In [1]:
```
!pip install pandas openpyxl
```

```
Requirement already satisfied: pandas in c:\users\hp\onedrive\documents\new\lib\s
ite-packages (2.1.4)
Requirement already satisfied: openpyxl in c:\users\hp\onedrive\documents\new\lib
\site-packages (3.0.10)
Requirement already satisfied: numpy<2,>=1.23.2 in c:\users\hp\onedrive\documents
\new\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\onedrive\doc
uments\new\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\onedrive\documents\new
\lib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\onedrive\documents\n
ew\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: et_xmlfile in c:\users\hp\onedrive\documents\new\l
ib\site-packages (from openpyxl) (1.1.0)
Requirement already satisfied: six>=1.5 in c:\users\hp\onedrive\documents\new\lib
\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [2]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_excel(r'C:\Users\HP\OneDrive\Desktop\pizza_sales_python.xlsx')
```

In [4]:
```
df.head()
```

Out[4]:

	order_details_id	order_id	pizza_id	quantity	order_date	month_name	week_na
0	1	1	hawaiian_m	1	2015-01-01	January	Thursc
1	2	2	classic_dlx_m	1	2015-01-01	January	Thursc
2	3	2	five_cheese_l	1	2015-01-01	January	Thursc
3	4	2	ital_supr_l	1	2015-01-01	January	Thursc
4	5	2	mexicana_m	1	2015-01-01	January	Thursc

In [5]:
```
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [6]:
```
df['order_date'] = pd.to_datetime(df['order_date'])
```

Sales Prediction

In [8]:
```
monthly = df.groupby(df['order_date'].dt.to_period('M'))['total_price'].sum().re
monthly['order_date'] = monthly['order_date'].dt.to_timestamp()
```

In [ ]:
```
```

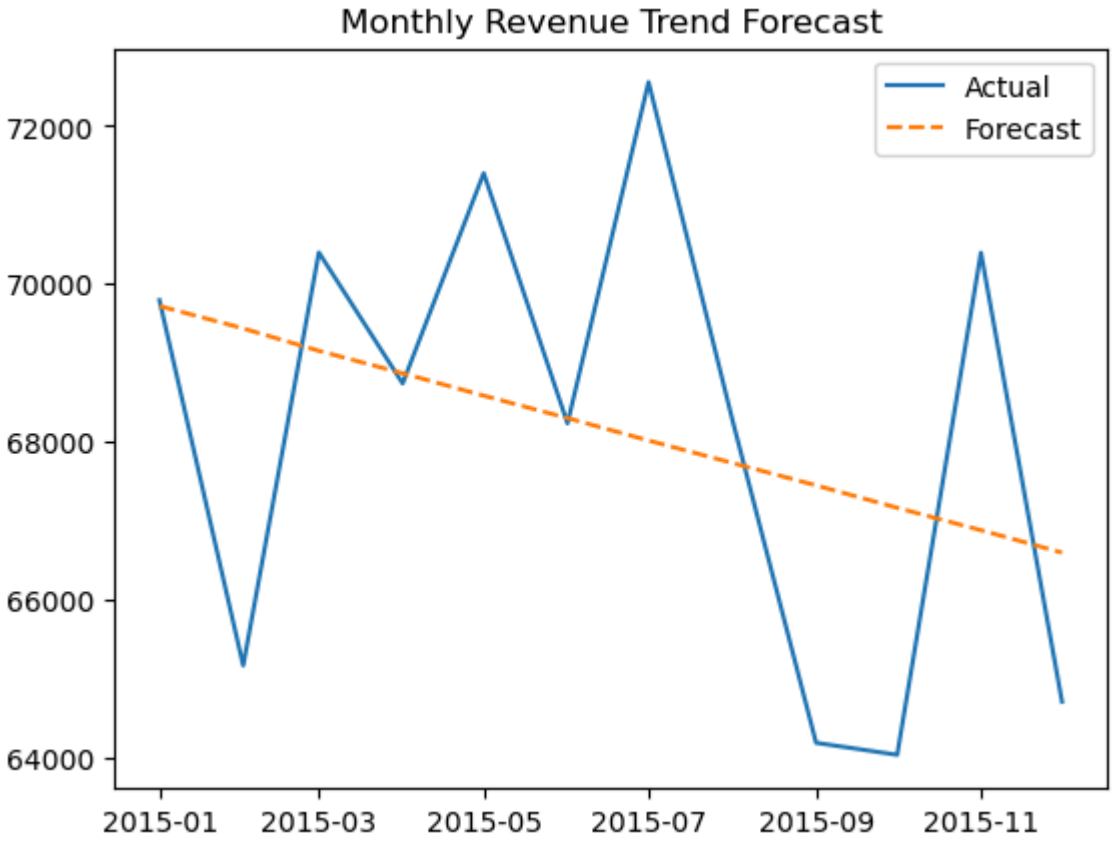
In [9]:
```
monthly['month_num'] = range(len(monthly))
X = monthly[['month_num']]
y = monthly['total_price']
```

In [10]:
```
model = LinearRegression()
model.fit(X, y)
```

Out[10]:
```
▾ LinearRegression
LinearRegression()
```

In [11]:
```
monthly['prediction'] = model.predict(X)
```

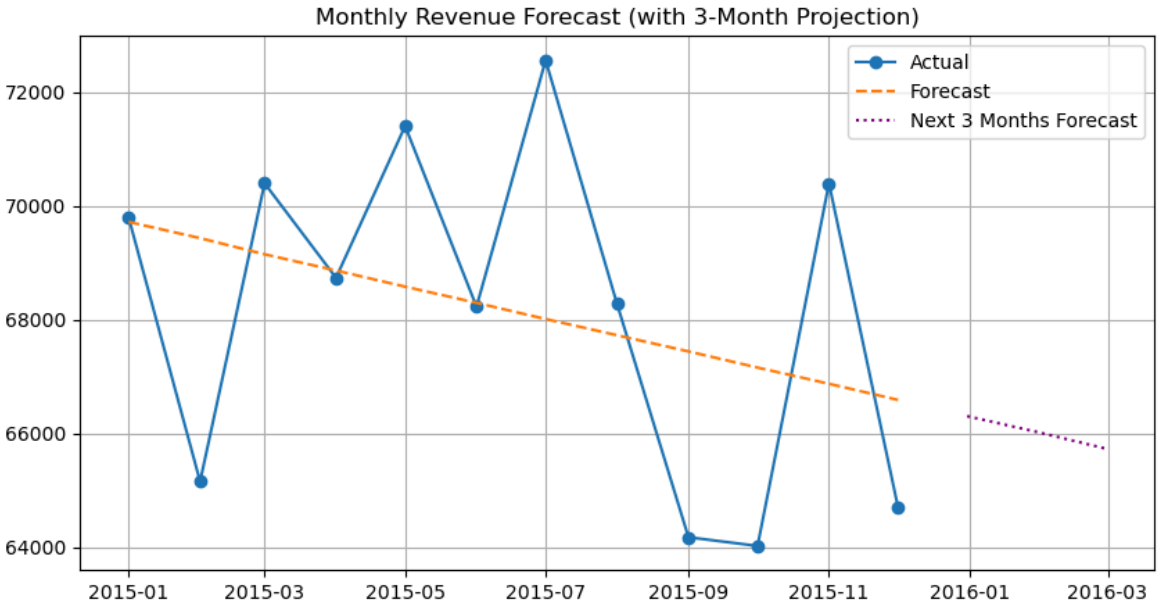
In [12]:
```
plt.plot(monthly['order_date'], y, label='Actual')
plt.plot(monthly['order_date'], monthly['prediction'], label='Forecast', linesty
plt.legend()
plt.title("Monthly Revenue Trend Forecast")
plt.show()
```



Add 3 future months (Jan–Mar 2016) to see where the trend goes

In [13]:
```
future_months = pd.DataFrame({'month_num': range(len(monthly), len(monthly)+3)})
future_preds = model.predict(future_months)
```

In [14]:
```
plt.figure(figsize=(10,5))
plt.plot(monthly['order_date'], y, label='Actual', marker='o')
plt.plot(monthly['order_date'], monthly['prediction'], label='Forecast', linesty
plt.plot(pd.date_range(start=monthly['order_date'].max(), periods=4, freq='M')[:
plt.title("Monthly Revenue Forecast (with 3-Month Projection)")
plt.legend()
plt.grid(True)
plt.show()
```



In [15]:
```
plt.savefig("monthly_revenue_forecast.png")
```

```
<Figure size 640x480 with 0 Axes>
```

In [17]:
```
df['day_part'] = pd.cut(df['hour_of_day'],
                        bins=[0, 11, 15, 18, 24],
                        labels=['Morning', 'Lunch', 'Evening', 'Late'])
```

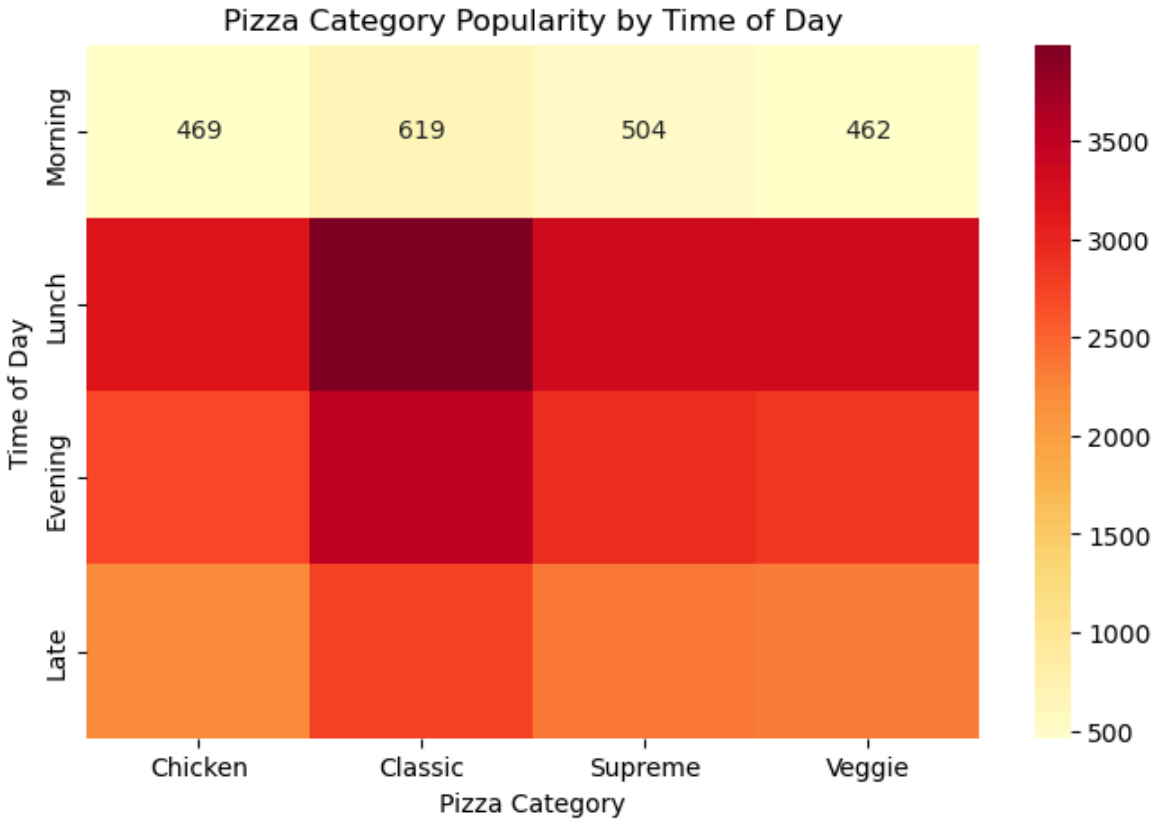
In [18]:
```
heat = df.groupby(['day_part', 'pizza_category'])['order_id'].nunique().unstack(
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_51448\1808406253.py:1: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future
version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  heat = df.groupby(['day_part', 'pizza_category'])['order_id'].nunique().unstack
()
```

In [19]:
```
heat = df.groupby(['day_part', 'pizza_category'], observed=True)['order_id'].nun
```

In [22]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
sns.heatmap(heat, annot=True, fmt='d', cmap='YlOrRd')
plt.title("Pizza Category Popularity by Time of Day")
plt.ylabel("Time of Day")
plt.xlabel("Pizza Category")
plt.show()
```



In [1]:
```
plt.savefig("pizza_category_popularity_by_time_of_day.png")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 1
----> 1 plt.savefig("pizza_category_popularity_by_time_of_day.png")

NameError: name 'plt' is not defined
```

In [ ]: