Project Stage - IV (Basic Machine Learning) ddl: 04/28/2023

Goals

The goal of Stage IV is to utlize machine learning and statistical models to predict the trend of COVID-19 cases / deaths.

Tasks for Stage IV:

Task 1: (70 pts)

- Team: (30)
  - Develop Linear and Non-Linear (polynomial) regression models for predicting cases and deaths in US.
    - Start your data from the first day of infections in US. X-Axis - number of days since the first case, Y-Axis - number of new cases and deaths.
    - Calculate and report Root Mean Square Error (RMSE) for your models (linear and non-linear). Discuss bias versus variance tradeoff.
    - Plot trend line along for the data along with the forecast of 1 week ahead.
    - Describe the trends as compared to other countries.
- Member: (40 pts)
  - Utilize Linear and Non-Linear (polynomial) regression models to compare trends for a single state (each member should choose different state) and its counties (top 5 with highest number of cases). Start your data from the first day of infections.
    - X-Axis - number of days since the first case, Y - Axis number of new cases and deaths. Calcluate error using RMSE.
    - Identify which counties are most at risk. Model for top 5 counties with cases within a state and describe their trends.
    - Utilize the hospital data to calculate the point of no return for a state. Use percentage occupancy / utilization to see which states are close and what their trend looks like.
    - Perform hypothesis tests on questions identified in Stage II
      - e.x. *Does higher employment data (overall employment numbers) lead to higher covid case numbers or more rapid increase in covid cases.*. Here you would compare the covid cases to the state or county level enrichment data to prove or disprove your null hypothesis. In this case there will be a two tail - two sample t-test to see if there is a difference and then one-tail - two sample t-test to show higher or lower.
    - Depending on your type of data you can also perform Chi-square test for categorical hypothesis testing.

Task 2: (30 pts)

- Member:
  - For each of the aforemention analysis plot graphs,
    - trend line
    - confidence intervals (error in prediction)
    - prediction path (forecast)

Deliverable

- Each member creates separate notebooks for member tasks. Upload all notebooks and reports to Canvas. Do not submit to Github, at least before the submission deadline, to avoid potential plagiarism.

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from math import sqrt
from sklearn.svm import SVR
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
## Loading the comprehensive dataset that includes new cases, deaths, weekly cases(
confirmed_cases = pd.read_csv('data/covid_comprehensive.csv')
confirmed_cases.head(5)
```

Out[ ]:

	countyFIPS	County Name	State	StateFIPS	2022-06-01_confirmed	2022-06-02_confirmed	2022-06-03_confirmed	2022-06-04_confirme
0	1001	Autauga County	AL	1	6	9	0	
1	1003	Baldwin County	AL	1	68	68	0	
2	1005	Barbour County	AL	1	3	4	0	
3	1007	Bibb County	AL	1	8	4	0	
4	1009	Blount County	AL	1	4	7	0	

5 rows × 682 columns

In [ ]:
```
group_cases_USA = confirmed_cases.sum()
group_cases_USA = group_cases_USA.iloc[218:342]
group_cases_USA = group_cases_USA[group_cases_USA.index.str.contains('Mean')]

group_deaths_USA = confirmed_cases.sum()
group_deaths_USA = group_deaths_USA.iloc[557:-1]
group_deaths_USA = group_deaths_USA[group_deaths_USA.index.str.contains('Mean')]

# Masking Negative values
group_deaths_USA = group_deaths_USA.mask(group_deaths_USA < 0, group_deaths_USA.mea
```

This function calculates the RMSE of logistic, linear and polynomial regression.
It then plots the trend line for logistic and polynomial regression model.
The second plot is the prediction for the next weeks cases.
It also prints the prediction made by both non linear regression model for the next weeks cases.

In [ ]:
```
def quest1(data):
    temp = data.index


    X =[]
    for i in range(len(temp)):
        X.append(i)
    Y= data.values
    X = np.array(X)
    X= X.reshape(-1,1)
    Y = Y.astype('int')
    Y
```

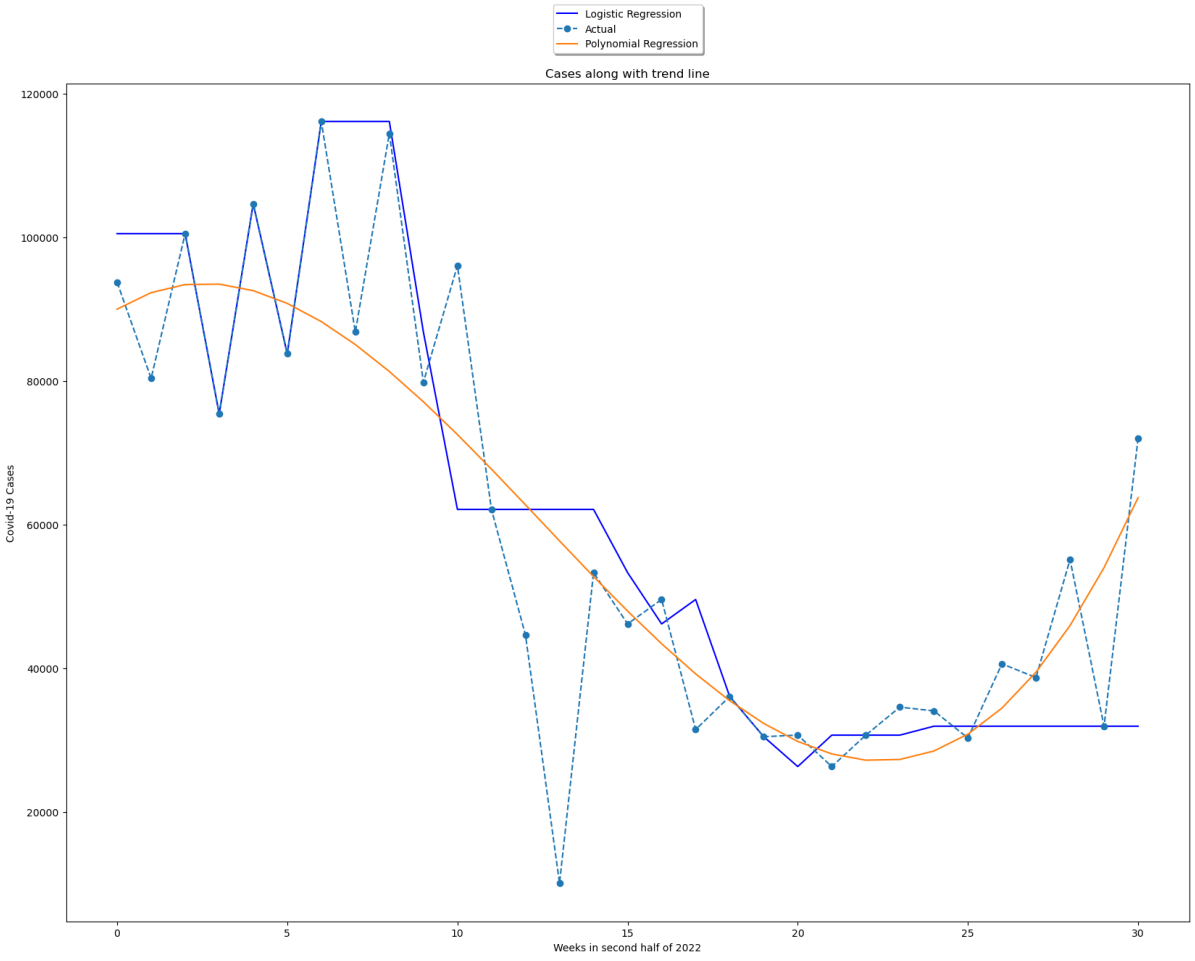
```
X_train_cases, X_test_cases, y_train_cases, y_test_cases = train_test_split(X,
pr = LinearRegression()
lrg = LogisticRegression()
lrl = LinearRegression()
poly = PolynomialFeatures(degree=3)
X_poly_train = poly.fit_transform(X_train_cases)
X_poly_test = poly.transform(X_test_cases)

pr.fit(X_poly_train, y_train_cases)
lrg.fit(X_train_cases,y_train_cases)
lrl.fit(X_train_cases,y_train_cases)

y_pred_pr = pr.predict(X_poly_test)
y_pred_lrl = lrl.predict(X_test_cases)
y_pred_lrg = lrg.predict(X_test_cases)

rmse_lrl = sqrt( mean_squared_error(y_test_cases, y_pred_lrl))
rmse_lrg = sqrt(mean_squared_error(y_test_cases, y_pred_lrg))
rmse_pr = sqrt(mean_squared_error(y_test_cases, y_pred_pr))


print(f'Logistic regression Root Mean Square Error (RMSE): {round(rmse_lrg,2)}'
print(f'Linear regression Root Mean Square Error (RMSE): {round(rmse_lrl,2)}')
print(f'Polynomial regression RMSE: {round(rmse_pr,2)}')
print('\n')

diff_pred_act = y_test_cases - y_pred_lrg
sum_diff = sum(diff_pred_act)
bias = sum_diff/len(y_test_cases)
print(f'Bias logistic regression model: {round(bias,2)}')

pred_var = y_pred_lrg.var()
print(f'Variance logistic regression model: {round(pred_var)}')

print('\n')

diff_pred_lrl = y_test_cases - y_pred_lrl
sum_diff = sum(diff_pred_lrl)
bias = sum_diff/len(y_test_cases)
print(f'Bias Linear regression model: {round(bias,2)}')

pred_var = y_pred_lrl.var()
print(f'Variance Linear regression model: {round(pred_var)}')

print('\n')

diff_pred_pr = y_test_cases - y_pred_pr
sum_diff = sum(diff_pred_pr)
bias = sum_diff/len(y_test_cases)
print(f'Bias Polynomial regression model: {round(bias,2)}')

pred_var = y_pred_pr.var()
print(f'Variance Polynomial regression model: {round(pred_var)}')

print('\n')
```

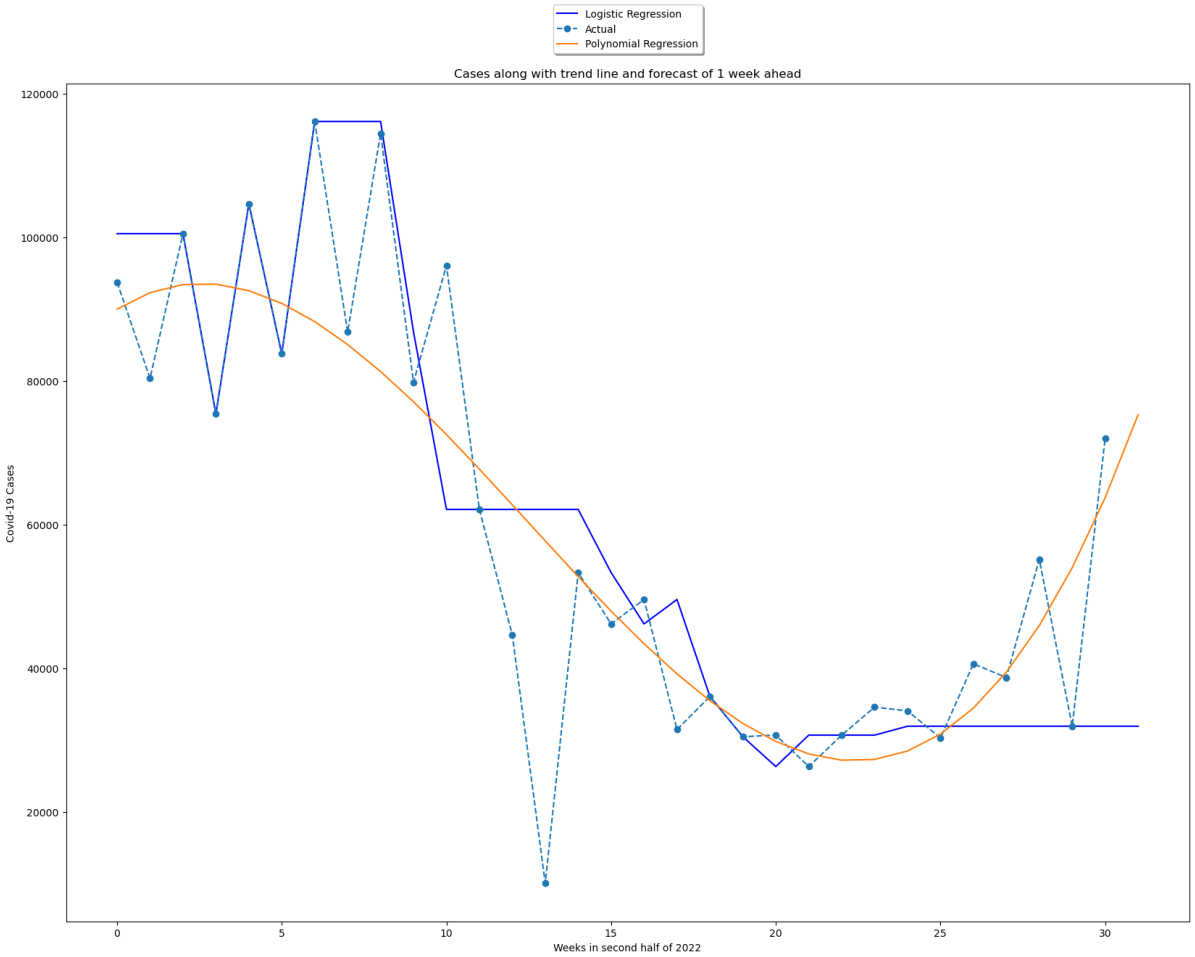
```
plt.figure(figsize=(20,15))
plt.title('Cases along with trend line')
plt.plot(X,
    lrg.predict(X),
    color='b',
    label = 'Logistic Regression')
plt.xlabel('Weeks in second half of 2022')
plt.ylabel('Covid-19 Cases')

