```
import pandas as pd
import numpy as np
from  statsmodels.tsa.api import SimpleExpSmoothing, Holt
import matplotlib.pyplot as plt
import datetime
from sklearn.metrics import mean_squared_error

def mse(y_true, y_pred):
    return np.mean(np.square(y_true-y_pred))

class myExpSmoothing:
    def __init__(self, y, alpha):
        self.alpha = alpha
        self.y0 = y[0]
        self.data = np.zeros(len(y))
        self.data[0] = self.y0

        for i in np.arange(1,len(y)):
            self.data[i] = self.predict(y[i-1], self.data[i-1])
    def predict(self, val_real, val_pred):
        return self.alpha * val_real + (1-self.alpha) * val_pred
```

The algerian export data

```
index=pd.date_range(start="1960", end="2018", freq="Y")
df=pd.read_csv('AlgeriaExport.txt',header=None).set_index(index)
numpy_df = df.to_numpy().flatten()
numpy_df
```

```
array([39.0431726 , 46.24455689, 19.79387268, 24.68468205,
25.08405873,
       22.60394356, 25.98619752, 23.43441677, 23.1356346 ,
23.78877682,
       22.0727334 , 18.44251915, 20.44956198, 25.503663  ,
38.74904361,
       33.68893622, 33.05458393, 30.58656693, 25.53583672,
31.14830021,
       34.33846147, 34.58725077, 30.92485632, 27.94180611,
25.71001618,
       23.58393289, 12.85475734, 14.27247473, 15.50786788,
18.63926334,
       23.44368508, 29.11782217, 25.31959428, 21.783877  ,
22.53072525,
       26.19477598, 29.76044833, 30.90631138, 22.57835401,
28.1501165 ,
       42.06971832, 36.68930475, 35.50453311, 38.24882911,
40.0532265 ,
       47.20519324, 48.81068822, 47.06816355, 47.97334514,
35.37165064,
       38.44454785, 38.78695388, 36.89054757, 33.20989779,
```

```
30.21911743,
       23.17177829, 20.86001063, 22.63888685])

fig, ax = plt.subplots(2, figsize = (10,16))
plt.suptitle("Optimization of exponential smoothing forecasting via
mean squared error")

alpha_min_stat = [-1, 1e18]
alpha_min_mine = [-1, 1e18]

bestFitMine = None
bestFitStat = None
alphas = np.arange(0.05,1,0.01)

mses_stat = []
mses_mine = []

# sweep through all alphas
for alpha in alphas:
    fit =
SimpleExpSmoothing(df,initialization_method="heuristic").fit(smoothing
_level=alpha,optimized=False).fittedvalues
    fitMine = myExpSmoothing(numpy_df, alpha).data
    #fcast = fit.forecast(3).rename(f"$\alpha = {alpha} $")
    mse_stat = mean_squared_error(fit, numpy_df)
    mse_mine = mean_squared_error(fitMine, numpy_df)

    mses_stat.append(mse_stat)
    mses_mine.append(mse_mine)

    print(r"For $\alpha =$"+ f"{alpha:.3f} the mse - stat is
{mse_stat:.3f} and mse - mine is {mse_mine:.3f}")
    if mse_mine < alpha_min_mine[1]:
        alpha_min_mine = [alpha, mse_mine]
        bestFitMine = fitMine
    if mse_stat < alpha_min_stat[1]:
        alpha_min_stat = [alpha, mse_stat]
        bestFitStat = fit

# make optimized prediction from the library
fit = SimpleExpSmoothing(df,initialization_method="estimated").fit()

a=fit.model.params["smoothing_level"]
m=mse(fit.fittedvalues, numpy_df)

# plot
ax[0].plot(df.index, numpy_df, label = 'original data')
ax[0].plot(df.index, bestFitStat, '--', alpha=0.9, label = r'Stat
prediction for $\alpha$=' + f'{alpha_min_stat[0]:.3f} and
mse={alpha_min_stat[1]:.3f}')
```

```python
ax[0].plot(df.index, bestFitMine, '-', alpha=0.9, label = r'My
prediction for $\alpha$=' + f'{alpha_min_mine[0]:.3f} and
mse={alpha_min_mine[1]:.3f}')
ax[0].plot(df.index, fit.fittedvalues, ':', alpha=0.8, label =
r'Optimized prediction for $\alpha$=' + f'{a:.3f} and mse={m:.3f}')
ax[0].legend()
ax[0].set_xlabel("Year")

ax[1].plot(alphas, mses_stat, label = 'from statsmodel')
ax[1].plot(alphas, mses_mine, label = 'from mymodel')
ax[1].set_xlabel(r"$\alpha$")
ax[1].set_ylabel("mse")
ax[1].set_title(r"Model parameter $\alpha$ optimization")
```

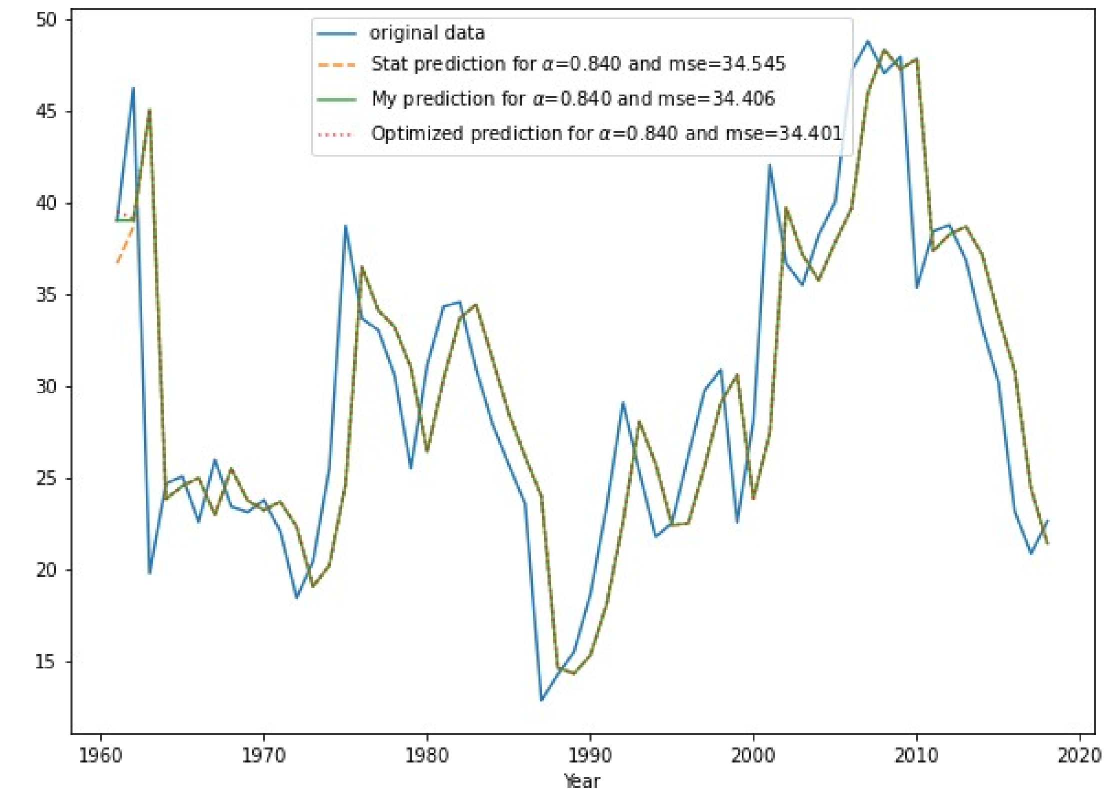
```
For $\alpha =$0.050 the mse - stat is 85.666 and mse - mine is 93.646
For $\alpha =$0.060 the mse - stat is 82.693 and mse - mine is 89.485
For $\alpha =$0.070 the mse - stat is 80.023 and mse - mine is 85.906
For $\alpha =$0.080 the mse - stat is 77.590 and mse - mine is 82.753
For $\alpha =$0.090 the mse - stat is 75.354 and mse - mine is 79.929
For $\alpha =$0.100 the mse - stat is 73.280 and mse - mine is 77.368
For $\alpha =$0.110 the mse - stat is 71.342 and mse - mine is 75.020
For $\alpha =$0.120 the mse - stat is 69.519 and mse - mine is 72.846
For $\alpha =$0.130 the mse - stat is 67.794 and mse - mine is 70.819
For $\alpha =$0.140 the mse - stat is 66.154 and mse - mine is 68.917
For $\alpha =$0.150 the mse - stat is 64.591 and mse - mine is 67.122
For $\alpha =$0.160 the mse - stat is 63.096 and mse - mine is 65.422
For $\alpha =$0.170 the mse - stat is 61.665 and mse - mine is 63.809
For $\alpha =$0.180 the mse - stat is 60.295 and mse - mine is 62.275
For $\alpha =$0.190 the mse - stat is 58.982 and mse - mine is 60.814
For $\alpha =$0.200 the mse - stat is 57.725 and mse - mine is 59.422
For $\alpha =$0.210 the mse - stat is 56.522 and mse - mine is 58.096
For $\alpha =$0.220 the mse - stat is 55.370 and mse - mine is 56.833
For $\alpha =$0.230 the mse - stat is 54.270 and mse - mine is 55.630
For $\alpha =$0.240 the mse - stat is 53.219 and mse - mine is 54.484
For $\alpha =$0.250 the mse - stat is 52.216 and mse - mine is 53.393
For $\alpha =$0.260 the mse - stat is 51.260 and mse - mine is 52.355
For $\alpha =$0.270 the mse - stat is 50.348 and mse - mine is 51.368
For $\alpha =$0.280 the mse - stat is 49.481 and mse - mine is 50.430
For $\alpha =$0.290 the mse - stat is 48.655 and mse - mine is 49.539
For $\alpha =$0.300 the mse - stat is 47.869 and mse - mine is 48.692
For $\alpha =$0.310 the mse - stat is 47.122 and mse - mine is 47.888
For $\alpha =$0.320 the mse - stat is 46.412 and mse - mine is 47.124
For $\alpha =$0.330 the mse - stat is 45.738 and mse - mine is 46.399
For $\alpha =$0.340 the mse - stat is 45.097 and mse - mine is 45.711
For $\alpha =$0.350 the mse - stat is 44.488 and mse - mine is 45.058
For $\alpha =$0.360 the mse - stat is 43.911 and mse - mine is 44.439
For $\alpha =$0.370 the mse - stat is 43.362 and mse - mine is 43.851
For $\alpha =$0.380 the mse - stat is 42.842 and mse - mine is 43.293
For $\alpha =$0.390 the mse - stat is 42.348 and mse - mine is 42.764
For $\alpha =$0.400 the mse - stat is 41.879 and mse - mine is 42.262
```

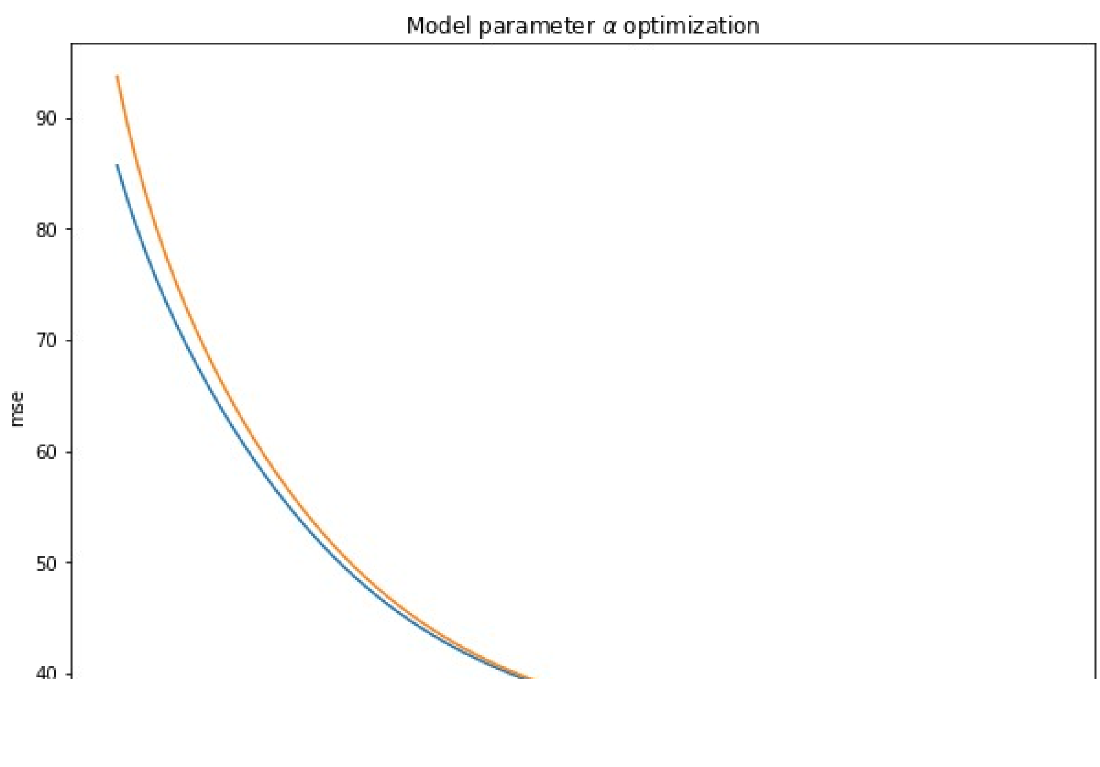
```
For $\alpha =$0.410 the mse - stat is 41.434 and mse - mine is 41.786
For $\alpha =$0.420 the mse - stat is 41.011 and mse - mine is 41.334
For $\alpha =$0.430 the mse - stat is 40.611 and mse - mine is 40.905
For $\alpha =$0.440 the mse - stat is 40.230 and mse - mine is 40.498
For $\alpha =$0.450 the mse - stat is 39.869 and mse - mine is 40.112
For $\alpha =$0.460 the mse - stat is 39.527 and mse - mine is 39.745
For $\alpha =$0.470 the mse - stat is 39.202 and mse - mine is 39.398
For $\alpha =$0.480 the mse - stat is 38.893 and mse - mine is 39.068
For $\alpha =$0.490 the mse - stat is 38.600 and mse - mine is 38.754
For $\alpha =$0.500 the mse - stat is 38.323 and mse - mine is 38.457
For $\alpha =$0.510 the mse - stat is 38.059 and mse - mine is 38.176
For $\alpha =$0.520 the mse - stat is 37.810 and mse - mine is 37.908
For $\alpha =$0.530 the mse - stat is 37.573 and mse - mine is 37.655
For $\alpha =$0.540 the mse - stat is 37.348 and mse - mine is 37.415
For $\alpha =$0.550 the mse - stat is 37.136 and mse - mine is 37.187
For $\alpha =$0.560 the mse - stat is 36.935 and mse - mine is 36.972
For $\alpha =$0.570 the mse - stat is 36.744 and mse - mine is 36.768
For $\alpha =$0.580 the mse - stat is 36.564 and mse - mine is 36.575
For $\alpha =$0.590 the mse - stat is 36.394 and mse - mine is 36.393
For $\alpha =$0.600 the mse - stat is 36.233 and mse - mine is 36.220
For $\alpha =$0.610 the mse - stat is 36.081 and mse - mine is 36.058
For $\alpha =$0.620 the mse - stat is 35.938 and mse - mine is 35.905
For $\alpha =$0.630 the mse - stat is 35.804 and mse - mine is 35.760
For $\alpha =$0.640 the mse - stat is 35.677 and mse - mine is 35.625
For $\alpha =$0.650 the mse - stat is 35.559 and mse - mine is 35.498
For $\alpha =$0.660 the mse - stat is 35.448 and mse - mine is 35.379
For $\alpha =$0.670 the mse - stat is 35.344 and mse - mine is 35.268
For $\alpha =$0.680 the mse - stat is 35.248 and mse - mine is 35.164
For $\alpha =$0.690 the mse - stat is 35.159 and mse - mine is 35.068
For $\alpha =$0.700 the mse - stat is 35.076 and mse - mine is 34.979
For $\alpha =$0.710 the mse - stat is 35.000 and mse - mine is 34.897
For $\alpha =$0.720 the mse - stat is 34.930 and mse - mine is 34.822
For $\alpha =$0.730 the mse - stat is 34.866 and mse - mine is 34.754
For $\alpha =$0.740 the mse - stat is 34.808 and mse - mine is 34.692
For $\alpha =$0.750 the mse - stat is 34.757 and mse - mine is 34.636
For $\alpha =$0.760 the mse - stat is 34.711 and mse - mine is 34.587
For $\alpha =$0.770 the mse - stat is 34.671 and mse - mine is 34.544
For $\alpha =$0.780 the mse - stat is 34.637 and mse - mine is 34.506
For $\alpha =$0.790 the mse - stat is 34.608 and mse - mine is 34.475
For $\alpha =$0.800 the mse - stat is 34.585 and mse - mine is 34.450
For $\alpha =$0.810 the mse - stat is 34.567 and mse - mine is 34.430
For $\alpha =$0.820 the mse - stat is 34.554 and mse - mine is 34.416
For $\alpha =$0.830 the mse - stat is 34.547 and mse - mine is 34.408
For $\alpha =$0.840 the mse - stat is 34.545 and mse - mine is 34.406
