Part 2

Predicting future data

First we load datasets into datafremes:

```
df_stores = pd.read_csv('C:/Users/krebrovic/Desktop/Zadatak/_data/stores_dataset.csv')

df_features = pd.read_csv('C:/Users/krebrovic/Desktop/Zadatak/_data/Features_dataset.csv',
parse_dates = ['Date'])

df_sales = pd.read_csv('C:/Users/krebrovic/Desktop/Zadatak/_data/sales_dataset.csv', parse_dates =
['Date'])
```

We need to check if data overlap and then delete unnecessary data:

```
df_features = df_features[df_features.Date.dt.date <= df_sales.Date.dt.date.max()]
```

we need to merge data and check datatypes:

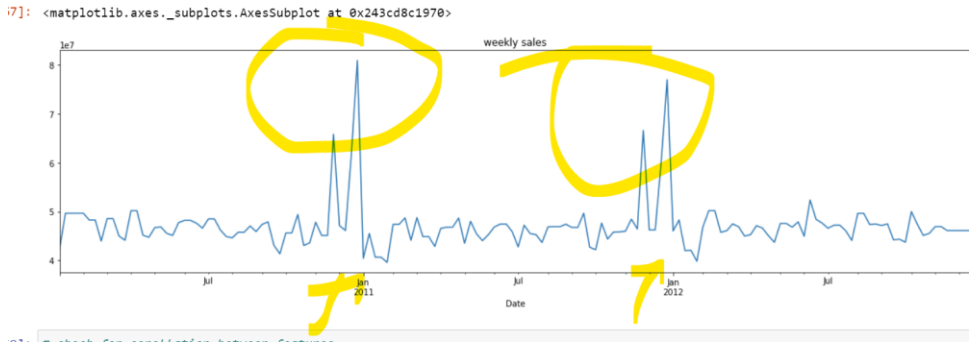
```
df_all_1 = df_features.merge(df_sales, 'right', on = ['Date', 'Store', 'IsHoliday'])

df_all = df_all_1.merge(df_stores, 'left', on = 'Store')

df_all.info()
```

check for missng data

```
df_all.isna().sum()
```

Check for seasonality and plota data

```
multi_plot = seasonal_decompose(df_by_date_new['Weekly_Sales'], model = 'add',
extrapolate_trend='freq')
```



check for correlation between features:

plt.figure(figsize=(15,8))

sns.heatmap(df_by_date_new.corr('spearman'), annot = True)

It seems that there is seasonality and we see there is correlation since it is time series, it's proven that methods like linear regression and similar methods work good for predicting future sales so I'll be using two models:

- Linear regression using Prophet

- Exponential Smoothing using Holt winters

I decided to use Linear regression because I used Prophet before and Exponential smoothing because I wanted to compare these two methods. Unfortunately, I didn’t cover any deep learning algorithm due to my challenged private time resources.

Model no.1 - Prophet

I set horizon to 8 weeks(56 days), took a sample of stores in a loop because I thought that looping through all stores would be an overkill in this assignment.

Code:

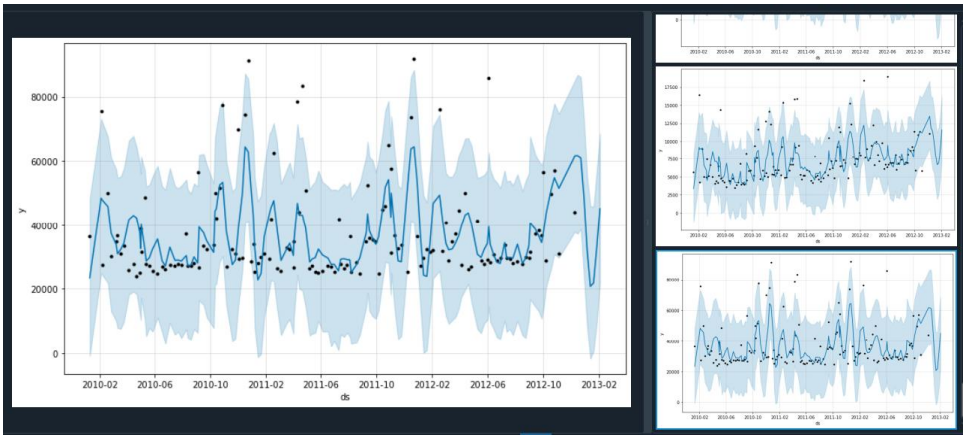
```
for x in [1,2,3,4]:
    store = df_all[(df_all.Store == x) & (df_all.Dept == 1)].sort_values('Date')
    df_all_p = store[['Date', 'Weekly_Sales']]
    df_all_p.columns = ['ds', 'y']
    model = Prophet(interval_width=0.95,daily_seasonality=False,
          weekly_seasonality=True)
# Fitting
    model.fit(df_all_p)
    future= model.make_future_dataframe(periods=8,freq='W')
#future.columns = ['ds']
#future['ds']= to_datetime(future['ds'])

# Prediction
    forecast = model.predict(future)
```

Results

Graphs for selected stores show that it is expected that in the period between 2013-01-06 and 2013-01-20 there is weak demand for products. That is due to the end of holyday time(Christmas and New Year). Demand again rises at the end of the January and in the beginning of the February.

For instance:

```
[8 rows x 19 columns]
           ds          yhat    yhat_lower    yhat_upper
150 2013-02-03  45108.128664  22065.778539  68622.480391
149 2013-01-27  32872.800061   9269.421682  56912.231226
148 2013-01-20  21785.728086    599.376457  45785.015738
147 2013-01-13  20727.218660  -1837.051294  45988.899621
146 2013-01-06  31850.921249   7849.238753  55006.361459
```



Validation of the model

By default, the initial training period is set to three times the horizon, and cutoffs are made every half a horizon. The initial period should be long enough to capture all of the components of the model, in particular seasonalities.

A forecast is made for every observed point between cutoff and cutoff + horizon. This dataframe can then be used to compute error measures of yhat vs. Y.

Code:

df_cv = cross_validation(model,horizon='56 days

df_p = performance_metrics(df_cv)

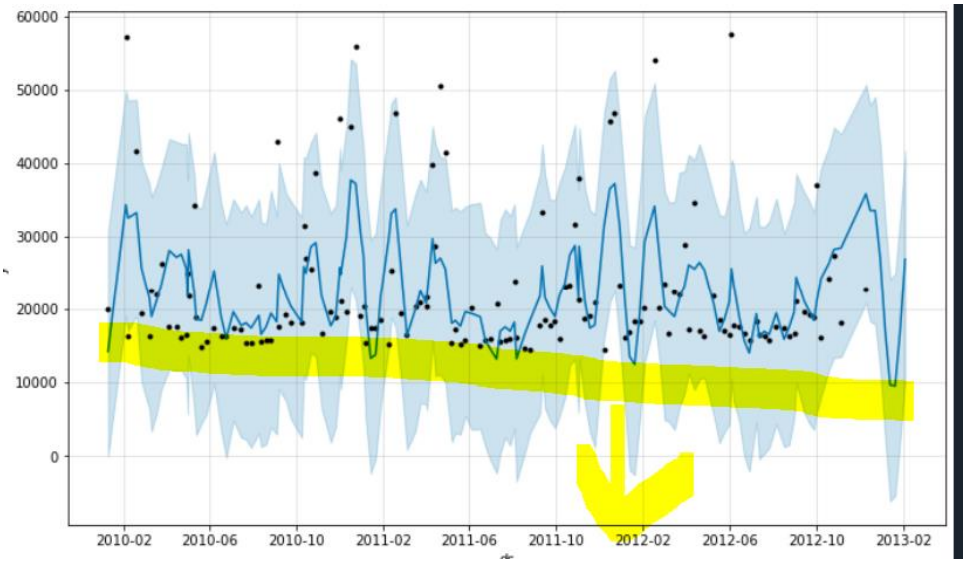
print(df_p.head())

Then I looked at MAPE indicator that is used to measure the performance of regression machine learning models. In our case there is 20-30% error in prediction between predicted data and real data which is considered to be “OK” result (based on information available on internet).

```
   horizon           mse          rmse           mae      mape     mdape  \
0   4 days  3.716361e+08  19277.865150  10446.887920  0.336053  0.154978
1   5 days  1.064944e+08  10319.611691   7775.324869  0.228400  0.154978
2   6 days  9.926028e+07   9962.945325   7206.519273  0.209565  0.144684
3   7 days  3.792554e+08  19474.480708  12315.887597  0.260098  0.144684
4  10 days  4.501476e+08  21216.681757  14226.215280  0.309675  0.154978

    coverage
0   0.941176
1   0.941176
2   0.941176
3   0.823529
4   0.764706
```

In some of the given stores, there is a sales downtrend where sales drops every year, so it is expected to drop even more.

I'd go further and analyze possible reasons for that, maybe CPI is dropping, maybe there is something else. Maybe some unsupervised deep learning algorithm could help us with that.



Conclusion for this model:

There is no strong correlation between variables, f-statistic is not so good, but MAPE shows predictions are "not great, not terrible". There is strong seasonality so we can expect trends to continue. We should analyze more data, on Store level to come to more clear conclusions.

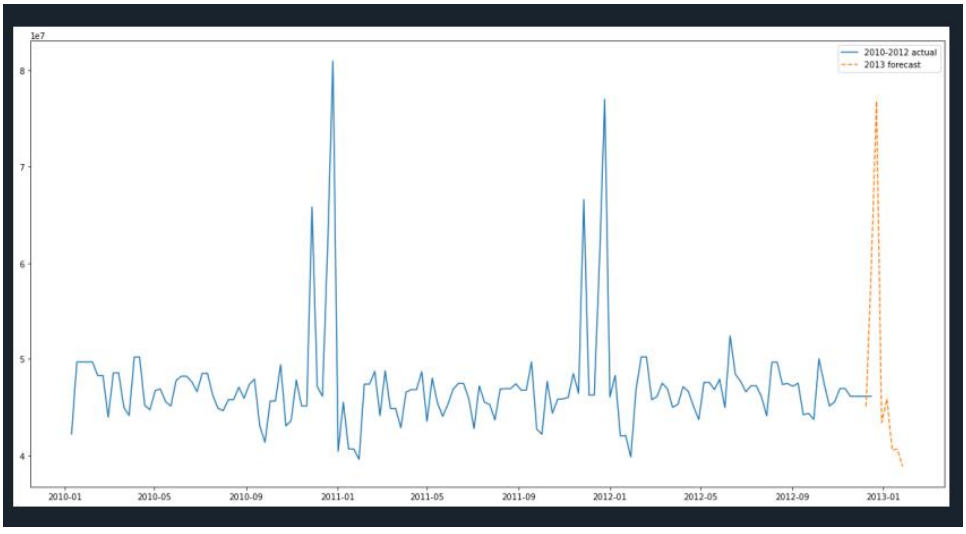
Model no.2 Holt Winters

We make future prediction on 8 weeks period using Exponential smoothing. Exponential smooting is method of forecasting that compares your prior *forecast* with your *actual* and then applies the difference between the two to the next forecast. Regression aims to fit a function to your data that gives you the "best fit". To predict future, Exponential smoothing uses past predictions and actual data and regression uses only past data.

Code:

```
fit_model = ExponentialSmoothing(df_by_date_new['Weekly_Sales'][:-2],
                                 trend = 'add',
                                 seasonal = 'add',
                                 seasonal_periods = 52).fit()


future_prediction = fit_model.forecast(8)

future_prediction
```

Result:



Validation

Lets calculate MAPE.

Code:

```
def mean_absolute_percentage_error(y_true, y_pred):
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

print("Mean Absolute Percentage Error =
{a}%".format(a=mean_absolute_percentage_error(df_by_date_new.Weekly_Sales[120:],future_predicti
on)))
```

MAPE is approximately 15 percent on all data.

MAPE on store level is little more problematic, it's stands between 40 and 50% which is not a good result.

Comparison

Lets see comparison Prophet vs Holt Winters

On Store level, Store 1, Department 1

Date	HW	Prophet
2013-01-27	26586.598812	16998.987845
2013-01-20	25789.771662	9508.202257
2013-01-13	24628.995640	9637.438609
2013-01-06	18696.517310	17154.743340
2012-12-30	21829.776121	27078.901297
2012-12-23	52781.105294	33565.676715
2012-12-16	45980.560728	33572.034994
2012-12-10	18013.427836	35847.147237

From this we can see that those two methods can give us completely different picture. It's due to the difference between exponential smoothing and linear regression. On store level, HW is performing a lot worse because of the ponder that this method have on historic data, new data is valued more than the old, while Prophet derives a function that values all the data equally.

Conclusion about the data and sales

I would say that datasets provided don't give us a lot of information to grasp on. There is also no meaningful correlation between features but there is obvious seasonality on big events such as Christmas and New Year, Superbowl, etc. That is expected and in some way models that use mathematical functions(Regresion,ARIMA,ARMA,Smoothing) can predict trend and seasonality, for example if there is obvius decline in sales through time, functions will predict further decline.

More meaningful connections between variables could have been discovered with deep learning models.

Regarding data, I'd say we have too little information about sales, It would be better if we take more detailed data from stores, statistics on level of product (which products are bought more) and take into consideration age and demographics, the more we know about the customer, the better predictions would be. We would also need data about competition, maybe the new "players" are coming into town causing our sales to shrink. Economic situation is another big factor, during recession periods sales are likely to go down and it is all about cycles.