plt.plot(X,Y,marker='o',linestyle='--',label = 'Actual')
plt.plot(X,pr.predict(poly.fit_transform(X)),label = 'Polynomial Regression')
plt.legend(
loc="upper center",
bbox_to_anchor=(0.5, 1.1),
ncol=1,
fancybox=True,
shadow=True,
)
plt.show()

next_week = []
for i in range(1):
    next_week.append(i + len(X))
next_week = np.array(next_week)
next_week= next_week.reshape(-1,1)

y_next_week = lrg.predict(next_week)
X_next = np.append(X, next_week)
X_next = X_next.reshape(-1,1)
y_pred_next_pr = pr.predict(poly.fit_transform(X_next))

plt.figure(figsize=(20,15))
plt.title('Cases along with trend line and forecast of 1 week ahead')
plt.plot(X_next,
        lrg.predict(X_next),
        color='b',
        label = 'Logistic Regression')
plt.xlabel('Weeks in second half of 2022')
plt.ylabel('Covid-19 Cases')

plt.plot(X,Y,marker='o',linestyle='--',label = 'Actual')
plt.plot(X_next,y_pred_next_pr,label = 'Polynomial Regression')
plt.legend(
    loc="upper center",
    bbox_to_anchor=(0.5, 1.1),
    ncol=1,
    fancybox=True,
    shadow=True,
)
plt.show()
print(f'The prediction for the next week of data using logistic regression is {
print(f'The prediction for the next week of data using Polynomial regression is
```

Weekly Covid-19 cases for the second half of 2022 using non-linear regression models

In [ ]:
```
quest1(group_cases_USA)
```

```
Logistic regression Root Mean Square Error (RMSE): 31402.83
Linear regression Root Mean Square Error (RMSE): 36031.97
Polynomial regression RMSE: 31492.7


Bias logistic regression model: -2771.0
Variance logistic regression model: 922617083


Bias Linear regression model: 5622.82
Variance Linear regression model: 245866846


Bias Polynomial regression model: 3793.82
Variance Polynomial regression model: 315466802
```



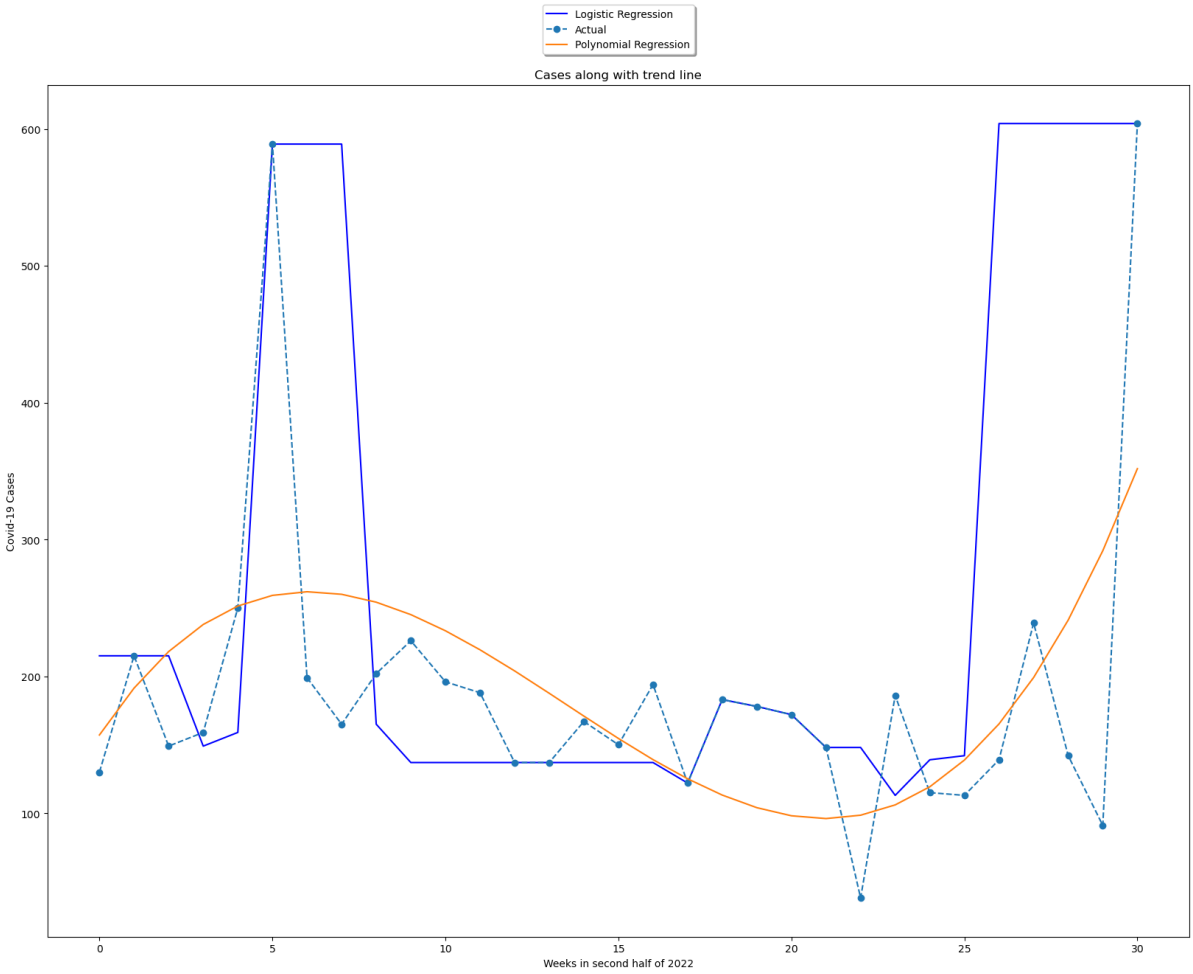


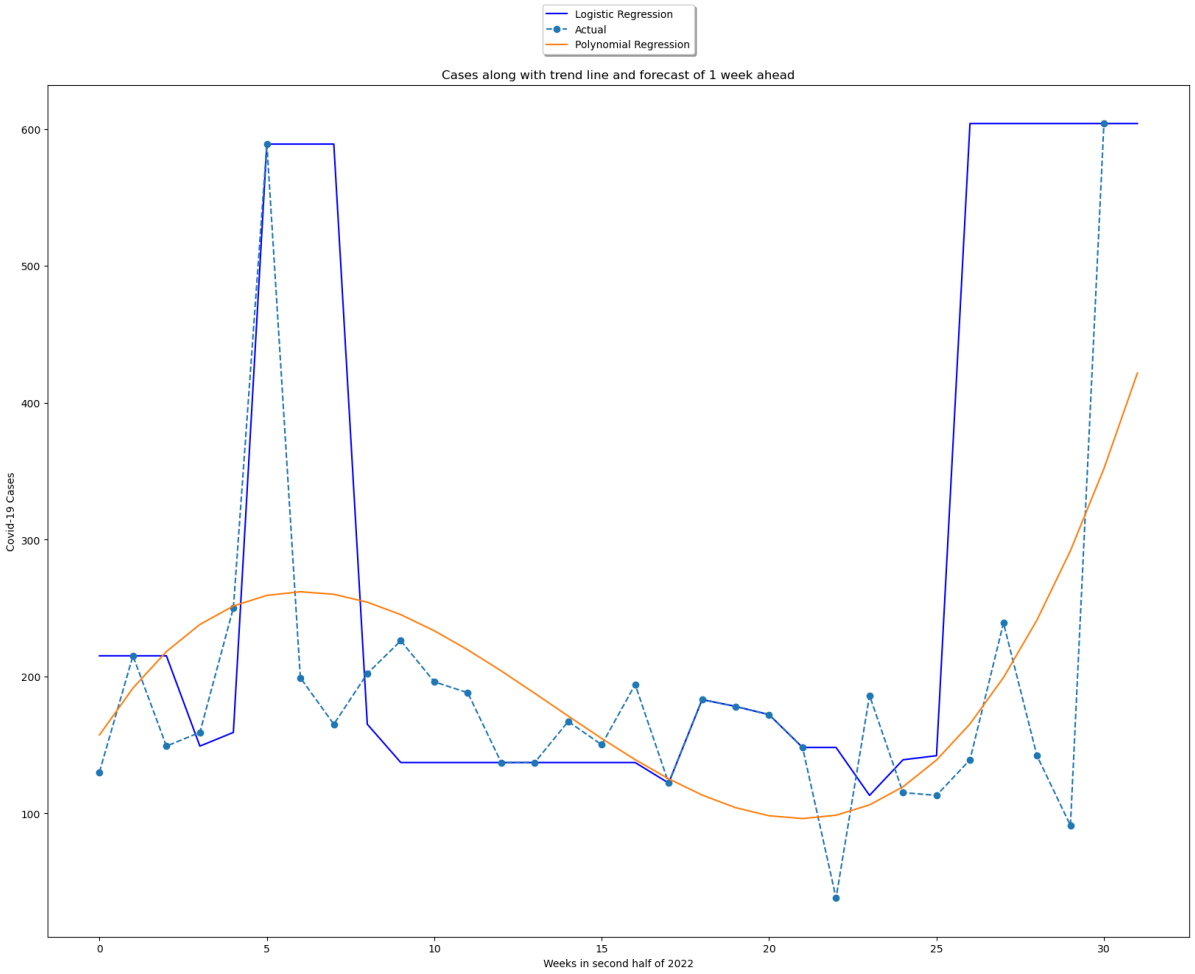

