

In [1]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
import numpy as np
import pandas as pd
```

In [3]:

```
df = pd.read_csv('/content/Airline_Passangers.csv')
```

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
Month      0
Passengers 0
dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

```
(144, 2)
```

Building ARMA(p,q) model:

- order = (1,0,1)

In [6]:

```
order = (1,0,1)
```

In [7]:

```
from statsmodels.tsa.arima_model import ARIMA
```

In [10]:

```
model = ARIMA(df.Passengers[:50],order=order)
```

In [11]:

```
ARMA_model = model.fit()
```

In [12]:

```
ARMA_model.summary2()
```

Out[12]:

Model:	ARMA	BIC:	428.5331
Dependent Variable:	Passengers	Log-Likelihood:	-206.44
Date:	2022-08-29 18:56	Scale:	1.0000
No. Observations:	50	Method:	css-mle
Df Model:	3	Sample:	0
Adjusted R-squared:	0.47		0

DT Residuals:	47	0
Converged:	1.0000	S.D. of innovations: 14.770
No. Iterations:	17.0000	HQIC: 423.797
AIC:	420.8850	

	Coef.	Std.Err.	t	P> t	[0.025	0.975]
const	158.8138	15.1420	10.4883	0.0000	129.1361	188.4916
ar.L1.Passengers	0.8397	0.0843	9.9666	0.0000	0.6746	1.0048
ma.L1.Passengers	0.2816	0.1407	2.0013	0.0454	0.0058	0.5574

	Real	Imaginary	Modulus	Frequency
AR.1	1.1909	0.0000	1.1909	0.0000
MA.1	-3.5510	0.0000	3.5510	0.5000

In [14]:

```
def get_mape(actual, pred):
    mape = np.round(np.mean(np.abs(100*(actual-pred)/actual)),2)
    return mape
```

In [16]:

```
from sklearn.metrics import mean_squared_error

def get_rmse(actual, pred):
    rmse = np.round(np.sqrt(mean_squared_error(actual, pred)),2)
    return rmse
```

In [18]:

```
ARMA_actual = df.Passengers[50:60]
ARMA_forecast = ARMA_model.predict(50,60)
```

In [21]:

```
df_ARMA = pd.DataFrame({'actual':ARMA_actual, 'pred':np.round(ARMA_forecast,0)})
df_ARMA
```

Out[21]:

	actual	pred
50	236.0	192.0
51	235.0	186.0
52	229.0	182.0
53	243.0	178.0
54	264.0	175.0
55	272.0	173.0
56	237.0	170.0
57	211.0	169.0
58	180.0	167.0
59	201.0	166.0
60	NaN	165.0

In [24]:

```
ARMA_mape = get_mape(df_ARMA.actual[:9],df_ARMA.pred[:9])
ARMA_mape
```

Out[24]:

23.59

In [26]:

```
ARMA_rmse = get_rmse(df_ARMA.actual[:9],df_ARMA.pred[:9])  
ARMA_rmse
```

Out[26]:

62.29

In []: