Task 2 - Diminos Case Study

Diminos Store - Delivery Time

Problem Statement 🍕

Kanav has started his own Pizza Store by getting the Franchise from the popular Pizza brand Diminos.
Diminos promises to deliver the pizza order within 31 minutes from the time the order was placed. Otherwise the pizza will be free for the customer.
In order to increase the revenue and profits Kanav is running the store 24 * 7. Recently Diminos gave a notice to Kanav that they will be measuring their stores' performance by looking at the metric - which is 95th Percentile on Order Delivery time should be less than 31 mins.
Kanav is worried that he might lose the franchise if he is not able to meet the metric and wants your help in order to understand his store's performance so that he can take some actions to prevent his business.

TASK

Assume that you are a freelance data scientist.

Help Kanav by analyzing the data and sharing insights to keep his business up and running.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the data

Load the required libraries

```
df = pd.read_csv(r"/content/diminos_data.csv")
```

View the data

```
df.head()
```

	order_id	order_placed_at	order_delivered_at
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385
3	1523114	2023-03-01 00:07:47	2023-03-01 00:46:19.019399
4	1523115	2023-03-01 00:09:03	2023-03-01 00:25:13.619056

Information about data

```
# basic info

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   order_id            15000 non-null  int64 
 1   order_placed_at     15000 non-null  object
 2   order_delivered_at  15000 non-null  object
dtypes: int64(1), object(2)
memory usage: 351.7+ KB
```

```
## Describe the data

df.describe()
```

	order_id
count	1.500000e+04
mean	1.530610e+06
std	4.330271e+03
min	1.523111e+06
25%	1.526861e+06
50%	1.530610e+06
75%	1.534360e+06
max	1.538110e+06

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%
order_id	15000.0	1530610.5	4330.271354	1523111.0	1526860.75	1530610.5	1534360.25

Check columns

```
df.columns
```

```
Index(['order_id', 'order_placed_at', 'order_delivered_at'], dtype='object')
```

Duplicated values

```
# check the duplicates

df.duplicated().sum()
```

```
0
```

```
df.columns
```

```
Index(['order_id', 'order_placed_at', 'order_delivered_at'], dtype='object')
```

Unique values in data

```
df['order_id'].nunique()
```

```
15000
```

```
df['order_placed_at'].nunique()
```

```
14953
```

```
df['order_delivered_at'].nunique()
```

```
15000
```

Find the null values

```
df.isnull().sum()
```

```
order_id             0
order_placed_at      0
order_delivered_at   0
dtype: int64
```

```
df.isnull().sum().sum()
```

```
0
```

Check datatypes

```
df.dtypes
```

```
order_id              int64
order_placed_at      object
order_delivered_at   object
dtype: object
```

Check Shape

```
df.shape
```

```
(15000, 3)
```

```
df.head(2).info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2 entries, 0 to 1
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   order_id            2 non-null      int64 
 1   order_placed_at     2 non-null      object
 2   order_delivered_at  2 non-null      object
dtypes: int64(1), object(2)
memory usage: 176.0+ bytes
```

```
df.head(3)
```

	order_id	order_placed_at	order_delivered_at
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385

Report:

datatype is given wrong

order_placed_at is datetime but showing object so we have to change both columns datetime

```
import datetime
```

```
df['order_placed_at'] = pd.to_datetime(df['order_placed_at'])
df['order_delivered_at'] = pd.to_datetime(df['order_delivered_at'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype         
---  ------              --------------  -----         
 0   order_id            15000 non-null  int64         
 1   order_placed_at     15000 non-null  datetime64[ns]
 2   order_delivered_at  15000 non-null  datetime64[ns]
dtypes: datetime64[ns](2), int64(1)
memory usage: 351.7 KB
```

Check First Order and Last Order

```
print("First Order :", min(df['order_placed_at']))
print("Last Order :", max(df['order_placed_at']))
```

```
First Order : 2023-03-01 00:00:59
Last Order : 2023-03-27 23:58:20
```

Check Fist Delivery and Last Delivery

```
print("First Delivery :", min(df['order_delivered_at']))
print("Last Delivery :", max(df['order_delivered_at']))
```

```
First Delivery : 2023-03-01 00:18:07.443132
Last Delivery : 2023-03-29 02:42:50.645252
```

Substracting time from order deliverey time to order placed time

```
df['deliverey_time'] = df['order_delivered_at'] -  df['order_placed_at']
```

```
df.head()
```

	order_id	order_placed_at	order_delivered_at	deliverey_time
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132	0 days 00:17:08.443132
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241	0 days 00:15:35.925241
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385	0 days 00:15:06.291385
3	1523114	2023-03-01 00:07:47	2023-03-01 00:46:19.019399	0 days 00:38:32.019399
4	1523115	2023-03-01 00:09:03	2023-03-01 00:25:13.619056	0 days 00:16:10.619056

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype          
---  ------              --------------  -----          
 0   order_id            15000 non-null  int64          
 1   order_placed_at     15000 non-null  datetime64[ns] 
 2   order_delivered_at  15000 non-null  datetime64[ns] 
 3   deliverey_time      15000 non-null  timedelta64[ns]
dtypes: datetime64[ns](2), int64(1), timedelta64[ns](1)
memory usage: 468.9 KB
```

```
df["deliverey_time"] = df["deliverey_time"].dt.total_seconds()/60
```

```
df.head()
```

	order_id	order_placed_at	order_delivered_at	deliverey_time
0	1523111	2023-03-01 00:00:59	2023-03-01 00:18:07.443132	17.140719
1	1523112	2023-03-01 00:03:59	2023-03-01 00:19:34.925241	15.598754
2	1523113	2023-03-01 00:07:22	2023-03-01 00:22:28.291385	15.104856
3	1523114	2023-03-01 00:07:47	2023-03-01 00:46:19.019399	38.533657
4	1523115	2023-03-01 00:09:03	2023-03-01 00:25:13.619056	16.176984

```
df['deliverey_time'].quantile(0.95)
```

```
27.261043996666658
```

```
_95th_percentile = round(df['deliverey_time'].quantile(0.95))
```

```
_95th_percentile
```

```
27
```

```
if _95th_percentile  > 31:
    print("Kanav is worried that he might lose the franchise if he is not able to meet the metric so that he can take some actions to pre
```

```
else:
    print("the store performance is good and the owner no need to worry")
```

```
 the store performance is good and the owner no need to worry
```

✓ 0s completed at 1:21 PM