

[48]:
```
regNN.save('regNN.h5')
```

18 *ARD REGRESSOR*

[187]:
```
from sklearn.linear_model import ARDRegression
```

[190]:
```
a=ARDRegression()
gg={'alpha_1':[1e-3,1e-5,1e-7,1e-9],
   'alpha_2':[1e-3,1e-5,1e-7],
   'lambda_1':[1e-1,1e-3,1e-5,1e-7],
   'n_iter':[100,200,300],
   'lambda_2':[1e-3,1e-5,1e-7,1e-9],
   'tol':[1e-3,1e-5,1e-7,1e-9]}
p(gg,a,'ardreg')
```

```
ARDRegression()

Best Parameters {'tol': 1e-07, 'n_iter': 300, 'lambda_2': 1e-07, 'lambda_1':
1e-07, 'alpha_2': 0.001, 'alpha_1': 0.001}
Best Score 22.66 %
```

[190]:

	Metric_Name	Values
0	R.sq(%)	23.160
1	Mean-Squared-Error	0.756
2	Rooted-Mean-Squared-Error	0.850
3	Mean-Absoulute-Error	0.723
4	Median-Absoulte-Error	0.666

19 CONCULSION:

ML MODEL FOR RECOMENDATION BASED ON ITS PERFORMANCES

- Adaptive Boosting Regressor
- Hist Gradient Boost Regressor
- Bagging Regressor
- Radius Neighbors Regressor
- Artificial Neural Network - Multiple Linear Regression Model

- We need more data and also diversified features which can able to capture the hidden patterns and trends among the customer behaviours purchasing in the store.