HW#1 Kehua Chu (UID:806153163)

In [1]:
```
import pandas as pd
import statsmodels.api as sm
```

1.) Import Data from FRED

In [2]:
```
data = pd.read_csv("TaylorRuleData.csv", index_col = 0)
```

In [3]:
```
data.index = pd.to_datetime(data.index)
```

In [4]:
```
data.dropna(inplace = True)
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 779 entries, 1959-01-01 to 2023-11-01
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   FedFunds        779 non-null    float64
 1   Unemployment    779 non-null    float64
 2   HousingStarts   779 non-null    float64
 3   Inflation       779 non-null    float64
dtypes: float64(4)
memory usage: 30.4 KB
```

In [6]:
```
data.head()
```

Out[6]:

	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 [7]:
```
split_1 = int(len(data)*0.6)
split_2 = int(len(data)*0.9)
data_in = data[:split_1]
data_out = data[split_1 : split_2]
data_hold = data[split_2:]
```

In [8]:
```
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 [9]:
```
# Add Constants
X_in = sm.add_constant(X_in)
```

```
X_out =  sm.add_constant(X_out)
X_hold =  sm.add_constant(X_hold)
```

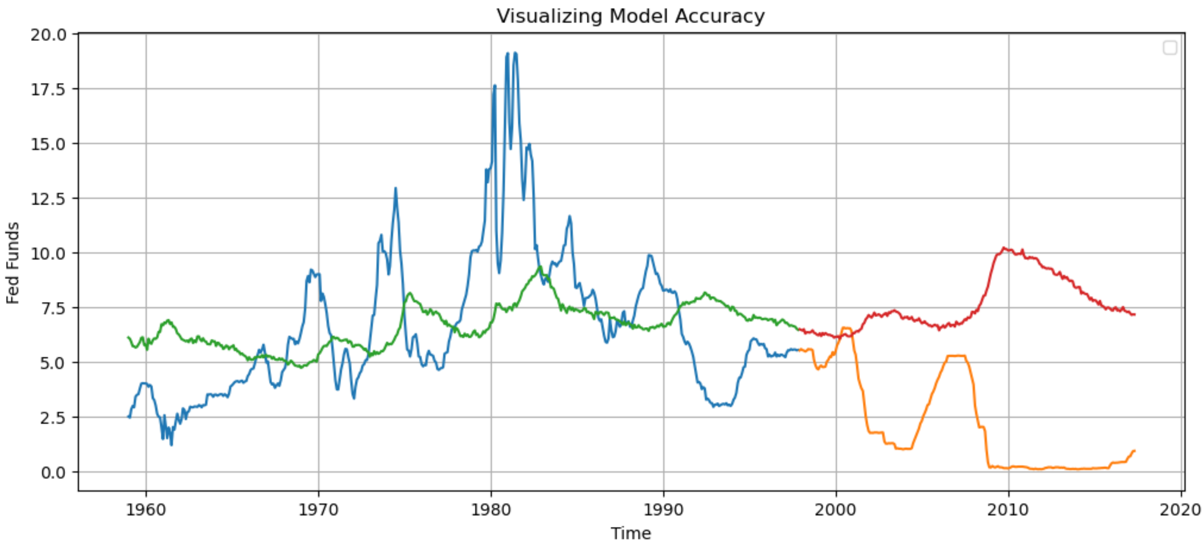
3.) Build a model that regresses FF~Unemp, HousingStarts, Inflation

In [10]:
```
model1 = sm.OLS(y_in, X_in).fit()
```

4.) Recreate the graph fro your model

In [13]:
```
import matplotlib.pyplot as plt
```

In [14]:
```
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([])
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 [15]:
```
from sklearn.metrics import mean_squared_error
```

In [16]:
```
in_mse_1 = mean_squared_error(model1.predict(X_in), y_in)
out_mse_1 = mean_squared_error(model1.predict(X_out),y_out)
```

In [17]:
```
print("Insample MSE : ", in_mse_1)
print("Outsample MSE : ", out_mse_1)
```

```
Insample MSE :  10.071422013168641
Outsample MSE :  40.36082783566751
```

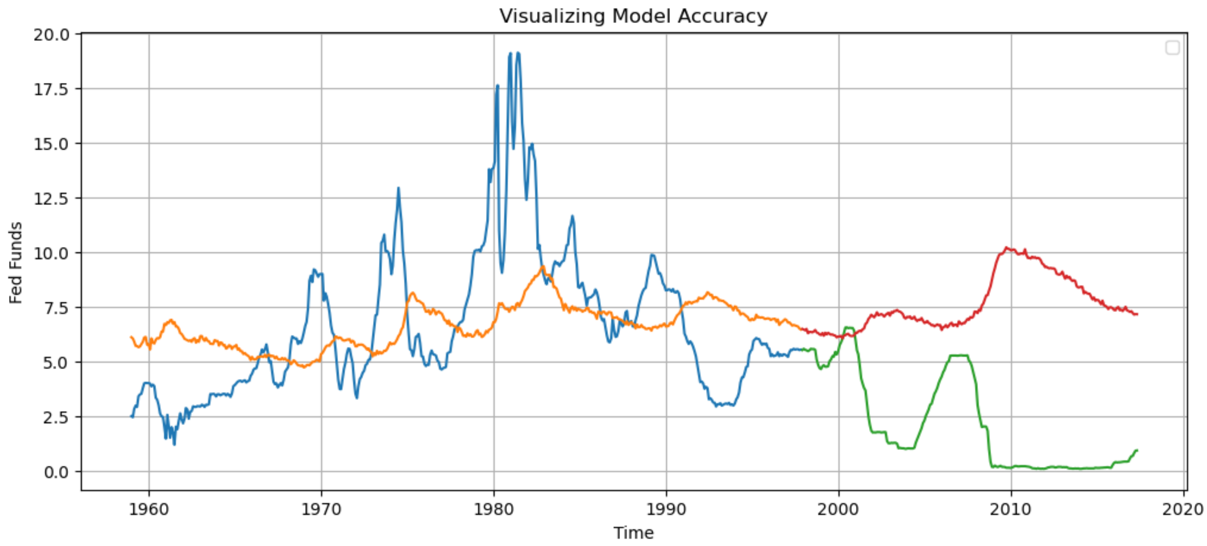
6.) Using a for loop. Repeat 3,4,5 for polynomial degrees 1,2,3

In [18]:
```
from sklearn.preprocessing import PolynomialFeatures
```

In [19]:
```
max_degrees = 3
```

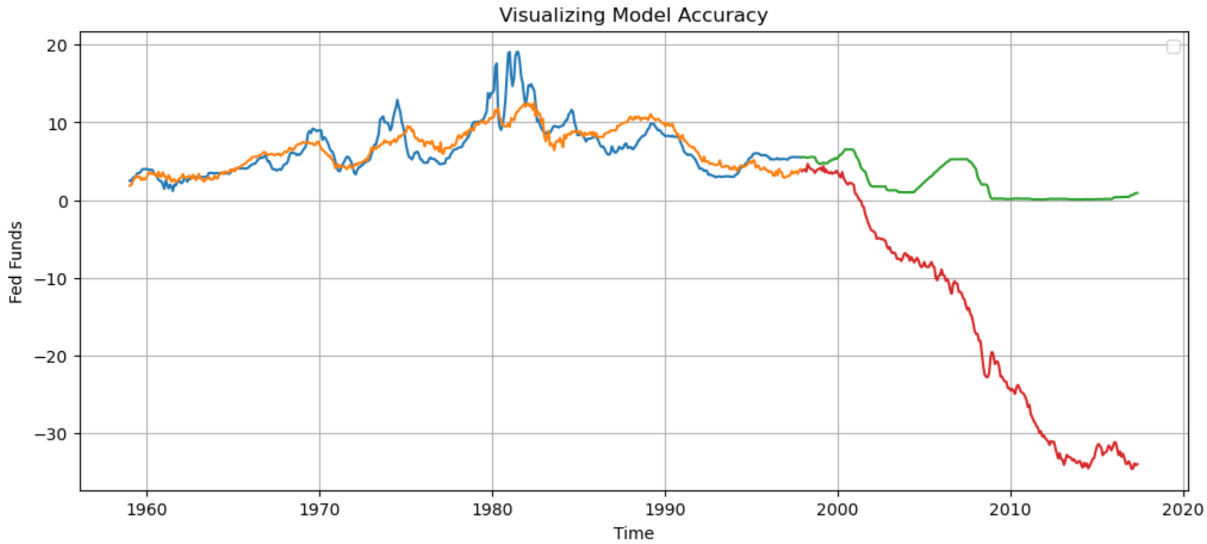
In [21]:
```
for degrees in range(1, max_degrees+1):
    print('DEGREE:', degrees)
    poly = PolynomialFeatures(degree = degrees)
    X_in_poly = poly.fit_transform(X_in)
    X_out_poly = poly.transform(X_out)   # without 'fit_'

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

    plt.figure(figsize = (12,5))

    ###

    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)

    print(in_preds.shape)
    print(y_in.shape)

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

    ###

    plt.ylabel("Fed Funds")
    plt.xlabel("Time")
    plt.title("Visualizing Model Accuracy")
    plt.legend([])
    plt.grid()
    plt.show()

    in_mse_1 = mean_squared_error(model1.predict(X_in_poly), y_in)
    out_mse_1 = mean_squared_error(model1.predict(X_out_poly),y_out)

    print("In MSE:", in_mse_1)
    print("Out MSE:",out_mse_1)
    print('____________________________________________________________________________________
```

```
DEGREE: 1
(467, 1)
(467,)
```



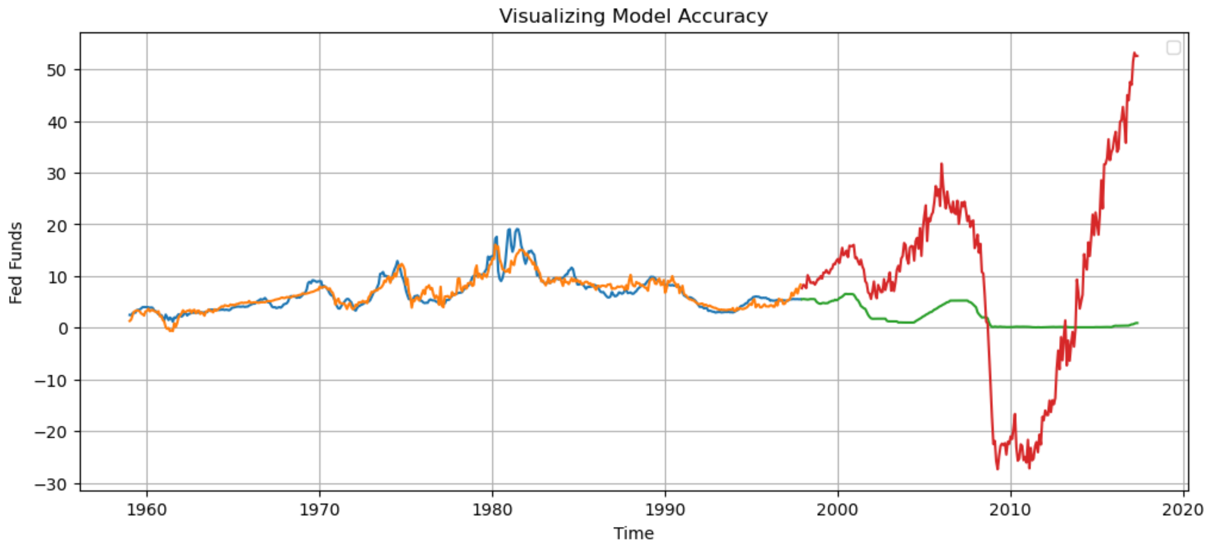

```
In MSE: 10.071422013168641
Out MSE: 40.36082783566679
```

```
DEGREE: 2
(467, 1)
(467,)
```




```
In MSE: 3.8634771392760676
Out MSE: 481.44650990363203
```

```
DEGREE: 3
(467, 1)
(467,)
```



```
In MSE: 1.8723636271946138
Out MSE: 371.76618900618945
```

7.) State your observations :

Looking at our graphs and results, we could see that while the power becomes bigger (the model becomes more complex), out in-sample MSE decreases. However, these results come up with larger out-sample MSE especially with teh increase of power. We could possibly infer that the flexibility of model would bring about the problem of overfitting.