In [ ]:
```
import pandas as pd
import statsmodels.api as sm
```

In [ ]:
```
import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)
```

1.) Import Data from FRED

In [ ]:
```
data = pd.read_csv("TaylorRuleData.csv", index_col = 0)
```

In [ ]:
```
data.index = pd.to_datetime(data.index)
```

In [ ]:
```
data.dropna(inplace = True)
```

In [ ]:
```
data.head()
```

Out[ ]:

	FedFunds	Unemployment	HousingStarts	Inflation
1959-01-01	2.48	6.0	1657.0	29.01
1959-02-01	2.43	5.9	1667.0	29.00
1959-03-01	2.80	5.6	1620.0	28.97
1959-04-01	2.96	5.2	1590.0	28.98
1959-05-01	2.90	5.1	1498.0	29.04

2.) Do Not Randomize, split your data into Train, Test Holdout

In [ ]:
```
split1 = int(len(data)*0.6)
split2 = int(len(data)*0.9)
data_in = data[:split1]
data_out = data[split1:split2]
data_hold = data[split2:]
```

In [ ]:
```
X_in = data_in.iloc[:,1:]
y_in = data_in.iloc[:,0]
X_out = data_out.iloc[:,1:]
y_out = data_out.iloc[:,0]
X_hold = data_hold.iloc[:,1:]
y_hold = data_hold.iloc[:,0]
```

In [ ]:
```
# Add Constants
X_in = sm.add_constant(X_in) # Add a constant to the model (?)
X_out = sm.add_constant(X_out)
X_hold = sm.add_constant(X_hold)
```

In [ ]:
```
X_in.head()
```

Out[ ]:

	const	Unemployment	HousingStarts	Inflation
1959-01-01	1.0	6.0	1657.0	29.01
1959-02-01	1.0	5.9	1667.0	29.00
1959-03-01	1.0	5.6	1620.0	28.97
1959-04-01	1.0	5.2	1590.0	28.98
1959-05-01	1.0	5.1	1498.0	29.04

In [ ]:
```
X_out.head()
```

Out[ ]:

	const	Unemployment	HousingStarts	Inflation
1997-12-01	1.0	4.7	1566.0	161.8
1998-01-01	1.0	4.6	1525.0	162.0
1998-02-01	1.0	4.6	1584.0	162.0
1998-03-01	1.0	4.7	1567.0	162.0
1998-04-01	1.0	4.3	1540.0	162.2

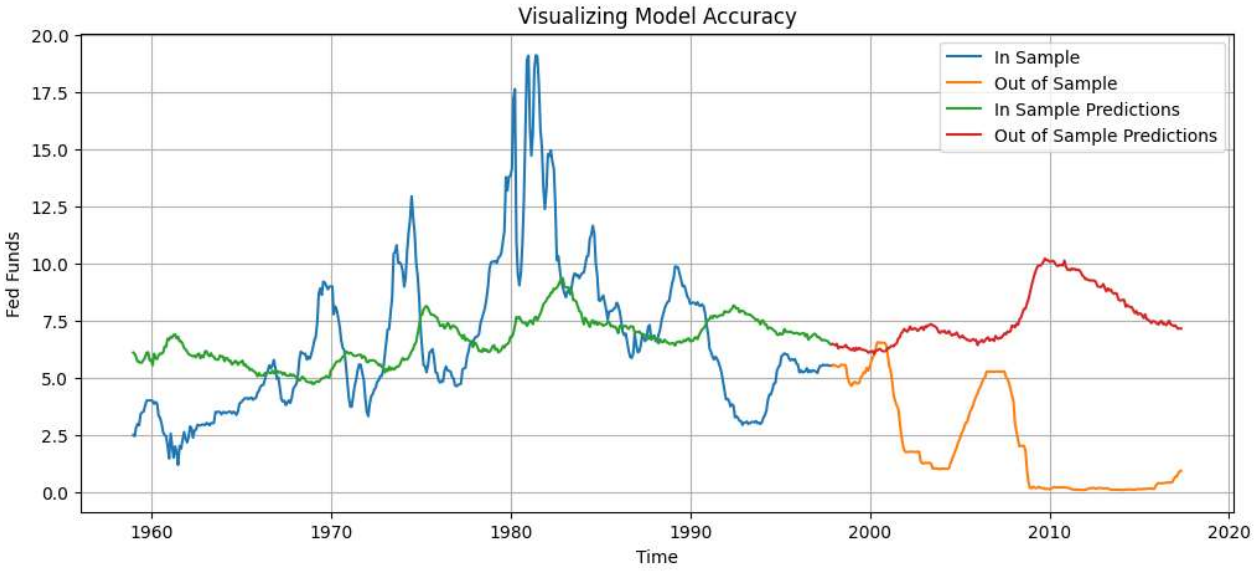
3.) Build a model that regresses FF~Unemp, HousingStarts, Inflation

In [ ]:
```
model1 = sm.OLS(y_in, X_in).fit()
```

4.) Recreate the graph fro your model

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
plt.figure(figsize = (12,5))

###
plt.plot(y_in)
plt.plot(y_out)
plt.plot(model1.predict(X_in))
plt.plot(model1.predict(X_out))
###

plt.ylabel("Fed Funds")
plt.xlabel("Time")
plt.title("Visualizing Model Accuracy")
plt.legend(["In Sample", "Out of Sample", "In Sample Predictions", "Out of Sample Predictions"])
plt.grid()
plt.show()
```



"All Models are wrong but some are useful" - 1976 George Box

5.) What are the in/out of sample MSEs

In [ ]:
```
from sklearn.metrics import mean_squared_error
```

In [ ]:
```
in_mse_1 = mean_squared_error(y_in, model1.predict(X_in))
out_mse_1 = mean_squared_error(y_out, model1.predict(X_out))
```

In [ ]:
```
print("Insample MSE : ", in_mse_1)
print("Outsample MSE : ", out_mse_1)
```

```
Insample MSE :  10.071422013168641
Outsample MSE :  40.36082783566852
```

6.) Using a for loop. Repeat 3,4,5 for polynomial degrees 1,2,3

In [ ]:
```
from sklearn.preprocessing import PolynomialFeatures
```

In [ ]:
```
max_degree = 3
```

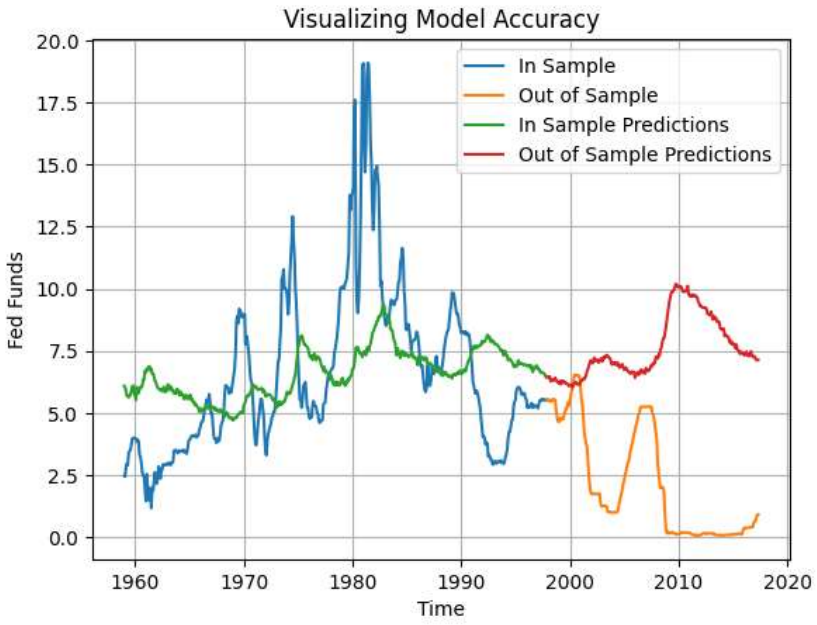
In [ ]:
```
for degree in range(1, 1+max_degree):
    print("_____________________________")
    print("DEGREES: ", degree)
    poly = PolynomialFeatures(degree = degree)
    X_in_poly = poly.fit_transform(X_in)
    X_out_poly = poly.fit_transform(X_out)

    # Q3
    model1 = sm.OLS(y_in, X_in_poly).fit()

    # Q4
```

```
    plt.figure

    in_preds = model1.predict(X_in_poly)
    in_preds = pd.DataFrame(in_preds, index = y_in.index)
    out_preds = model1.predict(X_out_poly)
    out_preds = pd.DataFrame(out_preds, index = y_out.index)

    # Q5
    in_mse_1 = mean_squared_error(y_in, model1.predict(X_in_poly))
    out_mse_1 = mean_squared_error(y_out, model1.predict(X_out_poly))
    print("Insample MSE : ", in_mse_1)
    print("Outsample MSE : ", out_mse_1)
    print("_____________________________")

    plt.plot(y_in)
    plt.plot(y_out)
    plt.plot(in_preds)
    plt.plot(out_preds)

    plt.ylabel("Fed Funds")
    plt.xlabel("Time")
    plt.title("Visualizing Model Accuracy")
    plt.legend(["In Sample", "Out of Sample", "In Sample Predictions", "Out of Sample Predictions"])
    plt.grid()
    plt.show()
```

_____

DEGREES: 1
Insample MSE : 10.071422013168641
Outsample MSE : 40.36082783566712

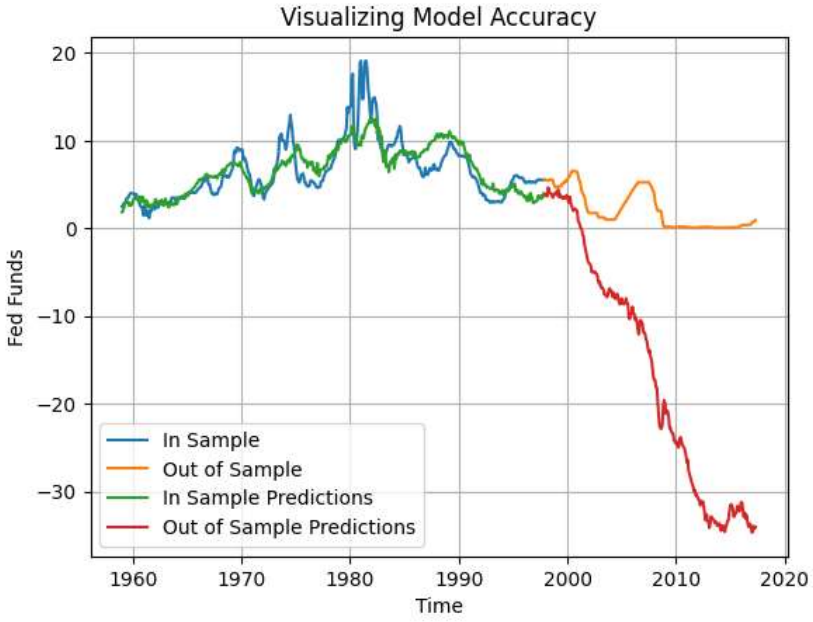
_____



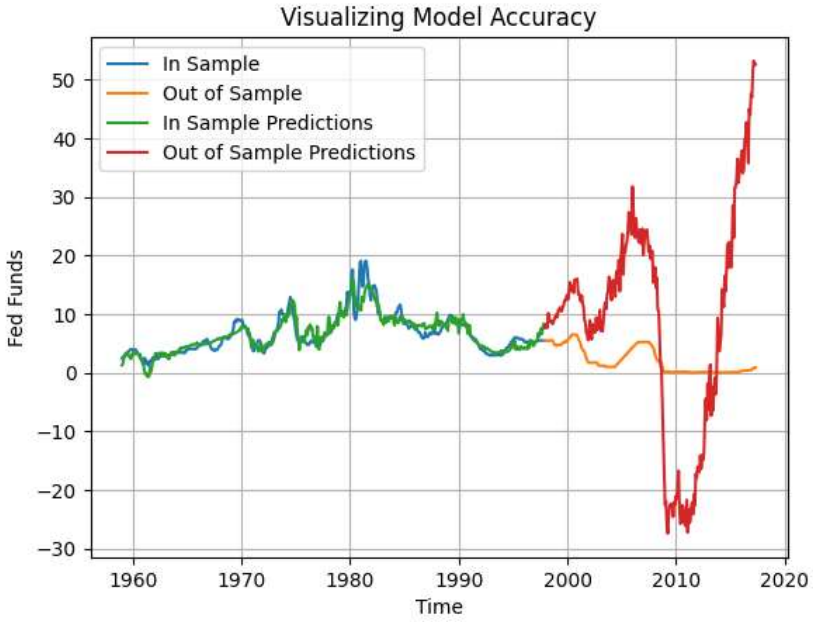
_____

DEGREES: 2
Insample MSE : 3.863477139276068
Outsample MSE : 481.44650991740434

_____



```
DEGREES:  3
Insample MSE :  1.8723636271946136
Outsample MSE :  371.76618900618945
```



7.) State your observations :

α^2

- According to the plots above, the greater the degree of the polynomial, the better the model fits the In-sample data, but the worse it fits the Out-of-sample data. This is a clear example of overfitting.
- When the degree is 2 and 3, the model is overfitting the data and cannot generalize well to out-of-sample data.
- When the degree is 1, the model is a little bit underfitting.