For $\alpha =$0.850 the mse - stat is 34.548 and mse - mine is 34.409
For $\alpha =$0.860 the mse - stat is 34.556 and mse - mine is 34.417
For $\alpha =$0.870 the mse - stat is 34.570 and mse - mine is 34.432
For $\alpha =$0.880 the mse - stat is 34.588 and mse - mine is 34.451
For $\alpha =$0.890 the mse - stat is 34.612 and mse - mine is 34.476
For $\alpha =$0.900 the mse - stat is 34.641 and mse - mine is 34.507
```

```
For $\alpha =$0.910 the mse - stat is 34.674 and mse - mine is 34.543
For $\alpha =$0.920 the mse - stat is 34.713 and mse - mine is 34.584
For $\alpha =$0.930 the mse - stat is 34.757 and mse - mine is 34.631
For $\alpha =$0.940 the mse - stat is 34.807 and mse - mine is 34.684
For $\alpha =$0.950 the mse - stat is 34.861 and mse - mine is 34.742
For $\alpha =$0.960 the mse - stat is 34.920 and mse - mine is 34.805
For $\alpha =$0.970 the mse - stat is 34.985 and mse - mine is 34.874
For $\alpha =$0.980 the mse - stat is 35.055 and mse - mine is 34.949
For $\alpha =$0.990 the mse - stat is 35.130 and mse - mine is 35.029
```

```
Text(0.5, 1.0, 'Model parameter $\\alpha$ optimization')
```

Optimization of exponential smoothing forecasting via mean squared error

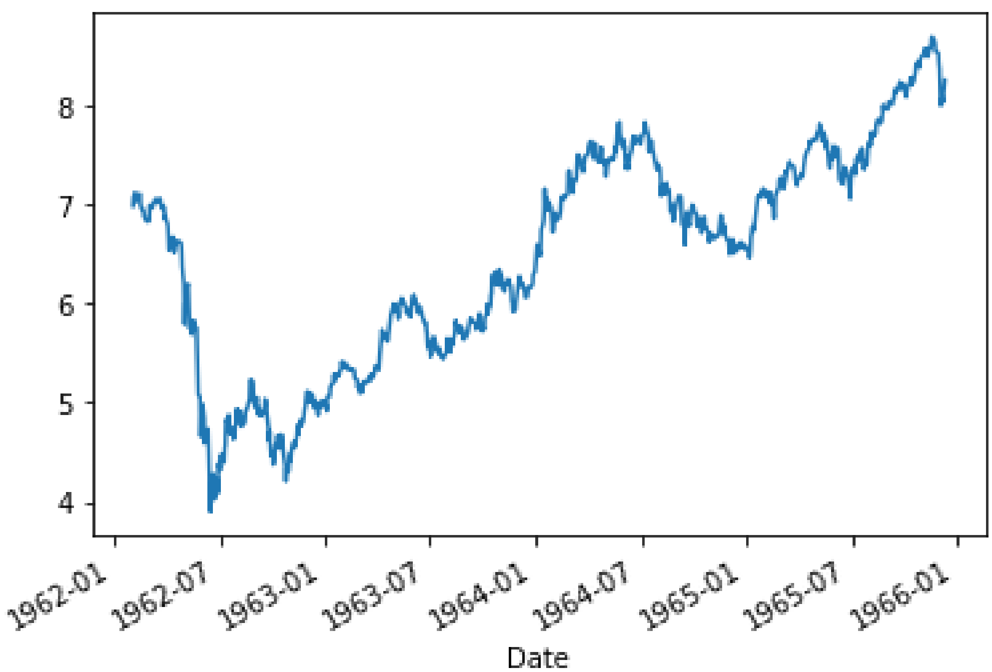




```
ibm = pd.read_csv('IBM.csv', index_col=['Date'],parse_dates = True)
ibm['Open'].plot()
#
ibm
```

```
                Open      High       Low     Close  Adj Close   Volume
Date
1962-02-01  6.978967  7.087317  6.978967  7.068196   1.577106   674670
1962-02-02  7.068196  7.112811  7.036329  7.112811   1.587062   533460
1962-02-05  7.112811  7.112811  6.985341  7.023582   1.567152   329490
1962-02-06  7.023582  7.036329  6.998088  7.029955   1.568787   274575
1962-02-07  7.036329  7.074570  7.036329  7.036329   1.570211   266730
...              ...       ...       ...       ...        ...      ...
1965-12-06  8.070427  8.102294  7.934990  7.998725   1.791781  1041816
1965-12-07  8.026609  8.142129  8.026609  8.102294   1.814982   407940
1965-12-08  8.134162  8.205864  8.134162  8.150096   1.825689   395388
1965-12-09  8.173996  8.221797  8.173996  8.217814   1.840859   332628
1965-12-10  8.245698  8.261632  8.245698  8.253665   1.848891   282420

[972 rows x 6 columns]
```

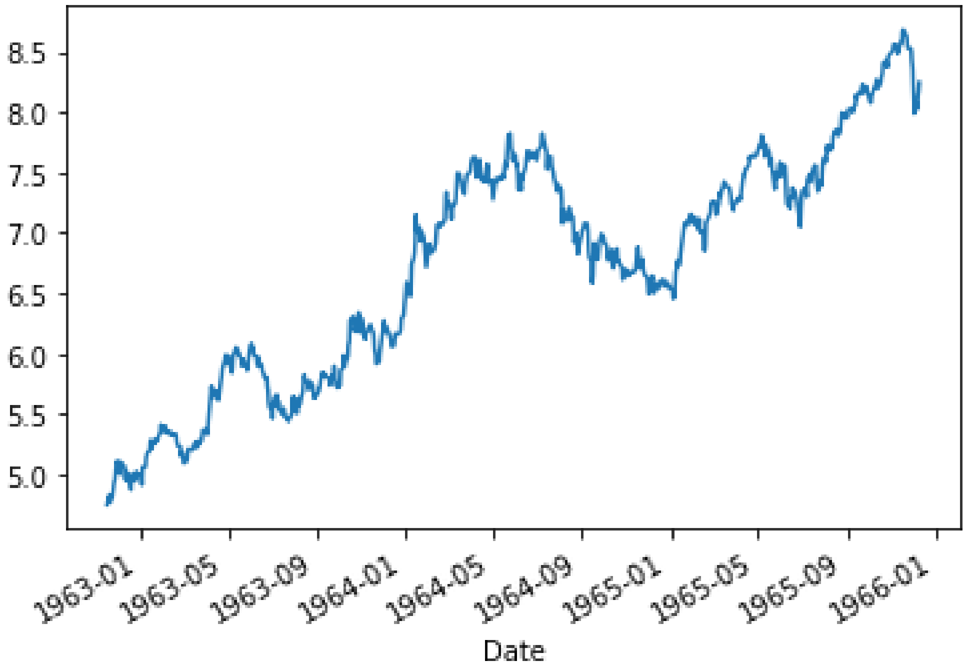



```
# linear part
timestart = datetime.datetime(1962,11,16)

timeend = datetime.datetime(1966,1,1)
print(timestart)
ibm_linear = ibm[timestart:timeend]
ibm_linear['Open'].plot()#.loc[mask]['Open'].plot()
```

```
1962-11-16 00:00:00
```

```
<AxesSubplot:xlabel='Date'>
```




```python
#set new frequency with ffill to get rid of the nans
ibmD = ibm_linear.resample('B').fillna('ffill')

fitSES =
SimpleExpSmoothing(ibm_linear['Open'],initialization_method="estimated
").fit()
fitHOLT = Holt(ibm_linear['Open'], exponential=True,
initialization_method="estimated").fit()
fittedSES = fitSES.fittedvalues
fittedHOLT = fitHOLT.fittedvalues
mseSes = mean_squared_error(ibm_linear['Open'], fittedSES)
mseHolt = mean_squared_error(ibm_linear['Open'], fittedHOLT)
print(
    f"\t->mse with SES = {mseSes}\n",
    f"\t->mse with Holt = {mseHolt}\n\n",
    f"\t{'Holt ' if mseSes>mseHolt else 'SES'} WINS!!!"
    )
fig,ax = plt.subplots()
ibm_linear['Open'].plot(style = '.-',ax = ax)
ax.plot(fittedSES, '.--', alpha=0.7, label = r'SES prediction')
ax.plot(fittedHOLT, '-', alpha=0.6, label = r'Holt prediction')
ax.legend()
#ax.set_xlabel("Year")
```

```
        ->mse with SES = 0.003608269773944319
        ->mse with Holt = 0.0035917694786575293
```

```
     Holt  WINS!!!
```

```
C:\Users\maxgr\anaconda3\lib\site-packages\statsmodels\tsa\base\
tsa_model.py:581: ValueWarning: A date index has been provided, but it
has no associated frequency information and so will be ignored when
e.g. forecasting.
  warnings.warn('A date index has been provided, but it has no'
C:\Users\maxgr\anaconda3\lib\site-packages\statsmodels\tsa\base\
tsa_model.py:581: ValueWarning: A date index has been provided, but it
has no associated frequency information and so will be ignored when
e.g. forecasting.
  warnings.warn('A date index has been provided, but it has no'
```

```
<matplotlib.legend.Legend at 0x22b92d4d0d0>
```