Exponential Smoothing Assignment
Merve Pakcan

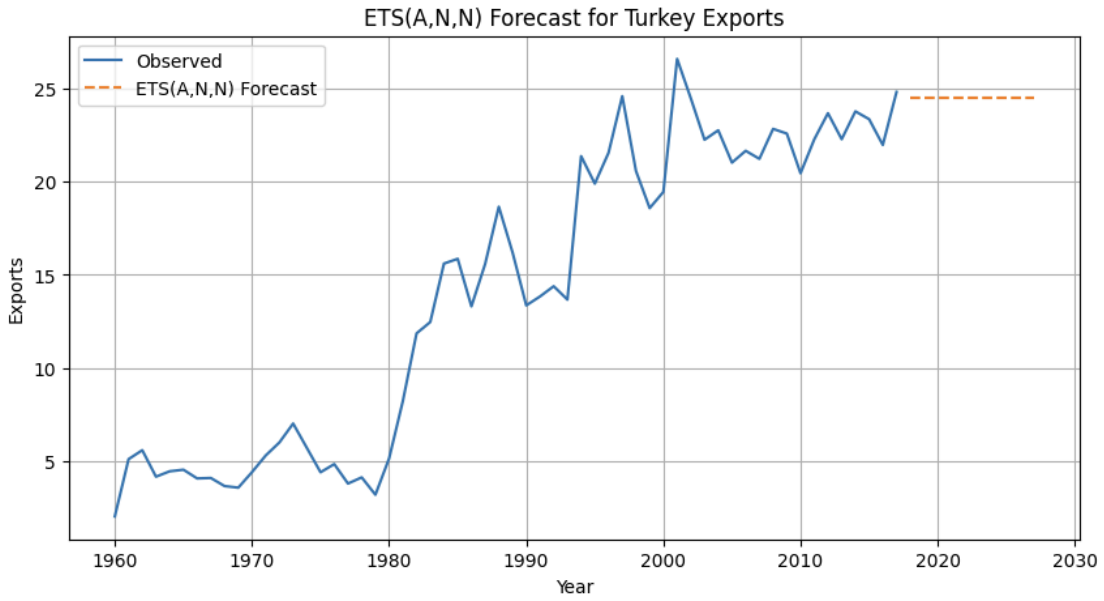
In this study, Turkey was selected to analyse the export performance of my home country within the global economy. The aim of this choice is to apply and compare exponential smoothing models to a real-world data set, in line with the objectives of the exercise.

1-Plot the Exports series and discuss the main features of the data.

The graph shows the annual exports of Turkey. The series generally increases over time, with a sharp rise around the early 2000s, followed by a slight decline and then a slower increase in later years. There are clear ups and downs between years. Since the data is annual, there is no seasonal pattern. Overall, the export series appears suitable for time series forecasting.

2- Use an ETS(A,N,N) model to forecast the series, and plot the forecasts.

The analysis in this all assignment done according to the book which is exponential smoothing framework described in ***Chapter 8 of*** **Forecasting: Principles and Practice**(https://otexts.com/fppy/nbs/08-exponential-smoothing.html#sec-expsmooth-exercises). *The* ***AutoETS*** *method was used to implement the ETS models discussed in the book*



*This model does not include a trend component, the forecasts remain constant and stay close to the last observed value, around **24.5.** This indicates that the model expects future exports to stay at a similar level.*

3- Compute the RMSE values for the training data.

3 - Compute the RMSE values for the training data.

```
from statsforecast import StatsForecast
from statsforecast.models import AutoETS

model_ann = AutoETS(
    season_length=1,
    model="ANN"
)

sf = StatsForecast(
    models=[model_ann],
    freq="YE"
)
```

```
sf.fit(sf_data)
```

```
StatsForecast(models=[AutoETS])
```

```
_ = sf.forecast(
    df=sf_data,
    h=1,
    fitted=True
)
```

```
insample = sf.forecast_fitted_values()
insample.head()
```

	unique_id	ds	y	AutoETS
0	Turkey	1960-01-01	2.055800	2.430276
1	Turkey	1961-01-01	5.124654	2.098310
2	Turkey	1962-01-01	5.603985	4.781111
3	Turkey	1963-01-01	4.184549	5.510574
4	Turkey	1964-01-01	4.473161	4.335076

Next steps: Generate code with insample | New interactive sheet

```
import numpy as np
y_true = sf_data["y"].values
y_hat  = insample["AutoETS"].values

rmse_ann = np.sqrt(np.mean((y_true - y_hat)**2))
rmse_ann
```

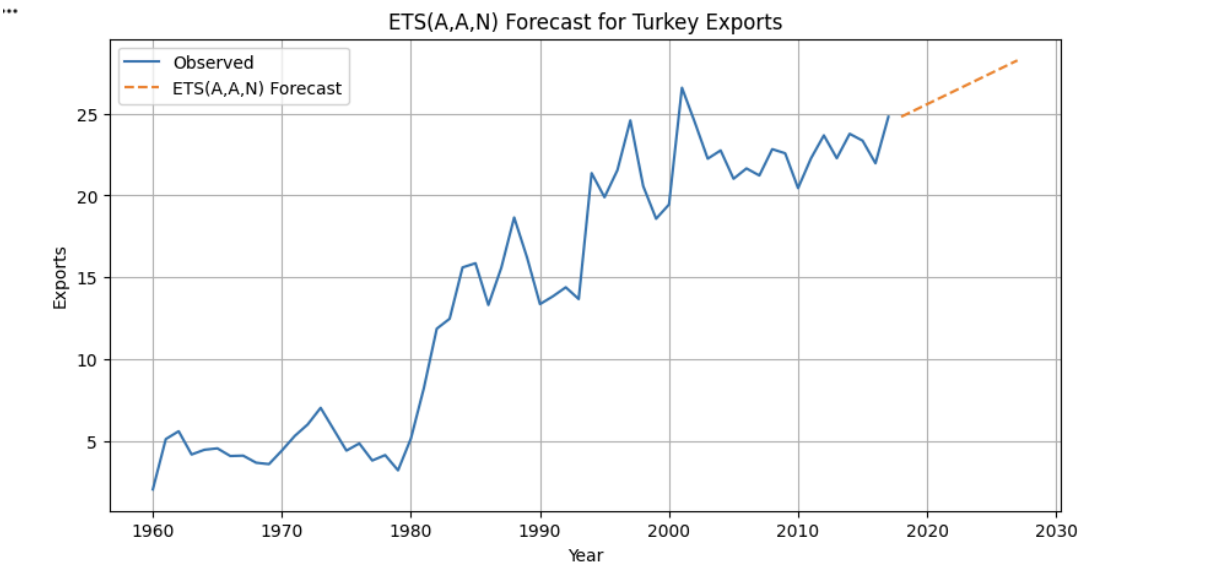
```
np.float64(2.183256424996884)
```

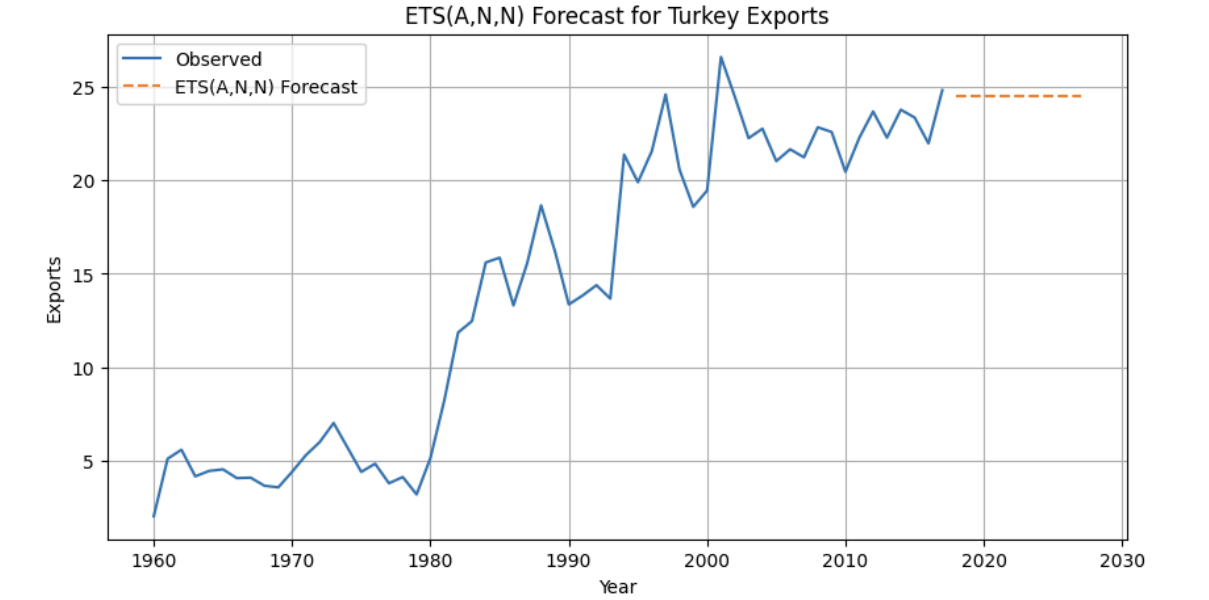
The RMSE of the ETS(A,N,N) model was calculated using one-step-ahead forecasts obtained from the AutoETS method in StatsForecast. The training RMSE value is 2.18, which shows that the model fits the data well.

4-Compare the results to those from an ETS(A,A,N) model. (Remember that the trended model is using one more parameter than the simpler model.) Discuss the merits of the two forecasting methods for this data set.

The ETS(A,A,N) model has a slightly lower training RMSE (~2.14) than the ETS(A,N,N) model (2.18).The ETS(A,A,N) model fits the data slightly better because it includes a trend. But, the improvement is very small.

5- Compare the forecasts from both methods. Which do you think is best?





When comparing the forecasts, the ETS(A,N,N) model produces a flat forecast that stays close to the most recent observed values,while ETS(A,A,N) model shows an upward trend into the future.

According to historical data, the data moves up and down over time, and in the last years exports do not clearly keep increasing. Although the ETS(A,A,N) model includes a trend and has a slightly lower RMSE, the difference is very small. For this reason, I think the

simpler ETS(A,N,N) model seems more appropriate, it produces more stable and safer forecasts.

6-Calculate a 95% prediction interval for the first forecast for each model, using the RMSE values and assuming normal errors. Compare your intervals with those produced using `StatsForecast`.

```
yhat_ann = fc["ETS_ANN"].iloc[0]
yhat_aan = fc["ETS_AAN"].iloc[0]

yhat_ann, yhat_aan
```

```
(np.float64(24.500118185909812), np.float64(24.80570051617729))
```

```
import numpy as np

# First forecast value
yhat_ann = fc["ETS_ANN"].iloc[0]

# 95% PI using RMSE
lower_ann = yhat_ann - 1.96 * rmse_ann
upper_ann = yhat_ann + 1.96 * rmse_ann

lower_ann, upper_ann
```

```
(np.float64(20.22093559291592), np.float64(28.779300778903703))
```

```
# First forecast value
yhat_aan = fc["ETS_AAN"].iloc[0]

# 95% PI using RMSE
lower_aan = yhat_aan - 1.96 * rmse_aan
upper_aan = yhat_aan + 1.96 * rmse_aan

lower_aan, upper_aan
```

```
(np.float64(20.617172644689603), np.float64(28.994228387664975))
```

I computed a 95% prediction interval for the first forecast of each model.
For ETS(A,N,N), the first forecast is about **24.5** and the 95% interval is **[20.22, 28.78]**.
For ETS(A,A,N), the first forecast is about **24.8** and the 95% interval is **[20.62, 28.99]**.

```python
from statsforecast import StatsForecast
from statsforecast.models import AutoETS

# ETS(A,N,N)
model_ann = AutoETS(
    season_length=1,
    model="ANN",
    alias="ETS_ANN"
)

# ETS(A,A,N)
model_aan = AutoETS(
    season_length=1,
    model="AAN",
    alias="ETS_AAN"
)

# Initializing StatsForecast with both models
sf = StatsForecast(
    models=[model_ann, model_aan],
    freq="YE"
)

# Fitting the models to the data
sf.fit(sf_data)

# Generating forecasts for the next 10 years
fc = sf.forecast(df=sf_data, h=10, level=[95])
fc.head()
```

	unique_id	ds	ETS_ANN	ETS_ANN-lo-95	ETS_ANN-hi-95	ETS_AAN	ETS_AAN-lo-95	ETS_AAN-hi-95
0	Turkey	2017-12-31	24.500118	20.065360	28.934876	24.805701	20.382203	29.229198
1	Turkey	2018-12-31	24.500118	18.573696	30.426540	25.188014	19.466403	30.909625
2	Turkey	2019-12-31	24.500118	17.388306	31.611931	25.570328	18.794661	32.345994
3	Turkey	2020-12-31	24.500118	16.374032	32.626204	25.952641	18.265913	33.639370
4	Turkey	2021-12-31	24.500118	15.473011	33.527226	26.334955	17.834057	34.835852

As you can see above, it is really so close with the intervals with StatsForecast before.