```
The prediction for the next week of data using logistic regression is 31903
The prediction for the next week of data using Polynomial regression is 75290
```

Weekly Covid-19 deaths for the second half of 2022 using non-linear regression models

In [ ]:
```
quest1(group_deaths_USA)
```

```
Logistic regression Root Mean Square Error (RMSE): 272.89
Linear regression Root Mean Square Error (RMSE): 55.52
Polynomial regression RMSE: 56.28


Bias logistic regression model: -191.75
Variance logistic regression model: 48093


Bias Linear regression model: -4.9
Variance Linear regression model: 370


Bias Polynomial regression model: 7.47
Variance Polynomial regression model: 3262
```


Logistic Regression
Actual
Polynomial Regression
Cases along with trend line
Covid-19 Cases
Weeks in second half of 2022
600
500
400
300
200
100
0
5
10
15
20
25
30




```
The prediction for the next week of data using logistic regression is 604
The prediction for the next week of data using Polynomial regression is 421
```

RMSE of the non-linear models(logistic regression and polynomial regression model) is less than the RMSE of the linear model for the weekly Covid-19 cases. However, we can see that the RMSE of linear model is less for the weekly Covid-19 deaths.
We decided to go it non-linear models because overall it was performing better if we are looking at the bias and variance as well as the RMSE.
We also noticed that the RMSE of non-linear models tend to decrease more rapidly as the size of the training set increases in comparison to the linear model.
Also, plotting trend line using non-linear models was visually more precise than linear model.

Lasly, we can see that the calculated bias is very low whereas the variance is very high. This means that our models were not complex enough and not capturing the important features in the data more effectively. This means our models are underfitting and should be trained using more data.

Comparing the U.S. to other Countries

In [ ]:
```
world_df = pd.read_csv("data/owid-covid-data-cases.csv")
world_df
```

Out[ ]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tota
0	AFG	Asia	Afghanistan	2020-02-24	5.0	5.0	NaN	
1	AFG	Asia	Afghanistan	2020-02-25	5.0	0.0	NaN	
2	AFG	Asia	Afghanistan	2020-02-26	5.0	0.0	NaN	
3	AFG	Asia	Afghanistan	2020-02-27	5.0	0.0	NaN	
4	AFG	Asia	Afghanistan	2020-02-28	5.0	0.0	NaN	
...	...	...	...	...	...	...	...	
260562	ZWE	Africa	Zimbabwe	2023-02-23	263921.0	NaN	NaN	
260563	ZWE	Africa	Zimbabwe	2023-02-24	263921.0	NaN	NaN	
260564	ZWE	Africa	Zimbabwe	2023-02-25	263921.0	NaN	NaN	
260565	ZWE	Africa	Zimbabwe	2023-02-26	263921.0	NaN	NaN	
260566	ZWE	Africa	Zimbabwe	2023-02-27	263921.0	NaN	NaN	

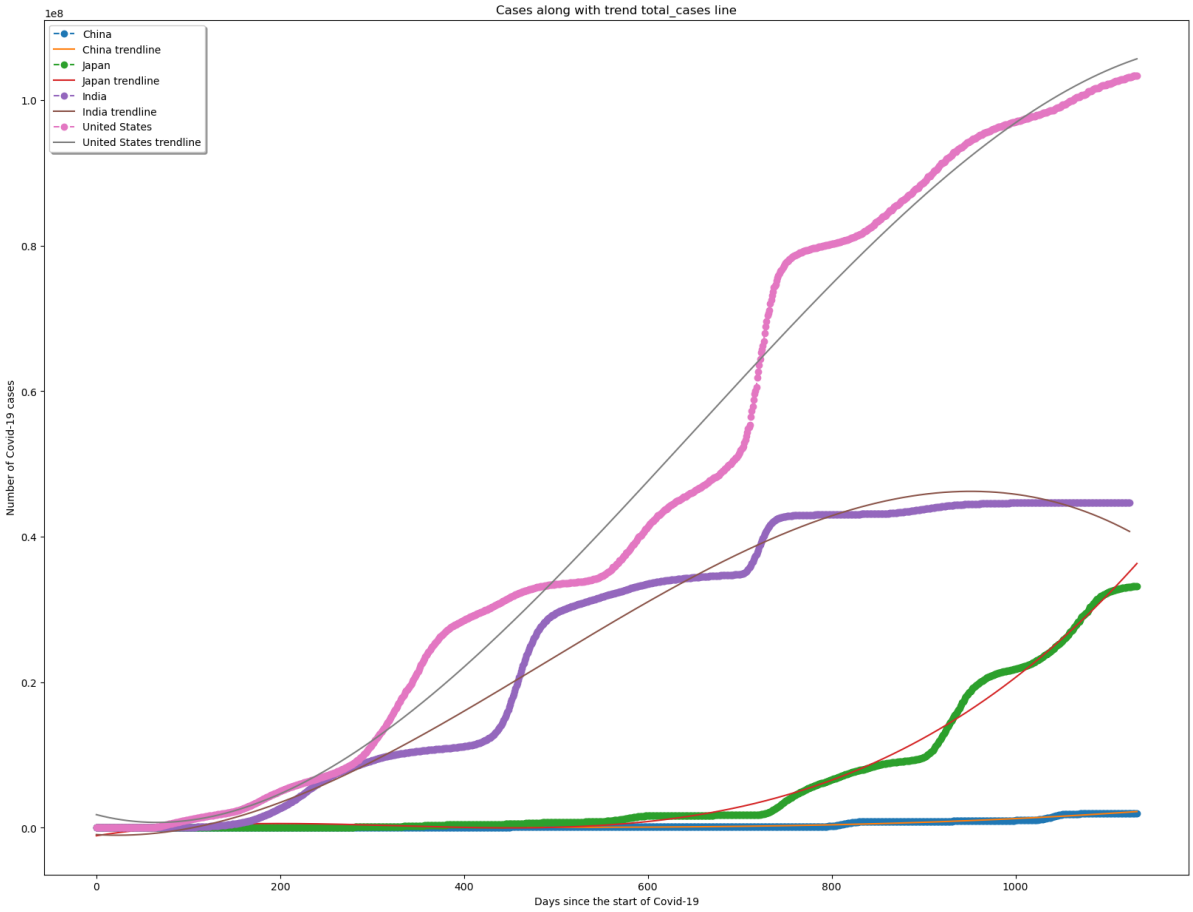
260567 rows × 67 columns

In [ ]:

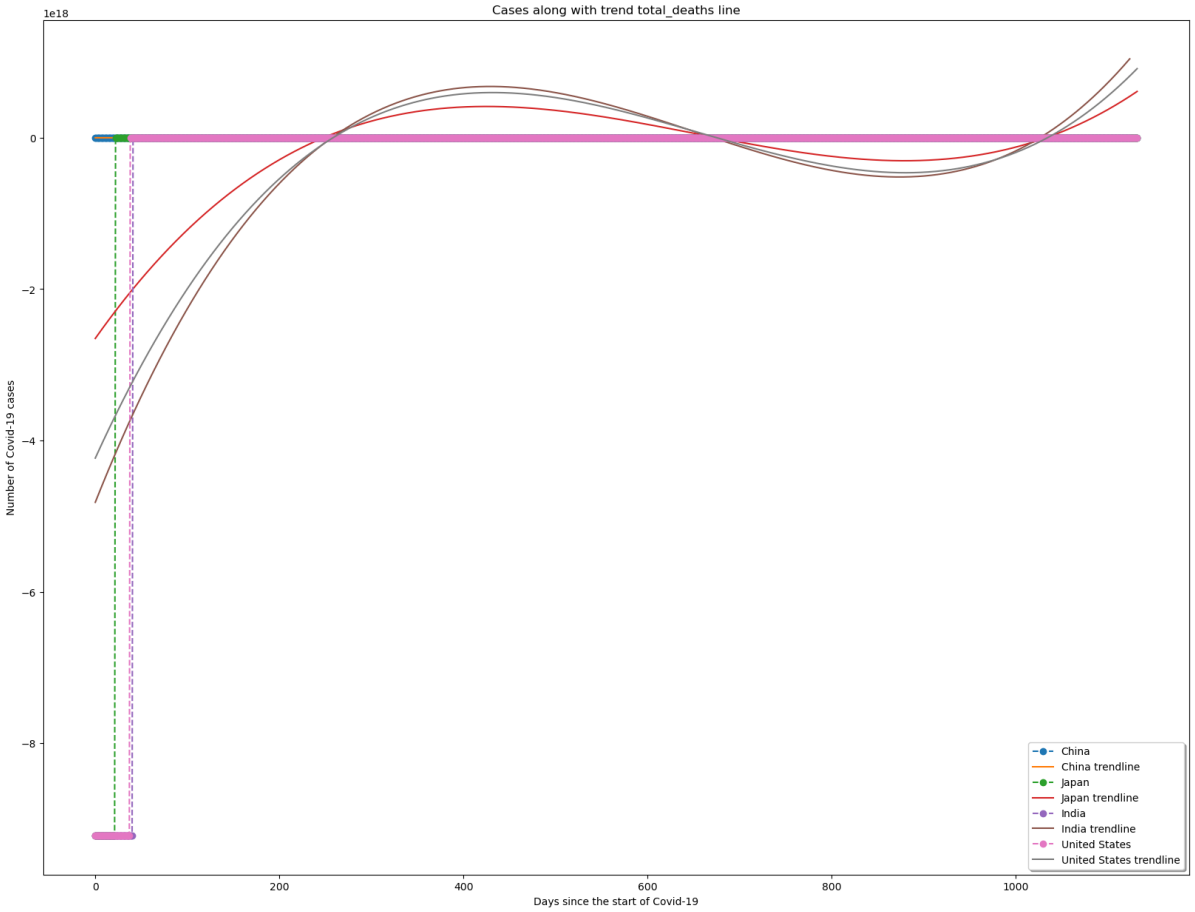
```
def question1_world(dataframe, columns, countries):
    plt.figure(figsize=(20,15))
    for country in countries:
        country_df = dataframe[dataframe['location'] == country][columns]
        temp = country_df.index

