BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

HCMUTE

# BÁO CÁO ĐỒ ÁN CUỐI KỲ

# ĐỀ TÀI: PHÂN TÍCH NHỮNG YẾU TỐ ẢNH HƯỞNG ĐẾN DOANH THU SIÊU THỊ WALMART

**Môn học: Phân tích Dữ liệu**
**Mã lớp học phần: DAAN436277_23_2_01**
**GVHD: ThS. Nguyễn Văn Thành**


**Nhóm sinh viên thực hiện: Nhóm 7**

| | |
|---|---|
| Nguyễn Đức Kha | 21133044 |
| Nguyễn Nhật Tân | 21133079 |
| Phan Công Danh | 21133014 |
| Lương Tường Vy | 21133093 |


**TP. Hồ Chí Minh, 16 tháng 05 năm 2024**

**DANH SÁCH THÀNH VIÊN THAM GIA**

**THỰC HIỆN ĐỀ TÀI VÀ VIẾT BÁO CÁO**

**Môn: Kho dữ liệu -** *HỌC KÌ II – NĂM HỌC 2023 – 2024*

| STT | HỌ VÀ TÊN | MSSV | TỶ LỆ ĐÓNG GÓP |
|---|---|---|---|
| 1 | Nguyễn Đức Kha | 21133044 | 100% |
| 2 | Nguyễn Nhật Tân | 21133079 | 100% |
| 3 | Phan Công Danh | 21133014 | 100% |
| 4 | Lương Tường Vy | 21133093 | 100% |

**Nhận xét của giảng viên:**

……

……

……

……

……

……

……

……

……

……

……

……

*Ngày … tháng 05 năm 2024*

*Giảng viên chấm điểm*

*Ths. Nguyễn Văn Thành*

# LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến **Trường Đại học Sư phạm Kỹ thuật TPHCM** đã đưa môn học **Phân tích Dữ Liệu** vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – **Th.S Thầy Nguyễn Văn Thành** đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Phân tích Dữ Liệu của Thầy, nhóm em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này. Bộ môn Phân tích Dữ Liệu là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài đồ án này khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong Thầy xem xét và góp ý để bài đồ án của chúng em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

# MỤC LỤC



Link video báo cáo :

# PHÂN CÔNG NHIỆM VỤ NHÓM 7

| **Nhiệm vụ** | Đức Kha | Nhật Tân | Công Danh | Tường Vy |
|---|---|---|---|---|
| Tìm kiếm tập dữ liệu | x | | | |
| Giới thiệu về tập dữ liệu | | | | x |
| Chuẩn bị các câu hỏi nghiên cứu, mục đích nghiên cứu | | x | | |
| Tiền xử lí dữ liệu | | | | x |
| Phân tích đơn biến | x | | | |
| Phân tích đa biến | x | | x | |
| Chứng minh các yếu tố ảnh hưởng đến doanh số bán hàng | | | x | |
| Mô hình hóa dữ liệu | | x | | |
| Kết luận | | | | x |
| Viết báo cáo | | | x | x |
| Video báo cáo | | x | | |

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Lý do chọn đề tài

Walmart đóng vai trò quan trọng trong ngành bán lẻ và có ảnh hưởng lớn đến thị trường. Việc phân tích yếu tố ảnh hưởng đến doanh thu của Walmart sẽ cung cấp thông tin quý giá về hoạt động kinh doanh và xu hướng trong ngành. Walmart tích lũy một lượng lớn dữ liệu liên quan đến doanh thu, gồm thông tin về doanh số bán hàng, giá cả, khách hàng, vị trí cửa hàng và chiến lược tiếp thị. Sử dụng phân tích dữ liệu, ta có thể khai thác thông tin này để hiểu rõ hơn về yếu tố ảnh hưởng đến doanh thu. Với việc nghiên cứu thành công của Walmart trong việc tăng trưởng doanh thu và duy trì vị trí thị trường sẽ mang lại những thông tin hữu ích và chiến lược áp dụng cho các lĩnh vực kinh doanh khác. Tiếp theo, phân tích dữ liệu về yếu tố ảnh hưởng đến doanh thu Walmart có tiềm năng ứng dụng rộng rãi cho các nhà quản lý, chuyên gia tiếp thị và người quan tâm đến ngành bán lẻ. Cuối cùng, đề tài này cũng góp phần phát triển kỹ năng phân tích dữ liệu và tăng cường khả năng cạnh tranh trên thị trường lao động trong thời đại số hóa.

## 1.2. Thông tin về tập dữ liệu

### 1.2.1. Nguồn dữ liệu

- Nhóm sử dụng Tập dữ liệu Walmart Sales được lấy từ trang web Kaggle (kaggle.com).
- Đường dẫn tải tập dữ liệu: Walmart Sales Database

### 1.2.2. Mô tả chi tiết tập dữ liệu

- Bảng chứa dữ liệu của 45 cửa hàng Walmart. Doanh số bán hàng hàng tuần, nhiệt độ không khí và giá nhiên liệu trong khu vực có cửa hàng cụ thể. Cũng như thông tin về chỉ số giá tiêu dùng và tỷ lệ thất nghiệp.
- Dữ liệu được lấy trong thời gian từ 2010 đến 2012. Bao gồm:
  - Số dòng: 6436
  - Số cột: 8

| Tên cột | Ý nghĩa |
|---|---|
| Store | Mã số của cửa hàng |
| Date | Ngày bắt đầu tuần bán hàng |
| Weekly_Sales | Doanh số tuần của cửa hàng |
| Holiday_Flag | Đánh dấu ngày lễ |
| Temperature | Nhiệt độ không khí trong khu vực |
| Fuel_Price | Chi phí nhiên liệu trong khu vực |
| CPI | Chỉ số giá tiêu dùng |
| Unemployment | Tỷ lệ thất nghiệp |

## 1.3. Giới thiệu các công cụ được sử dụng

- Exel: Xem tập dữ liệu
- Visual Studio Code: Phân tích dữ liệu

# CHƯƠNG 2: KIỂM TRA VÀ ĐÁNH GIÁ SƠ BỘ VỀ DỮ LIỆU (EDA).

## 2.1. Kiểm tra dữ liệu

### 2.1.1. Kiểm tra kiểu giá trị của dữ liệu

```
# Kiểm tra kiểu dữ liệu
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6435 entries, 0 to 6434
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Store         6435 non-null   int64  
 1   Date          6435 non-null   object 
 2   Weekly_Sales  6435 non-null   float64
 3   Holiday_Flag  6435 non-null   int64  
 4   Temperature   6435 non-null   float64
 5   Fuel_Price    6435 non-null   float64
 6   CPI           6435 non-null   float64
 7   Unemployment  6435 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 402.3+ KB
```

### 2.1.2. Chuyển đổi kiểu dữ liệu cho cột Date

Entrée [280]:
```
# Chuyển đổi kiểu dữ liệu cột Date
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y')

#df['Week'] = df['Date'].dt.isocalendar().week

df.tail()
```

Out[280]:

| | Store | Date | Weekly_Sales | Holiday_Flag | Temperature | Fuel_Price | CPI | Unemployment |
|---|---|---|---|---|---|---|---|---|
| **6430** | 45 | 2012-09-28 | 713173.95 | 0 | 64.88 | 3.997 | 192.013558 | 8.684 |
| **6431** | 45 | 2012-10-05 | 733455.07 | 0 | 64.89 | 3.985 | 192.170412 | 8.667 |
| **6432** | 45 | 2012-10-12 | 734464.36 | 0 | 54.47 | 4.000 | 192.327265 | 8.667 |
| **6433** | 45 | 2012-10-19 | 718125.53 | 0 | 56.47 | 3.969 | 192.330854 | 8.667 |
| **6434** | 45 | 2012-10-26 | 760281.43 | 0 | 58.85 | 3.882 | 192.308899 | 8.667 |

### 2.1.3. Kiểm tra các dữ liệu Null hoặc bị trùng lặp

Entrée [281]:
```
# Kiểm tra dữ liệu bị thiếu
df.isnull().sum()
```

Out[281]:
```
Store           0
Date            0
Weekly_Sales    0
Holiday_Flag    0
Temperature     0
Fuel_Price      0
CPI             0
Unemployment    0
dtype: int64
```

Không có giá trị bị thiếu nên ta không cần xử lí

## 2.1.4. Kiểm tra describe thống kê của tập dữ liệu

```
# Loại bỏ outliers
# z_scores = np.abs(stats.zscore(df.select_dtypes(include=[np.number])))
# df = df[(z_scores < 3).all(axis=1)]
```

0.0s Python

```
# In ra các giá trị thống kê của tập dữ liệu
df.describe().style.background_gradient(cmap='bone_r')
```

0.1s Python

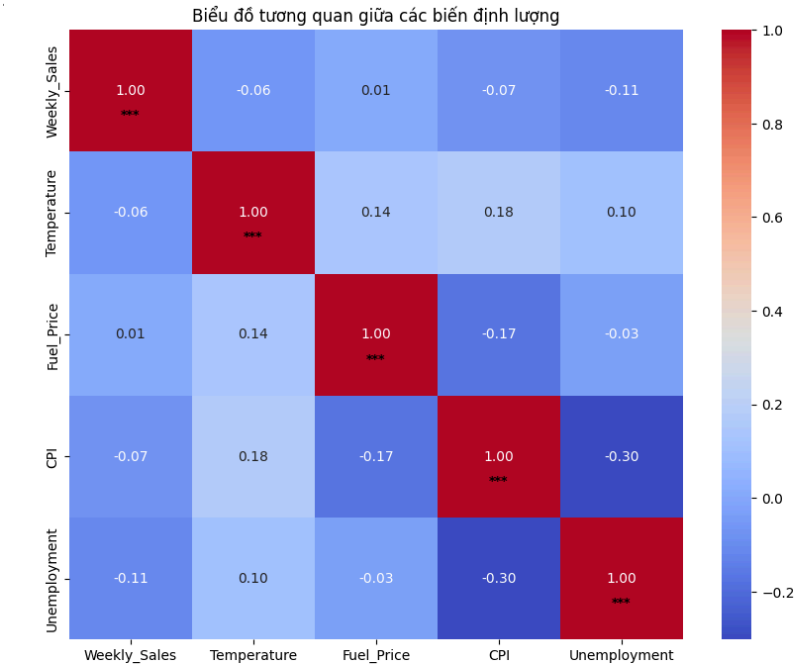
| | Store | Date | Weekly_Sales | Holiday_Flag | Temperature | Fuel_Price | CPI | Unemployment |
|---|---|---|---|---|---|---|---|---|
| count | 6435.000000 | 6435 | 6435.000000 | 6435.000000 | 6435.000000 | 6435.000000 | 6435.000000 | 6435.000000 |
| mean | 23.000000 | 2011-06-17 00:00:00 | 1046964.877562 | 0.069930 | 60.663782 | 3.358607 | 171.578394 | 7.999151 |
| min | 1.000000 | 2010-02-05 00:00:00 | 209986.250000 | 0.000000 | -2.060000 | 2.472000 | 126.064000 | 3.879000 |
| 25% | 12.000000 | 2010-10-08 00:00:00 | 553350.105000 | 0.000000 | 47.460000 | 2.933000 | 131.735000 | 6.891000 |
| 50% | 23.000000 | 2011-06-17 00:00:00 | 960746.040000 | 0.000000 | 62.670000 | 3.445000 | 182.616521 | 7.874000 |
| 75% | 34.000000 | 2012-02-24 00:00:00 | 1420158.660000 | 0.000000 | 74.940000 | 3.735000 | 212.743293 | 8.622000 |
| max | 45.000000 | 2012-10-26 00:00:00 | 3818686.450000 | 1.000000 | 100.140000 | 4.468000 | 227.232807 | 14.313000 |
| std | 12.988182 | nan | 564366.622054 | 0.255049 | 18.444933 | 0.459020 | 39.356712 | 1.875885 |

Weekly_Sales có độ lệch chuẩn khá cao bằng 1/2 trung vị, gợi ý rằng doanh số bán hàng hàng tuần không ổn định và có thể chịu ảnh hưởng mạnh mẽ bởi các sự kiện nhất định (ví dụ như Black Friday, Giáng sinh, hoặc các ngày lễ lớn khác).

## 2.2. Phân tích sơ bộ

### 2.2.1. Phân tích đơn biến

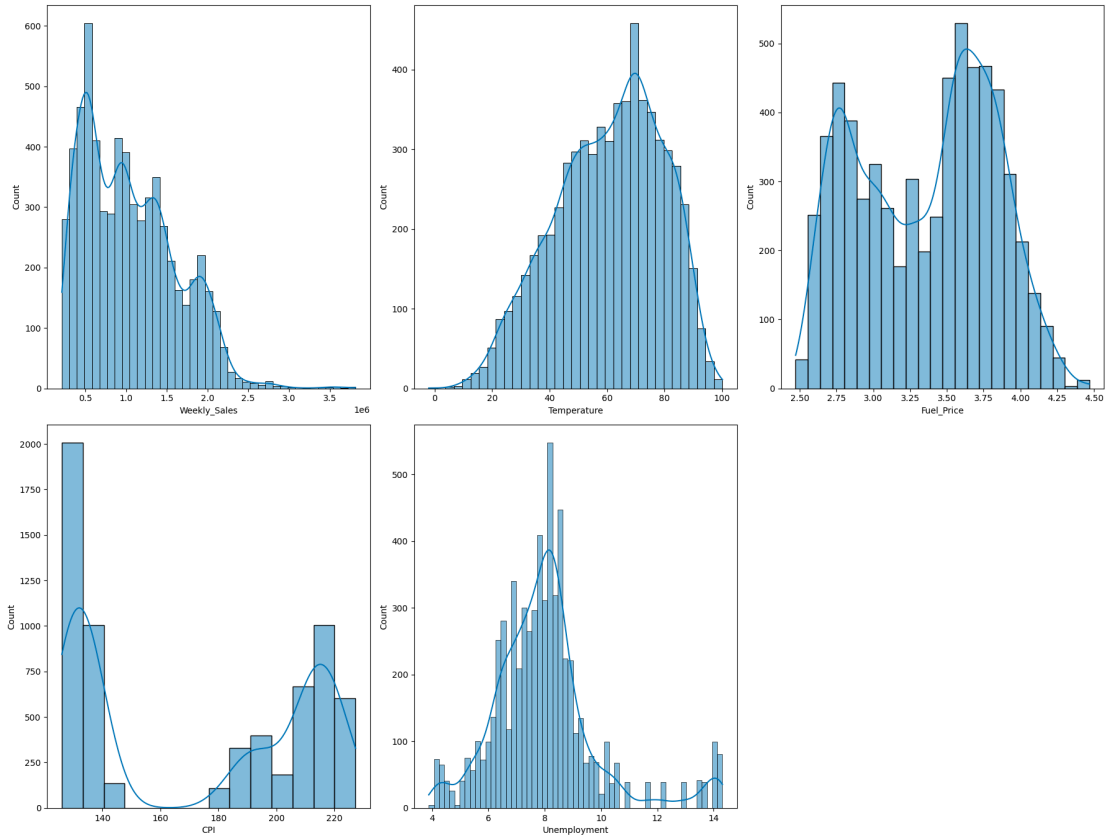
```
df1 = df.drop([df.columns[0],
df.columns[1],df.columns[8],df.columns[9],df.columns[10],df.columns[3]], axis=1)
corr_matrix = df1.corr()
# Vẽ biểu đồ nhiệt tương quan
plt.figure(figsize=(10, 8))
ax = sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm',
annot_kws={"size": 10})
# Thêm chú thích cho các mức độ tương quan đáng kể
significant_level = 0.5  # Giả định mức độ tương quan đáng kể là 0.5
for i, row in enumerate(corr_matrix.values):
    for j, value in enumerate(row):
        if value > significant_level or value < -significant_level:
            text_coords = ax.get_xticks()[j], ax.get_yticks()[i] + 0.2
            plt.text(text_coords[0], text_coords[1], '***', ha='center',
va='center', color='black', fontsize=9, fontweight='bold')
# Hiển thị biểu đồ
plt.title('Biểu đồ tương quan giữa các biến định lượng')
plt.show()
```




Không có sự tương quan tuyến tính nào đáng kể ở bảng trên. Nhưng có vẻ như tỉ lệ thất nghiệp cao ít nhiều ảnh hưởng tới CPI tại cửa hàng.

```
features = ['Weekly_Sales', 'Temperature', 'Fuel_Price', 'CPI', 'Unemployment']
# Set the figure size
plt.figure(figsize=(18, 20))
# Loop through each column in your dataset
for i, col in enumerate(features):
    # Create subplots
    plt.subplot(3, 3, i+1)

    # Plot histogram for the current column
    sns.histplot(data=df, x=col, kde=True)
plt.tight_layout()
plt.show()
```




**Doanh số bán hàng hàng tuần:** Biểu đồ doanh số bán hàng hàng tuần biểu thị sự phân phối lệch phải, cho thấy rằng có tương đối ít trường hợp doanh số bán hàng rất cao so với số lượng bán hàng thấp hơn. Điều này có thể cho thấy doanh số bán hàng thỉnh thoảng tăng đột biến hoặc một vài giai đoạn hoạt động hiệu quả.

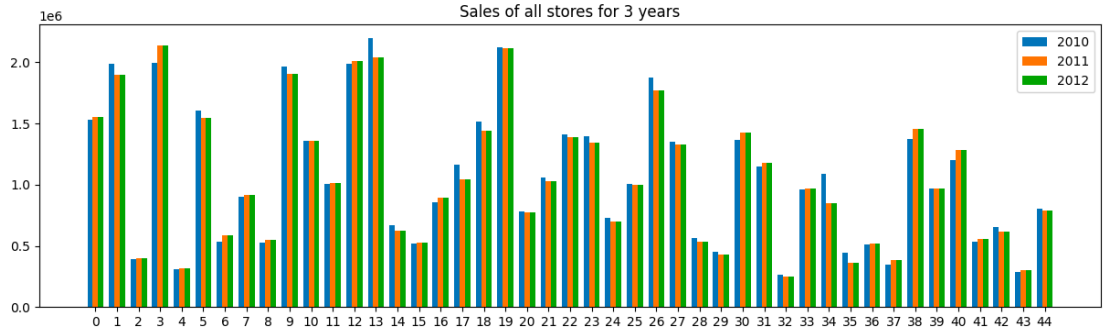
**Nhiệt độ và thất nghiệp:** Biểu đồ về nhiệt độ và tỷ lệ thất nghiệp thể hiện sự phân bố gần như bình thường, cho thấy phần lớn các điểm dữ liệu tập trung quanh giá trị trung bình với tương đối ít ngoại lệ. Điều này cho thấy những yếu tố này có thể tuân theo các mô hình điển hình mà không có sai lệch đáng kể.

**Giá nhiên liệu, CPI:** Biểu đồ giá nhiên liệu và CPI thể hiện sự phân bố hai chiều, cho thấy sự hiện diện của hai đỉnh hoặc hai chế độ riêng biệt trong dữ liệu. Điều này có thể ngụ ý sự tồn tại của các điều kiện hoặc trạng thái thị trường khác nhau trong tập dữ liệu, có khả năng chỉ ra các tình huống kinh tế hoặc hành vi của người tiêu dùng khác nhau.

### 2.2.2. Phân tích hai biến

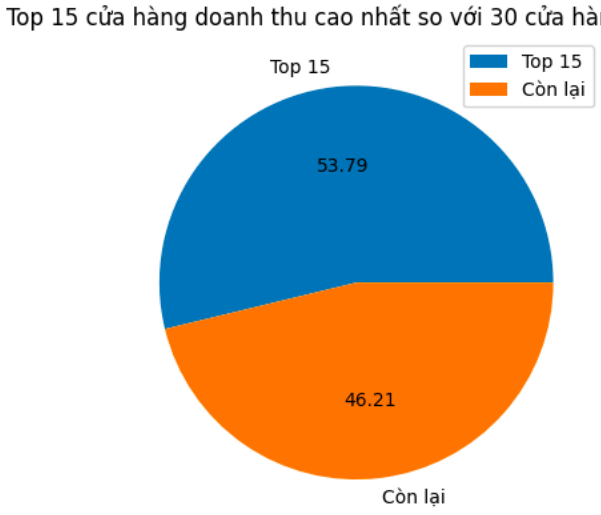
```
def year_wise_sales(year):
    filt = df['Year'] == year
    result = df.loc[filt].groupby(by='Store')
    values = result['Weekly_Sales'].mean()
    return values
year_2010 = year_wise_sales(2010)
year_2011 = year_wise_sales(2011)
year_2012 = year_wise_sales(2011)
width = 0.25
x_index =np.arange(len(year_2010))
```

```
plt.figure(figsize=(15,4))
plt.title('Sales of all stores for 3 years')
plt.bar(x_index-width,year_2010,width=width,label='2010')
plt.bar(x_index,year_2011,width=width,label='2011')
plt.bar(x_index+width,year_2012,width=width,label='2012')
plt.xticks(x_index)
plt.legend()
plt.show()
```



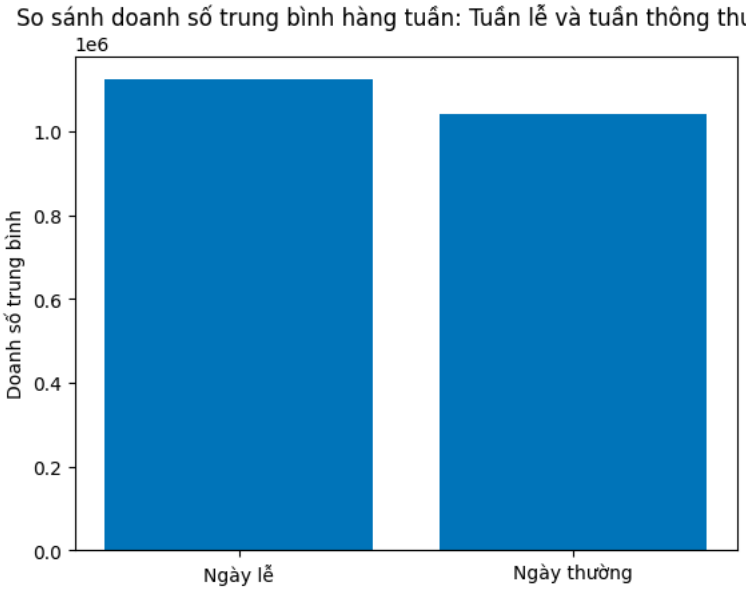

Có sự chênh lệch lớn giữa doanh thu của các cửa hàng dù ở bất cứ thời điểm nào.

```
stores = df.groupby(by='Store')
top_stores= sum(stores['Weekly_Sales'].sum().sort_values(ascending=False)[:15])
other_stores = sum(stores['Weekly_Sales'].sum().sort_values(ascending=False)[15:])
total_sales = sum(stores['Weekly_Sales'].sum())
top = (top_stores / total_sales) *100
bottom = (other_stores / total_sales) *100
sales = [top,bottom]
labels = ['Top 15','Còn lại']
plt.title('Top 15 cửa hàng doanh thu cao nhất so với 30 cửa hàng còn lại')
plt.pie(sales,labels=labels,autopct='%0.2f')
plt.legend()
plt.show()
```



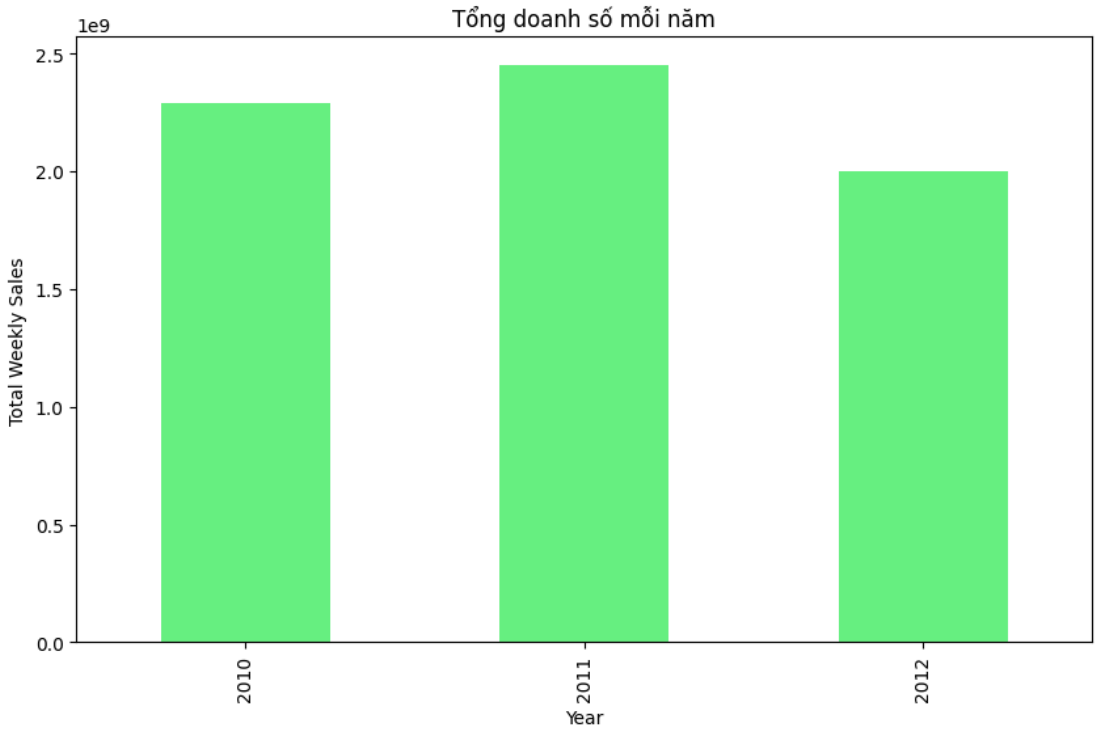

```
average_sales_holiday = df[df['Holiday_Flag'] == 1]['Weekly_Sales'].mean()
```

```
average_sales_regular = df[df['Holiday_Flag'] == 0]['Weekly_Sales'].mean()
plt.bar(['Ngày lễ', 'Ngày thường'], [average_sales_holiday, average_sales_regular])
plt.ylabel('Doanh số trung bình')
plt.title('So sánh doanh số trung bình hàng tuần: Tuần lễ và tuần thông thường')
plt.show()
```

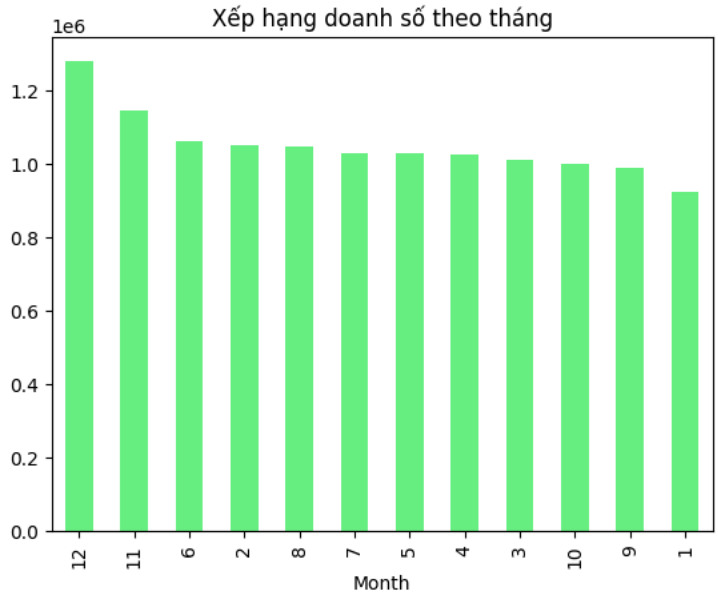



Ta thấy doanh số ngày lễ lớn hơn ngày bình thường nhưng không đáng kể.

```
ear_sales = df.groupby(df['Year'])['Weekly_Sales'].sum()
# Vẽ biểu đồ thanh cho doanh số hàng tuần theo năm
plt.figure(figsize=(10, 6))
year_sales.plot(kind='bar', color='lightgreen')
plt.title('Tổng doanh số mỗi năm')
plt.xlabel('Year')
plt.ylabel('Total Weekly Sales')
plt.show()
```

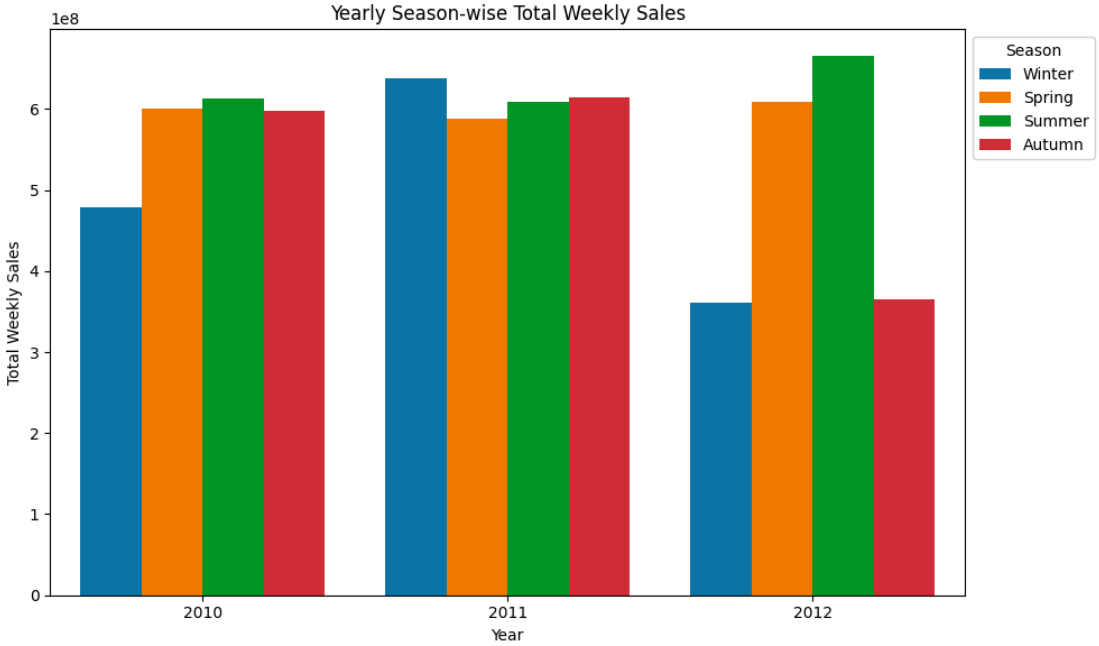



```
months = df.groupby(by='Month')
months['Weekly_Sales'].mean().sort_values(ascending=False).plot(kind='bar',color='l
ightgreen')
plt.title('Xếp hạng doanh số theo tháng')
```




Vào các dịp cuối năm người dân cho nhu cầu mua sắm cao hơn để chuẩn bị cho lễ Tạ ơn, Giáng sinh.

```
seasonwise_weekly_sales = {}
for season in df['Season'].unique():
    season_sales = df[df['Season'] == season].groupby('Year')['Weekly_Sales'].sum()
    seasonwise_weekly_sales[season] = season_sales
plot_data = []
for season, sales in seasonwise_weekly_sales.items():
    for year, weekly_sales in sales.items():
        plot_data.append({'Year': year, 'Season': season, 'Weekly Sales':
weekly_sales})
plot_data = pd.DataFrame(plot_data)
fig, ax = plt.subplots(figsize=(10, 6))
sns.barplot(data=plot_data, x='Year', y='Weekly Sales', hue='Season', ax=ax, ci=None)
ax.set_xlabel('Year')
ax.set_ylabel('Total Weekly Sales')
ax.set_title('Yearly Season-wise Total Weekly Sales')
ax.legend(title='Season', loc='upper left', bbox_to_anchor=(1, 1))
plt.tight_layout()
plt.show()
```



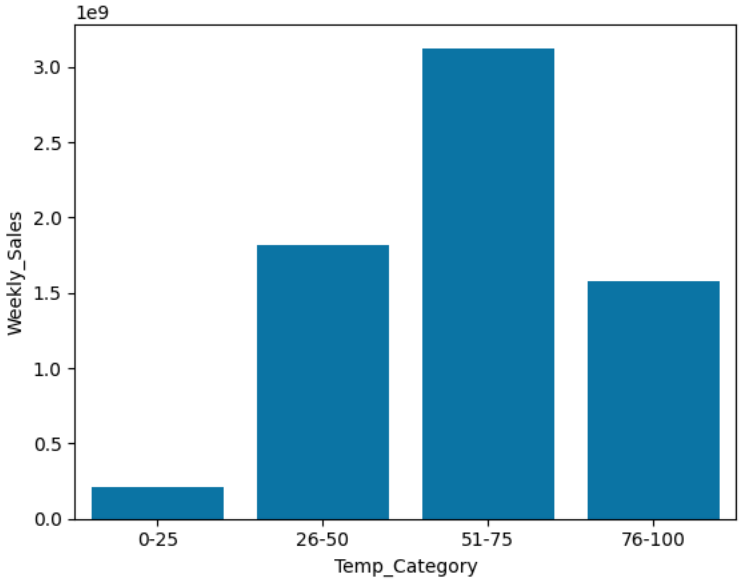

Các mùa có doanh thu cao nhất:
- 2010: Mùa xuân
- 2011: Mùa thu
- 2012: Mùa hè

### 2.2.3. Chứng minh các yếu tố ảnh hưởng đến doanh số bán hàng

#### a. Temperature v/s Sales

```
df["Temperature"].min()
```

```
df["Temperature"].max()
bins = [0, 25, 50, 75, 100]  # Tạo bin để sắp xếp các giá trị nhiệt độ
labels = ["0-25", "26-50", "51-75", "76-100"]  # Labels
#Phân loại nhiệt độ vào các khoảng chia trước
df["Temp_Category"] = pd.cut(df["Temperature"], bins=bins, labels=labels)
df["Temp_Category"].value_counts()
df_temp_sales = df.groupby("Temp_Category")["Weekly_Sales"].sum().reset_index()
sns.barplot(x="Temp_Category", y="Weekly_Sales", data=df_temp_sales)
```




Ta được ra được kết luận:

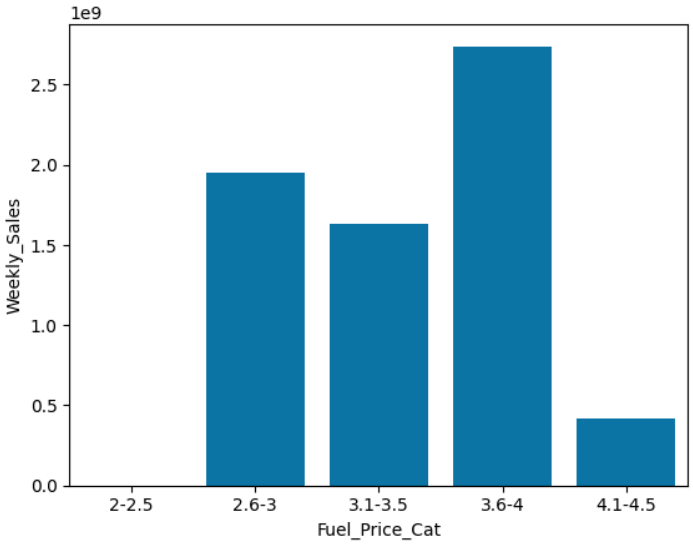
- Cửa hàng hoạt động tốt trong nhiệt độ vừa phải là 26 - 50

- Bán hàng đã đạt đỉnh ở nhiệt độ trung bình đến cao là 51 - 75

- Doanh số giảm ở nhiệt độ rất cao và thấp, đó là 0 - 25 & 76 -100

**b. Fuel_Price v/s Sales**

```
df["Fuel_Price"].min()
df["Fuel_Price"].max()
bins1 = [2, 2.5, 3, 3.5, 4, 4.5, ]  # Tạo bin và label để phân loại giá nhiên liệu
labels1 = ["2-2.5", "2.6-3", "3.1-3.5", "3.6-4", "4.1-4.5"]
df["Fuel_Price_Cat"] = pd.cut(df["Fuel_Price"], bins=bins1, labels=labels1)
df["Fuel_Price_Cat"].value_counts()
df_Fuel_Price_Sales                                                    =
df.groupby("Fuel_Price_Cat")["Weekly_Sales"].sum().reset_index()
df_Fuel_Price_Sales
sns.barplot(x="Fuel_Price_Cat", y="Weekly_Sales", data=df_Fuel_Price_Sales)
```




Ta kết luận được:

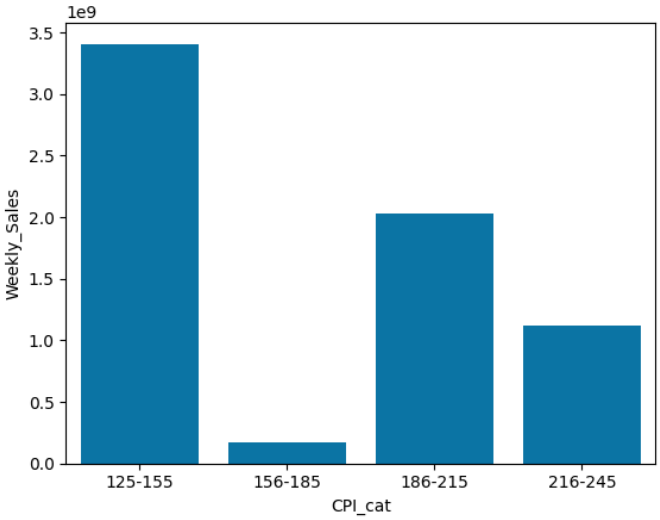
- Khi giá nhiên liệu được kiểm duyệt, hiệu suất của các cửa hàng cũng rất cao.

- Nhưng giá nhiên liệu đã vượt quá mức độ trung bình là 3,5, và doanh số cũng tăng rất cao.

- Và khi giá nhiên liệu tăng cao, sức mua của người dân đã giảm.

**c. CPI v/s Sales**

```
df['CPI'].min()
df['CPI'].max()
bins2=[125,155,185,215,245]
labels2=['125-155','156-185','186-215','216-245',]
df['CPI_cat']= pd.cut(df['CPI'],bins=bins2,labels=labels2)
df['CPI_cat'].value_counts()
df_Cpi_Sales = df.groupby('CPI_cat')['Weekly_Sales'].sum().reset_index()
df_Cpi_Sales
sns.barplot(x='CPI_cat',y='Weekly_Sales',data=df_Cpi_Sales)
```




Ta kết luận được:

- Khi chỉ số tiêu dùng thấp hơn thì hiệu quả hoạt động của các cửa hàng cao.

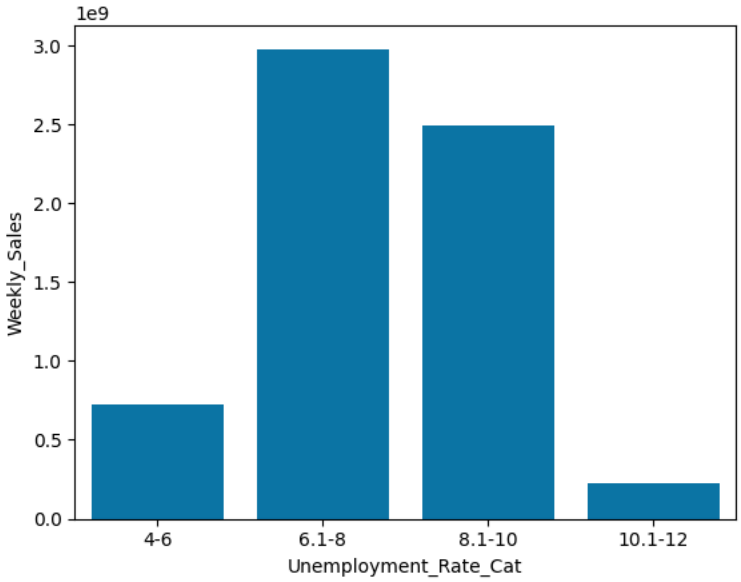
- Trong thời đầu tiên, doanh số bán hàng giảm đáng kể.

- Ở cấp độ thứ ba khi chỉ số tiêu dùng tăng cao, người dân có thể lo sợ về giá cả trong tương lai và cũng mua thêm hàng tạp hóa cho nhu cầu trong tương lai.

- Cuối cùng khi lạm phát lên đến đỉnh điểm thì doanh số bán hàng lại giảm xuống.

**d. Unemployment V/s Sales**

```
df['Unemployment'].min()
df['Unemployment'].max()
bins3=[4,6,8,10,12]
labels3=['4-6','6.1-8','8.1-10','10.1-12']
df['Unemployment_Rate_Cat']= pd.cut(df['Unemployment'],bins=bins3,labels=labels3)
df['Unemployment_Rate_Cat'].value_counts()
df_Un_Emp_Sales                                                                =
df.groupby('Unemployment_Rate_Cat')['Weekly_Sales'].sum().reset_index()
sns.barplot(x='Unemployment_Rate_Cat',y='Weekly_Sales',data=df_Un_Emp_Sales)
```




Ta kết luận được:

- Khi tỷ lệ thất nghiệp ở mức từ 6 đến 10, các cửa hàng đang hoạt động tốt

- Nhưng bất cứ khi nào tỷ lệ thất nghiệp tăng lên thì doanh số bán hàng lại giảm

# CHƯƠNG 3: LỰA CHỌN MÔ HÌNH PHÂN TÍCH VÀ KẾT QUẢ

## 3.1. Mã hóa dữ liệu và Correlation

### a. Random Forest

```
df=pd.read_csv('…/Walmart_sales.csv')
df

y = df['Weekly_Sales']
x=df.drop(columns = ["Weekly_Sales","Date"])
x.info()
```

### b. KNN

```
y = df['Weekly_Sales']
feature_names = df.columns.drop(["Weekly_Sales", "Date"])
```

## 3.2. Lựa chọn mô hình

### a. Random Forest

```
from sklearn.ensemble import RandomForestRegressor
rf= RandomForestRegressor()
model=rf.fit(x,y)
model.score(x,y)
```

### b. KNN

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import math

x = df.drop(columns=[feature, 'Weekly_Sales','Date'])
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Khởi tạo và huấn luyện mô hình KNN
knn = KNeighborsRegressor(n_neighbors=3)
knn.fit(X_train, y_train)

# Dự đoán trên tập kiểm tra không loại bỏ đặc trưng nào
y_pred = knn.predict(X_test)
rmseDataSet = math.sqrt(mean_squared_error(y_test, y_pred))
```

## 3.3. Chọn mô hình: sử dụng các mô hình như sau

### a. Random Forest

```
from sklearn.metrics import mean_squared_error, r2_score
import math
y_pred = model.predict(x)
# Tính toán chỉ số
mse = mean_squared_error(y, y_pred)
rmse = math.sqrt(mse)
r2 = r2_score(y, y_pred)
print("RMSE:", rmse)
print("R-squared:", r2)
```

**Kết quả:**

RMSE: 52939.98182663073

R-squared: 0.9911993970215646

**Nhận xét:**

- RMSE = 52353 so với giá trị trung bình cột Weekly_Sales 1046964 mức chênh lệch dự đoán chỉ 5%
- Với chỉ R-squared = 0.99 mô hình Random-Forest phù hợp với tập dữ liệu

### b. KNN

```
rmse_results = {}
for feature  in feature_names:
    # Tạo một DataFrame mới bằng cách loại bỏ một đặc trưng
    x = df.drop(columns=[feature , 'Weekly_Sales','Date'])  # Loại bỏ cột mục tiêu và đặc
trưng hiện tại

    # Chia dữ liệu thành tập huấn luyện và tập kiểm tra
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

    # Khởi tạo và huấn luyện mô hình KNN
    knn = KNeighborsRegressor(n_neighbors=5)
    knn.fit(X_train, y_train)

    # Dự đoán trên tập kiểm tra
    y_pred = knn.predict(X_test)

    # Tính RMSE và lưu vào dictionary
    rmse = math.sqrt(mean_squared_error(y_test, y_pred))
    rmse_results[feature] = rmse
```

## 3.4. Kết quả

### a. Random Forest

```
importances = model.feature_importances_
feature_names = x.columns
feature_importances = sorted(zip(feature_names, importances), key=lambda x: x[1],
reverse=True)
for feature, importance in feature_importances:
    print(f"Importance of {feature}: {importance:.2f}")
```

**Kết quả:**
Importance of Store: 0.66
Importance of CPI: 0.17
Importance of Unemployment: 0.11
Importance of Temperature: 0.03
Importance of Fuel_Price: 0.03
Importance of Holiday_Flag: 0.00
**Nhận xét:**
- **Store (0.68):** Đặc trưng này có tầm quan trọng cao nhất, chiếm 68% tổng tầm quan trọng. Điều này cho thấy vị trí hoặc số hiệu của cửa hàng có ảnh hưởng lớn nhất đến kết quả dự đoán, có thể do các yếu tố như kích thước cửa hàng, vị trí địa lý, lượng khách hàng, hoặc các yếu tố quản lý khác nhau giữa các cửa hàng.
- **CPI (0.17):** Chỉ số giá tiêu dùng (CPI) là đặc trưng quan trọng thứ hai, chiếm 17% tầm quan trọng. Điều này cho thấy CPI, một chỉ số kinh tế phản ánh mức giá tiêu dùng chung, cũng có

tác động đáng kể đến doanh số bán hàng. CPI cao có thể ảnh hưởng đến sức mua của người tiêu dùng.
- **Unemployment (0.10):** Tỷ lệ thất nghiệp chiếm 10% tầm quan trọng. Điều này cho thấy tình hình việc làm tại khu vực gần cửa hàng cũng ảnh hưởng tới doanh thu, với suy đoán rằng tỷ lệ thất nghiệp cao có thể làm giảm khả năng chi tiêu của người tiêu dùng.
- **Temperature (0.03)** và Fuel Price (0.02): Nhiệt độ và giá nhiên liệu có tầm quan trọng thấp hơn nhiều, lần lượt là 3% và 2%. Những đặc trưng này có ảnh hưởng nhỏ đến dự đoán doanh thu, có thể do chúng không trực tiếp ảnh hưởng đến quyết định mua sắm hàng ngày của người tiêu dùng như các yếu tố kinh tế khác.
- **Holiday_Flag (0.00):** Có vẻ như việc có phải là ngày lễ hay không không ảnh hưởng đáng kể đến doanh thu trong mô hình này, điều này có thể bất ngờ vì người ta thường kỳ vọng ngày lễ sẽ có sự tăng doanh số.Store (0.68): Đặc trưng này có tầm quan trọng cao nhất, chiếm 68% tổng tầm quan trọng. Điều này cho thấy vị trí hoặc số hiệu của cửa hàng có ảnh hưởng lớn nhất đến kết quả dự đoán, có thể do các yếu tố như kích thước cửa hàng, vị trí địa lý, lượng khách hàng, hoặc các yếu tố quản lý khác nhau giữa các cửa hàng.
- **CPI (0.17):** Chỉ số giá tiêu dùng (CPI) là đặc trưng quan trọng thứ hai, chiếm 17% tầm quan trọng. Điều này cho thấy CPI, một chỉ số kinh tế phản ánh mức giá tiêu dùng chung, cũng có tác động đáng kể đến doanh số bán hàng. CPI cao có thể ảnh hưởng đến sức mua của người tiêu dùng.
- **Unemployment (0.10):** Tỷ lệ thất nghiệp chiếm 10% tầm quan trọng. Điều này cho thấy tình hình việc làm tại khu vực gần cửa hàng cũng ảnh hưởng tới doanh thu, với suy đoán rằng tỷ lệ thất nghiệp cao có thể làm giảm khả năng chi tiêu của người tiêu dùng.
- **Temperature (0.03)** và Fuel Price (0.02): Nhiệt độ và giá nhiên liệu có tầm quan trọng thấp hơn nhiều, lần lượt là 3% và 2%. Những đặc trưng này có ảnh hưởng nhỏ đến dự đoán doanh thu, có thể do chúng không trực tiếp ảnh hưởng đến quyết định mua sắm hàng ngày của người tiêu dùng như các yếu tố kinh tế khác.
- **Holiday_Flag (0.00):** Có vẻ như việc có phải là ngày lễ hay không không ảnh hưởng đáng kể đến doanh thu trong mô hình này, điều này có thể bất ngờ vì người ta thường kỳ vọng ngày lễ sẽ có sự tăng doanh số.

### b. KNN

```
# In ra kết quả RMSE khi loại bỏ từng đặc trưng
print(f"RMSE full is : {rmseDataSet:.4f}")
for feature, rmse in rmse_results.items():
    print(f"RMSE when '{feature}' is removed: {rmse:.4f}")
```

**Kết quả:**
RMSE full is : 289114.1088
RMSE when 'Store' is removed: 588396.3380
RMSE when 'Holiday_Flag' is removed: 303147.5627
RMSE when 'Temperature' is removed: 156547.5207
RMSE when 'Fuel_Price' is removed: 304984.8224
RMSE when 'CPI' is removed: 321040.4716
RMSE when 'Unemployment' is removed: 327015.5912
**Nhận xét:**
- **'Store':** Khi loại bỏ đặc trưng này, RMSE tăng đáng kể từ 289114.1 lên 590288.67. Điều này cho thấy 'Store' là đặc trưng rất quan trọng và có ảnh hưởng lớn đến dự đoán doanh thu hàng tuần. Sự khác biệt lớn về RMSE chứng tỏ rằng thông tin về cửa hàng cụ thể đóng vai trò chính trong việc xác định doanh số bán hàng.

- **'Holiday_Flag':** Khi loại bỏ đặc trưng này, RMSE thực tế lại giảm nhẹ từ 327015.59 xuống 303147.56. Điều này có thể cho thấy rằng 'Holiday_Flag' có thể không cần thiết hoặc thậm chí làm giảm hiệu quả của mô hình. Có thể trong dữ liệu có đủ thông tin khác để bù đắp cho sự thiếu hụt này hoặc cột này không mang lại thông tin hữu ích cho mô hình.
- **'Temperature', 'Fuel_Price','Unemployment' và 'CPI':** Khi loại bỏ các đặc trưng này, RMSE tăng nhẹ hoặc giảm nhẹ, cho thấy chúng có tác động vừa phải đến dự đoán của mô hình. Đặc biệt, 'Temperature' có vẻ như là đặc trưng quan trọng hơn cả vì khi loại bỏ nó, RMSE giảm khá nhiều (156547.52), điều này có thể gợi ý rằng nhiệt độ có ảnh hưởng đến mô hình nhưng có thể làm giảm độ chính xác khi có mặt.

# CHƯƠNG 4: ĐÁNH GIÁ VÀ KẾT LUẬN

## 4.1. Kết quả đạt được

1) **Hiểu và tiền xử lý dữ liệu**
   a) Dữ liệu được tải từ file CSV và bao gồm các thông tin như doanh số hàng tuần, ngày bán hàng, nhiệt độ, giá nhiên liệu, CPI, tỷ lệ thất nghiệp và cờ đánh dấu các tuần lễ đặc biệt.
   b) Tiến hành kiểm tra và làm sạch dữ liệu nếu cần thiết, bao gồm việc xử lý các giá trị thiếu, ngoại lệ và chuyển đổi dữ liệu về dạng phù hợp cho mô hình học máy.
2) **Xây dựng mô hình dự đoán**
   a) Sử dụng các mô hình hồi quy như K-Nearest Neighbors (KNN) và Random Forest để dự đoán doanh số bán hàng.
   b) Chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu suất mô hình.

## 4.2. Những hạn chế

1) **Dữ liệu không đầy đủ hoặc không đồng nhất**: Dữ liệu bán hàng lịch sử có thể không đầy đủ hoặc không đồng nhất, dẫn đến mô hình dự đoán không phản ánh chính xác thực tế. Ví dụ, thiếu dữ liệu về một số tuần hoặc các biến số kinh tế không được cập nhật kịp thời.
2) **Không có dữ liệu tương lai**: Mô hình chỉ dựa trên dữ liệu lịch sử và không thể dự đoán chính xác các sự kiện không lường trước được hoặc các thay đổi đột ngột trong thị trường.

## 4.3. Tài liệu tham khảo

https://www.kaggle.com/datasets/yasserh/walmart-dataset/data
https://www.youtube.com/watch?v=xi0vhXFPegw