```
import numpy as np
import pandas as pd
from matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.tree import DecisionTree
import sklearn.metrics as metrics
```

```
data = pd.read_csv("C:\Users\rishav\Desktop\cinemaTicket.csv")
```

```
data.head()
```

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occupancy	total_price	total_pay	capacity	date	month	quarter	day	year
0	1492	29	2860000	22	0	7	4	13.00	1000000.0	20	07/5/2018	07	3	5	2018
1	1492	304	2860000	42	0	3	8.00	980000.0	42	07/06/2018	07	3	2	6	2018
2	1492	469	2500000	32	0	4	20.00	800000.0	32	10/10/2018	10	4	2	3	2018
3	1492	429	1200000	12	0	7	12.0	1000000.0	12	10/28/2018	10	4	2	5	2018
4	1492	524	5350000	15	0	1	16.67	800000.0	15	04/30/2018	04	2	2	1	2018

```
data.tail()
```

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occupancy	total_price	total_pay	capacity	date	month	quarter	day
142616	1569	456	1320000	22	0	2	3.66	600000.0	33	06/04/2017	2017-6-24	11	4	4
142617	1569	479	1200000	18	0	7	30.12	800000.0	18	01/08/2018	2018-1-8	11	4	1
142621	1569	524	1080000	9	0	3	9.20	1320000.0	9	06/08/2017	2017-6-11	11	4	4
142622	1569	529	800000	8	0	7	3.00	1200000.0	8	10/10/2018	2018-10-14	11	4	3
142625	1569	466	200000	5	0	1	1.79	500000.0	5	07/13/2018	2018-7-13	11	4	3

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142524 entries, 0 to 142524
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   film_code    142524 non-null  int64
 1   cinema_code  142524 non-null  int64
 2   total_sales  142524 non-null  int64
 3   tickets_sold 142524 non-null  int64
 4   tickets_out  142524 non-null  int64
 5   show_time    142524 non-null  int64
 6   occupancy    142524 non-null  int64
 7   occupancy    142524 non-null  int64
 8   total_price  142524 non-null  int64
 9   total_pay    142524 non-null  int64
 10  capacity     142524 non-null  int64
 11  date         142524 non-null  object
 12  month        142524 non-null  int64
 13  quarter      142524 non-null  int64
 14  day          142524 non-null  int64
dtype: int64
memory usage: 16.3 MB
```

```
data.dtypes(inplace=True)
data['date'] = pd.DatetimeIndex(data['date'])
data['year'] = pd.DatetimeIndex(data['date'])
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142524 entries, 0 to 142524
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   film_code    142524 non-null  int64
 1   cinema_code  142524 non-null  int64
 2   total_sales  142524 non-null  int64
 3   tickets_sold 142524 non-null  int64
 4   tickets_out  142524 non-null  int64
 5   show_time    142524 non-null  int64
 6   occupancy    142524 non-null  int64
 7   occupancy    142524 non-null  int64
 8   total_price  142524 non-null  int64
 9   total_pay    142524 non-null  int64
 10  capacity     142524 non-null  int64
 11  date         142524 non-null  datetime64[ns]
 12  month        142524 non-null  int64
 13  quarter      142524 non-null  int64
 14  day          142524 non-null  int64
dtype: int64
memory usage: 16.3 MB
```

```
data.shape
```

```
(142524, 14)
```

```
data.isnull()*100/data.shape[0]
```

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occupancy	total_price	total_pay	capacity	date	month	quarter	day	year
0000	1038	101	800000	13	0	3	hash 8920E7BB3F	13	hash 2018-05-31	0	2018-05-31	0	2	31	2018
0770	1038	314	2130000	30	0	4	hash 8000000000	30	hash 2018-08-31	0	2018-08-31	0	3	4	2018
0000	1038	477	800000	22	0	6	hash 3000000000	22	hash 2018-08-07	0	2018-08-07	0	3	7	2018
2180D	1071	275	200000	5	0	1	hash 5000000000	5	hash 2018-06-11	0	2018-06-11	0	2	11	2018
2180D	1071	320	100000	1	0	1	hash 5000000000	1	hash 2018-06-30	0	2018-06-30	0	2	30	2018
1A007	1092	340	300000	6	0	1	hash 5000000000	6	hash 2018-07	0	2018-07-1	0	3	1	2018
1B4D8	1092	340	100000	3	0	1	hash 5000000000	3	hash 2018-06-30	0	2018-06-30	0	3	30	2018
1B8D9	1092	340	100000	1	0	1	hash 5000000000	2	hash 2018-06-01	0	2018-06-01	0	3	1	2018
1B7D3	1092	340	500000	10	0	1	hash 5000000000	10	hash 2018-06-24	0	2018-06-24	0	3	24	2018
1B8D9	1092	341	3600000	36	0	1	hash 8000120000	36	hash 2018-06-23	0	2018-06-23	0	3	23	2018

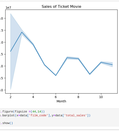
128 rows × 15 columns

```
data['tomorrow'] = data['tomorrow'].fillna(0)
data['this_price'] = data['this_price'].fillna(0)
```

```
print(data.nunique())
```

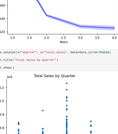
```
film_code        49
cinema_code     204
total_sales     1362
tickets_sold     202
tickets_out        1
show_time         15
occupancy         35
total_price      1936
total_pay        1362
capacity         1362
date              365
month              12
quarter             4
day                31
year                1
week              365
weekDay             7
dtype: int64
```

```
sns.lineplot(x='month', y='occupancy', data=data)
plt.title('Sales of Ticket Movie')
plt.xlabel('Month')
plt.ylabel('Total of Sale')
plt.show()
```

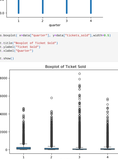



```
px.figure(figsize=(10,14))
sns.barplot(x=data['film_code'], y=data['total_sales'])
plt.show()
```

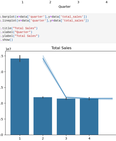
```
px.lineplot(data=data, y='quarter', y='ticket_sold', color='Q')
plt.title("Tickets Sold by Quarter")
plt.xlabel("Month")
plt.ylabel("Occupancy")
plt.show()
```



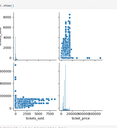

```
px.catplot(x='quarter', y='total_sales', data=data,jitter=False)
plt.title("Total Sales by Quarter")
plt.show()
```



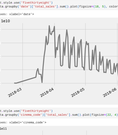

```
sns.boxplot(x =data['quarter'], y =data['tickets_sold'],width=0.5)
plt.title("Boxplot of Ticket Sold")
plt.xlabel("Ticket Sold")
plt.ylabel("quarter")
plt.show()
```




```
px.barplot(data=data, x='quarter', y=data['total_sales'])
px.lineplot(data=data, x='quarter', y=data['total_sales'])
px.title("Total Sales")
px.xlabel("Quarter")
px.ylabel("Total Sales")
px.show()
```



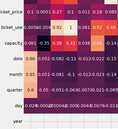

```
px.pairplot(data=data, vars=['tickets_sold', 'ticket_price'])
px.show()
```




```
px.lineplot(data=data, x='date', y=data['total_sales'])
px.title("Total Sales")
px.xlabel("Date")
px.ylabel("Total Sales")
px.show()
```

```
px.style.use('fivethirtyeight')
px.groupby('cinema_code')['total_sales'].sum().plot(figsize=(15, 4), color='grey')
px.show()
```

```
px.style.use('fivethirtyeight')
px.groupby('film_code')['total_sales'].sum().plot(figsize=(15, 4), color='grey')
px.show()
```



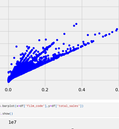

```
px.style.use('fivethirtyeight')
px.bar(data=data, x='film_code', y=data['total_sales'], color='grey')
px.show()
px.figure(figsize=(8, 6))
px.boxplot(data=data, x='film_code', y=data['total_sales'], palette='grey')
px.show()
```

```
C:\Users\rishav\Appdata\Local\Temp\ipykernel_12908\3908239418.py:5: FutureWarning:

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.
```

```
px.barplot(data=data, x='film_code', y=data['total_sales'], palette='grey')
px.show, sharex=True, sharey=True, ncols=2, figsize=(7, 4)
px.scatterplot(data=data, x='film_code', y='total_sales')
px.show()
px.figure(figsize=(8, 6))
px.heatmap(data.corr(), annot=True)
px.show()
```

```
# let's see the correlation between the columns
px.figure(figsize=(8, 6))
sns.heatmap(data.corr(), cmap=px.
cmap('Greys'), annot=True)
px.show()
```




```
#Assigning X values based on correlation with y
X = data[['film_code', 'cinema_code', 'tickets_sold', 'ticket_out', 'capacity']]
y = data['total_sales']
```

```
#Splitting the data into training and testing
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=0)
```

```
lr = LinearRegression()
lr.fit(X_train,y_train)
```

```
LinearRegression()
```

```
#PREDICTING THE DATA
y_pred = lr.predict(X_test)
```

```
px.figure(figsize=(10,6))
px.scatter(y_test,y_pred,color='r')
px.show()
```

```
sns.barplot(x=df['film_code'],y=df['total_sales'])
px.show()
```