        X =[]
        for i in range(len(temp)):
            X.append(i)
        Y = country_df.values
        X = np.array(X)
        X= X.reshape(-1,1)
        Y = Y.astype('int')

        X_train_cases, X_test_cases, y_train_cases, y_test_cases = train_test_split

        lrl = LinearRegression()
        poly = PolynomialFeatures(degree=3)
        X_poly_fit = poly.fit_transform(X_train_cases)
        lrl.fit(X_poly_fit,y_train_cases)

        y_pred_lrl = lrl.predict(poly.transform(X_test_cases))
```

```
        rmse_lrl = sqrt( mean_squared_error(y_test_cases, y_pred_lrl))

        print(f'Polynomial regression Root Mean Square Error (RMSE) for {country}:

        # Plot the data for the current country
        plt.plot(X,Y,marker='o',linestyle='--',
            label=country)

        plt.plot(X, lrl.predict(poly.transform(X)), label=f'{country} trendline')

    plt.title(f'Cases along with trend {columns} line')
    plt.legend(fancybox=True, shadow=True)
    plt.xlabel('Days since the start of Covid-19')
    plt.ylabel('Number of Covid-19 cases')
    plt.show()

question1_world(world_df, 'total_cases', ['China','Japan','India', 'United States']
question1_world(world_df, 'total_deaths', ['China','Japan','India', 'United States'
```

```
Polynomial regression Root Mean Square Error (RMSE) for China: 119920.72
Polynomial regression Root Mean Square Error (RMSE) for Japan: 925635.78
Polynomial regression Root Mean Square Error (RMSE) for India: 2535835.83
Polynomial regression Root Mean Square Error (RMSE) for United States: 4726324.8
```



```
Polynomial regression Root Mean Square Error (RMSE) for China: 12509.39
Polynomial regression Root Mean Square Error (RMSE) for Japan: 1.0415629691676637e
+18
Polynomial regression Root Mean Square Error (RMSE) for India: 9.278251376311277e+
17
Polynomial regression Root Mean Square Error (RMSE) for United States: 1.528264760
1864164e+18
```



Analysis

Looking at the data plotted above, it goes to show that the United States on average had the highest amount of cases of the past few years. However, compared to China, they had lower amounts. Which is interesting to note, since China in particular has a higher/denser population then the United States. We believe that this is the result of erroneous reporting from them. The Total deaths, had India place irst, with the United States closely behind. Below is the total amount of deaths based each country.

In [ ]:
```python
def analysis_world(dataframe, countries):
    for country in countries:
        country_df = dataframe[dataframe['location'] == country]
        total_cases = country_df['total_cases'].iloc[-1]
        total_deaths = country_df['total_deaths'].iloc[-1]
        population = country_df['population'].iloc[-1]

        print(f"Country: {country}")
        print(f"Total Cases: {total_cases}")
```

```
        print(f"Total Deaths: {total_deaths}")
        print(f"Population: {population}")
        print("\n")
analysis_world(world_df, ['China','Japan','India', 'United States'])
```

```
Country: China
Total Cases: 2023904.0
Total Deaths: 87468.0
Population: 1425887360.0


Country: Japan
Total Cases: 33212438.0
Total Deaths: 72328.0
Population: 123951696.0


Country: India
Total Cases: 44687597.0
Total Deaths: 530771.0
Population: 1417173120.0


Country: United States
Total Cases: 103389954.0
Total Deaths: 1119550.0
Population: 338289856.0
```

In [ ]: