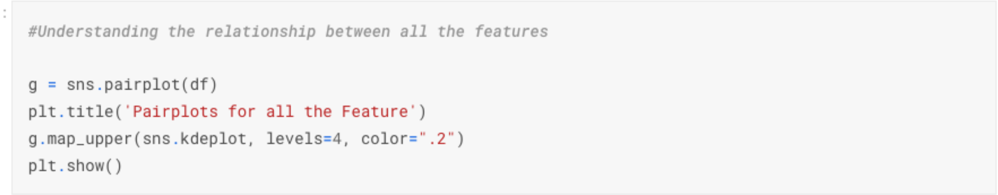
Walmart Sales Forecasting

1. **Walmart Sales Prediction - (Best ML Algorithms) by M Yasser**
   link : https://www.kaggle.com/code/yasserh/walmart-sales-prediction-best-ml-algorithms

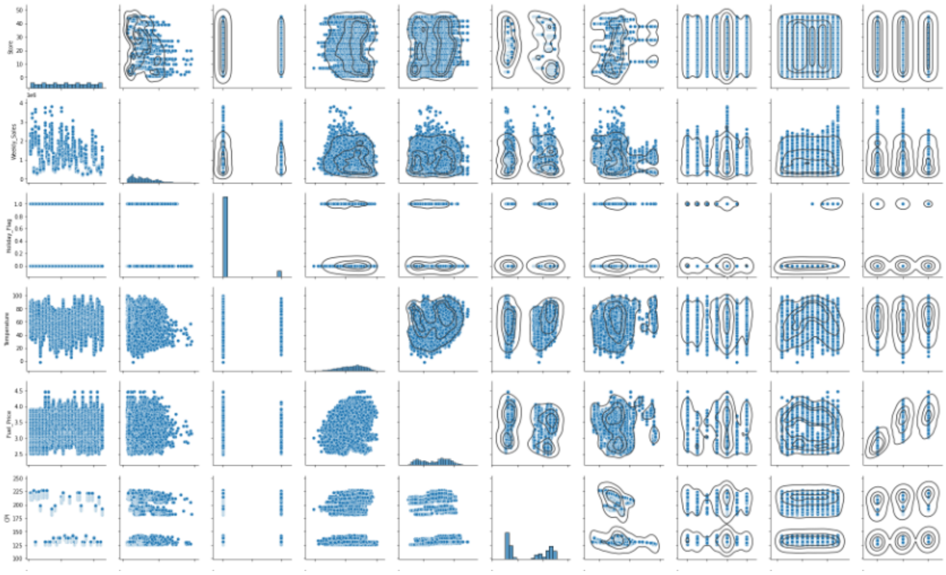
Out of all the work on the Walmart Sales forecasting code I have reviewed, I appreciate Yasser's work on the Walmart dataset the most. He demonstrated his ability to apply machine learning to real-world issues with his systematic and effectively carried out approach, which was guided by his machine learning experience. He has given **great thought to the data** and **used the proper preprocessing techniques**, such as feature engineering, feature scaling, and outlier removal. Additionally, he **divided the data into training and testing sets, trained and evaluated several regression models**, including Multiple Linear Regression (MLR), Ridge Linear Regression (RLR), Lasso Linear Regression (LLR), Elastic-Net Regression (ENR), and Polynomial Regression (PNR) on the training set, tested the model, and then generated predictions using new data. In conclusion, he has recommended using MLR to **predict weekly sales for Walmart stores**.

Some of the highlights of his works are as below:

1. Understanding **the relationships between all the features and identifying that some features have linear relationship**

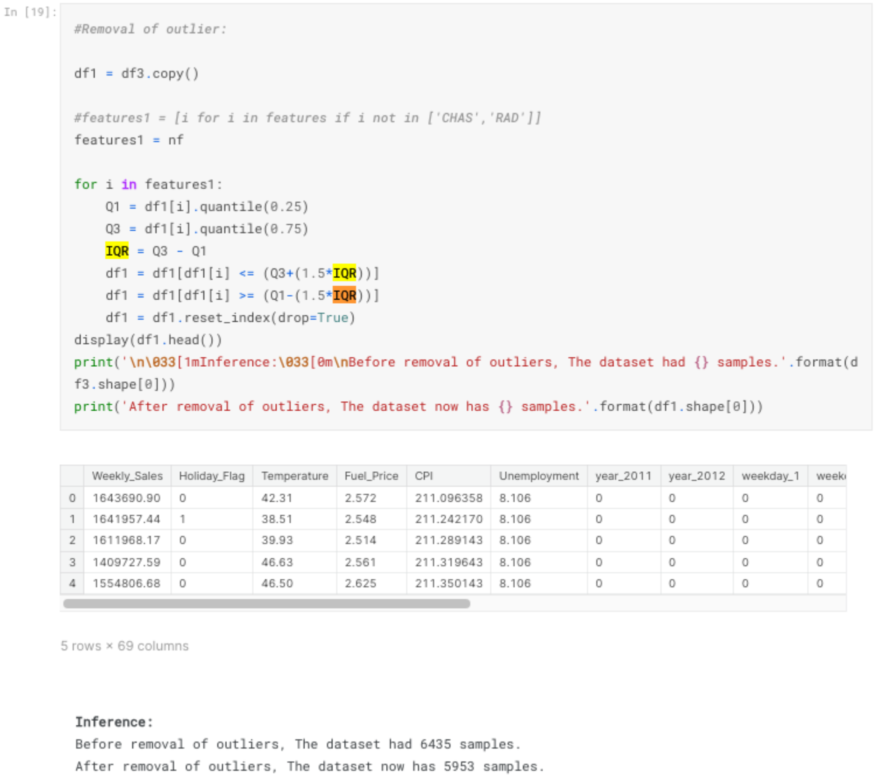


```
#Understanding the relationship between all the features

g = sns.pairplot(df)
plt.title('Pairplots for all the Feature')
g.map_upper(sns.kdeplot, levels=4, color=".2")
plt.show()
```



2. **Removal of outlier**: The interquartile range (IQR) approach was used in the code to remove outliers from the Walmart dataset- The IQR for each feature was first calculated. The middle 50% of the data's spread is measured by the IQR. It is calculated by taking the first quartile (Q1) and third quartile (Q3) and subtracting them. The code lowered the

Walmart dataset from 6435 samples to 5953 samples by removing 482 data points. It indicates that **there were quite a few outliers in the sample.**



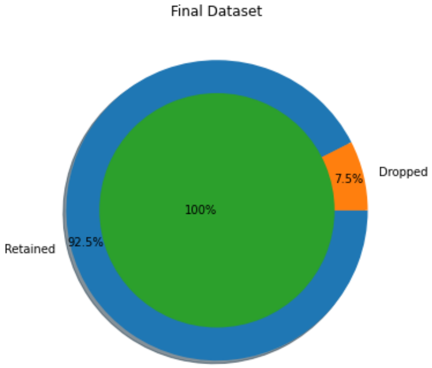
```
In [19]:
#Removal of outlier:

df1 = df3.copy()

#features1 = [i for i in features if i not in ['CHAS','RAD']]
features1 = nf

for i in features1:
    Q1 = df1[i].quantile(0.25)
    Q3 = df1[i].quantile(0.75)
    IQR = Q3 - Q1
    df1 = df1[df1[i] <= (Q3+(1.5*IQR))]
    df1 = df1[df1[i] >= (Q1-(1.5*IQR))]
    df1 = df1.reset_index(drop=True)
display(df1.head())
print('\n\033[1mInference:\033[0m\nBefore removal of outliers, The dataset had {} samples.'.format(d
f3.shape[0]))
print('After removal of outliers, The dataset now has {} samples.'.format(df1.shape[0]))
```

	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	year_2011	year_2012	weekday_1	week
0	1643690.90	0	42.31	2.572	211.096358	8.106	0	0	0	0
1	1641957.44	1	38.51	2.548	211.242170	8.106	0	0	0	0
2	1611968.17	0	39.93	2.514	211.289143	8.106	0	0	0	0
3	1409727.59	0	46.63	2.561	211.319643	8.106	0	0	0	0
4	1554806.68	0	46.50	2.625	211.350143	8.106	0	0	0	0

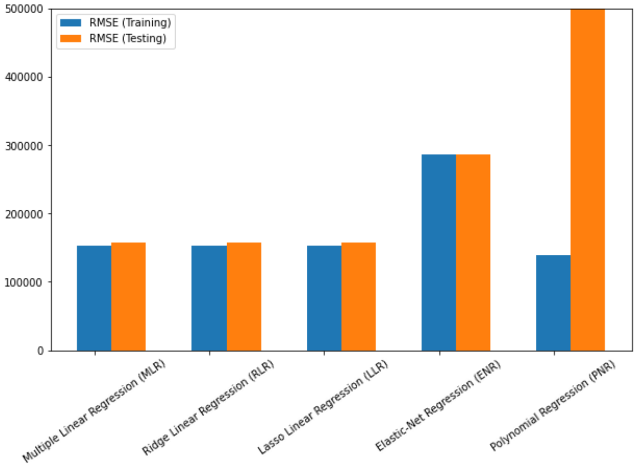
5 rows × 69 columns

```
Inference:
Before removal of outliers, The dataset had 6435 samples.
After removal of outliers, The dataset now has 5953 samples.
```

3. **Final Dataset size after performing Preprocessing -** the cleansing procedure, 482 samples had been thrown away while 7.49% of the data was kept.



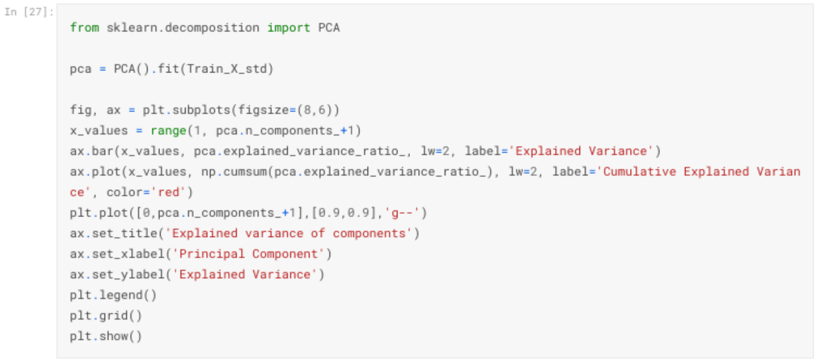
4. Using R2 and RMSE scores, Yasser compared five regression models (MLR, RLR, LLR, ENR, and PNR) on the Walmart dataset. With the highest R2 scores and the lowest RMSE values for both the training and testing sets, **MLR beat all other models, suggesting the best fit and prediction accuracy**. Despite having outstanding training performance, the other models displayed possible overfitting on the testing set.



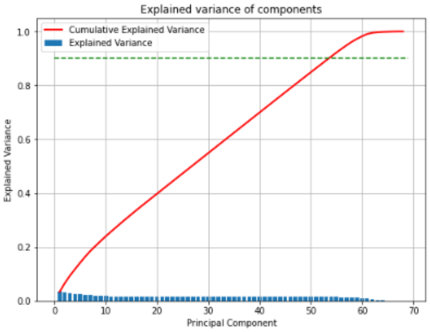
Things that I found in the code that I think not covered in the class

1. **interquartile range (IQR):** I recognized IQR as a fresh idea that the class had not yet discussed. I discovered how to calculate the interquartile range (IQR) for each feature in a dataset from YAseer's code. I was able to identify data points that differed by more than 1.5 times from the IQR in this way. These outliers were then removed from the dataset, leading to a more consistent and reliable data set for examination.

2. **Feature elimination using PCA decomposition:**



```
In [27]:
from sklearn.decomposition import PCA

pca = PCA().fit(Train_X_std)

fig, ax = plt.subplots(figsize=(8,6))
x_values = range(1, pca.n_components_+1)
ax.bar(x_values, pca.explained_variance_ratio_, lw=2, label='Explained Variance')
ax.plot(x_values, np.cumsum(pca.explained_variance_ratio_), lw=2, label='Cumulative Explained Varian
ce', color='red')
plt.plot([0,pca.n_components_+1],[0.9,0.9],'g--')
ax.set_title('Explained variance of components')
ax.set_xlabel('Principal Component')
ax.set_ylabel('Explained Variance')
plt.legend()
plt.grid()
plt.show()
```



Principal component analysis (PCA) is used to remove useless or unnecessary data from a collection of data. Because of the decrease in data noise, this could improve the performance of machine learning models.

On the Walmart dataset, Yasser's code above code applies **feature elimination using PCA**. A PCA object is first created and fitted to the standardized training data. After that, t**he total**

explained variance curve is plotted to show which key elements explain the majority of the data's instability. In the end, it eliminates any main elements which are responsible for less than 90% of the data's variation.

By lowering the amount of noise in the data and enhancing the model's ability to adapt to fresh data, this procedure can be used to improve the performance of a regression model trained to forecast weekly sales for Walmart locations.

My own ideas, observations and improvements:

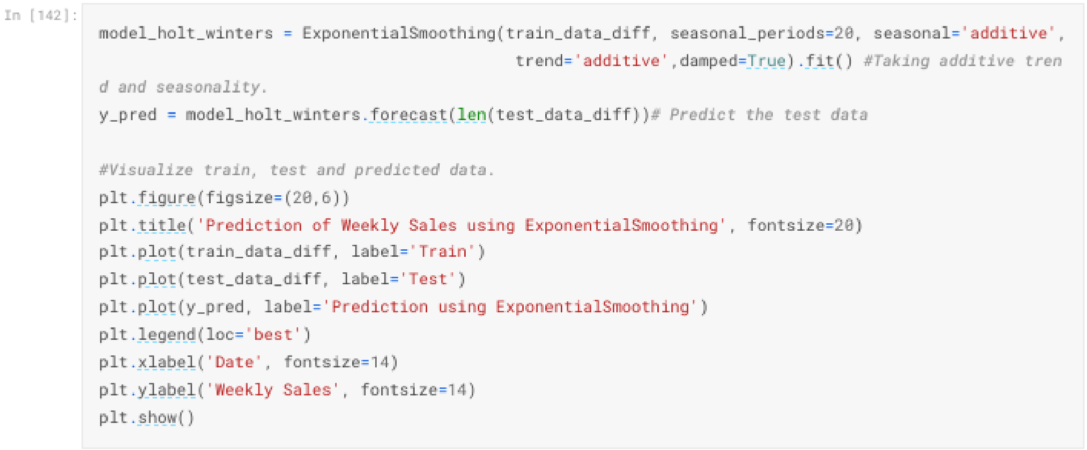
1. Data preprocessing: Though Yasseer had done feature engineering, feature scaling, and outlier removal, **he did not perform imputation to handle missing values**. Imputations such as mean imputation, median imputation, or mode imputation could be helpful.
2. Model selection: Yasseer's has used regression models - MLR, RLR, LLR, ENR, and PNR but I think that it a more systematic approach to model selection, such as **cross-validation could allows us to evaluate the performance of a model on multiple held out datasets, which can help to reduce overfitting.**
3. Hyperparameter tuning: Yasseer's code currently uses the default hyperparameters for all of the regression models but I think that **it would be beneficial to tune the hyperparameters of each model to improve its performance**. Grid search or random search could be very helpful.
4. Use a more complex model: A more complex model, like a **gradient boosting machine or a random forest, could be able to better capture the non-linear associations in the dat**a.
5. Use a time series model: Weekly sales are the target variable in the dataset. Since sales data is frequently time-series data, it might be valuable to predict future sales using a **time-series model, like an ARIMA model**.

2. Walmart Sales Forecasting by Aslan Ahmedev
Link : https://www.kaggle.com/code/aslanahmedov/walmart-sales-forecasting

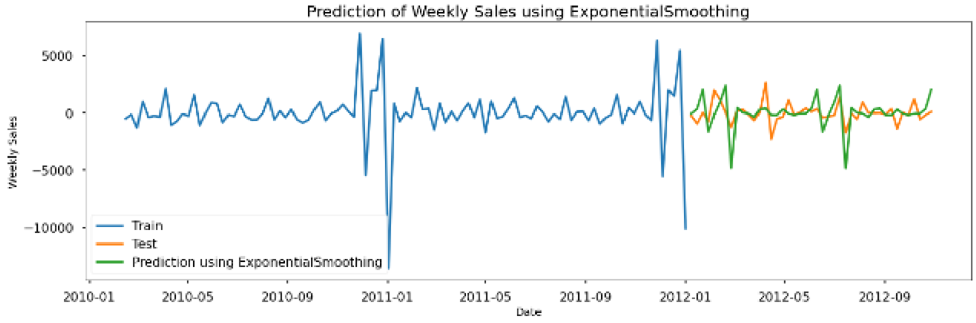
I also referred to Ashlan Ahmedov code where he performed a fantastic job of investigating the data, identifying trends, and selecting the most appropriate machine learning models. With the **Exponential Smoothing model,** he got a satisfactory outcome, and he examined the effectiveness of his model using the **weighted mean absolute error (WMAE)** measure.

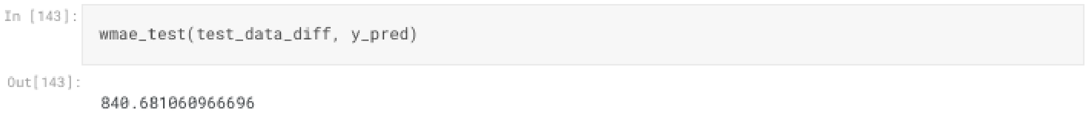
The Exponential Smoothing model, which has a **WMAE of 821**, gave him the best results. This indicates that the model has an average inaccuracy of $821 when predicting the weekly sales. This also indicates that **the model's inaccuracy is roughly 5% of the weekly average sales**.

The one thing which impressed me is selecting the holidays with highest sales and **demonstrating that holidays have a major impact on weekly sales**. By using this data, one may increase the accuracy of weekly sales forecasts and make better decisions regarding sales and managing stock.

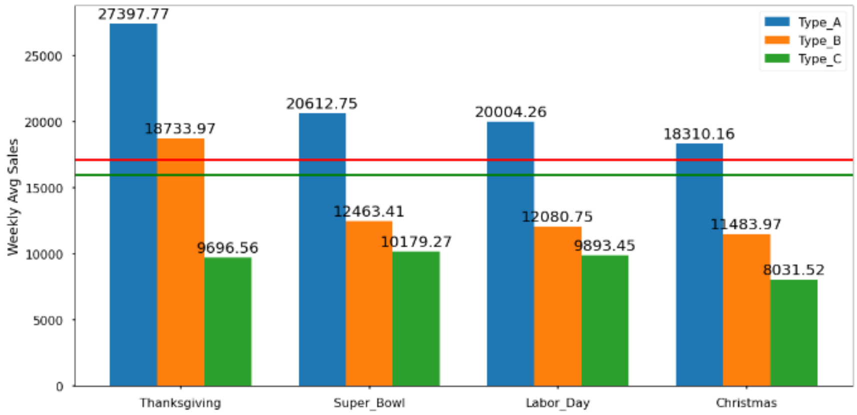


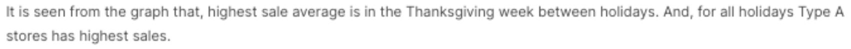
```
In [142]:
model_holt_winters = ExponentialSmoothing(train_data_diff, seasonal_periods=20, seasonal='additive',
                                           trend='additive',damped=True).fit() #Taking additive tren
d and seasonality.
y_pred = model_holt_winters.forecast(len(test_data_diff))# Predict the test data

#Visualize train, test and predicted data.
plt.figure(figsize=(20,6))
plt.title('Prediction of Weekly Sales using ExponentialSmoothing', fontsize=20)
plt.plot(train_data_diff, label='Train')
plt.plot(test_data_diff, label='Test')
plt.plot(y_pred, label='Prediction using ExponentialSmoothing')
plt.legend(loc='best')
plt.xlabel('Date', fontsize=14)
plt.ylabel('Weekly Sales', fontsize=14)
plt.show()
```





```
In [143]:
wmae_test(test_data_diff, y_pred)

Out[143]:
840.681060966696
```





It is seen from the graph that, highest sale average is in the Thanksgiving week between holidays. And, for all holidays Type A stores has highest sales.

Using automatic parameter tuning algorithms to identify the ideal parameters for his model is one important area where he could make improvements.

Overall, Ashlan performed an excellent job on his work and demonstrated that he had a solid understanding of time series forecasting and machine learning.