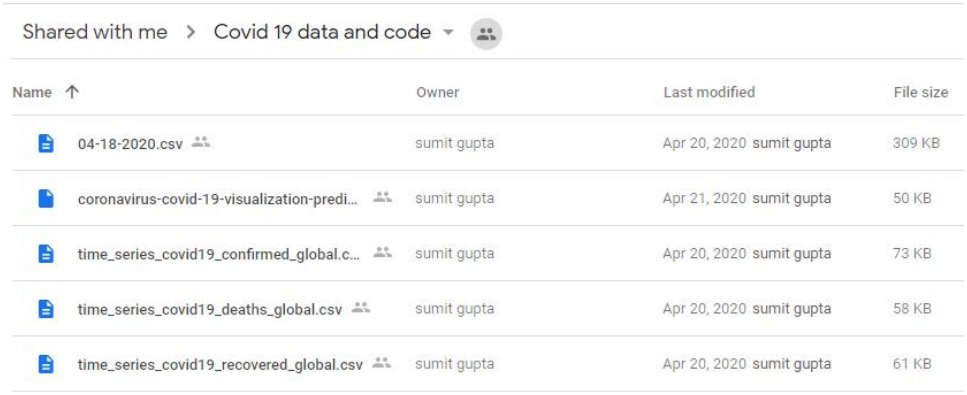
JichenDai_CS 524_Lab_#5

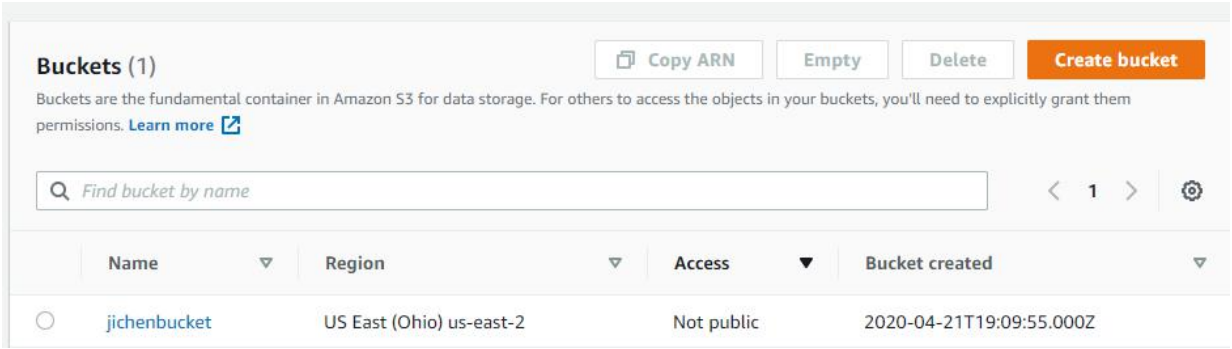
A. Download data.at:

https://drive.google.com/drive/folders/1thu0jOyjMAJ33gdtcC-25OHm-N78PYHc

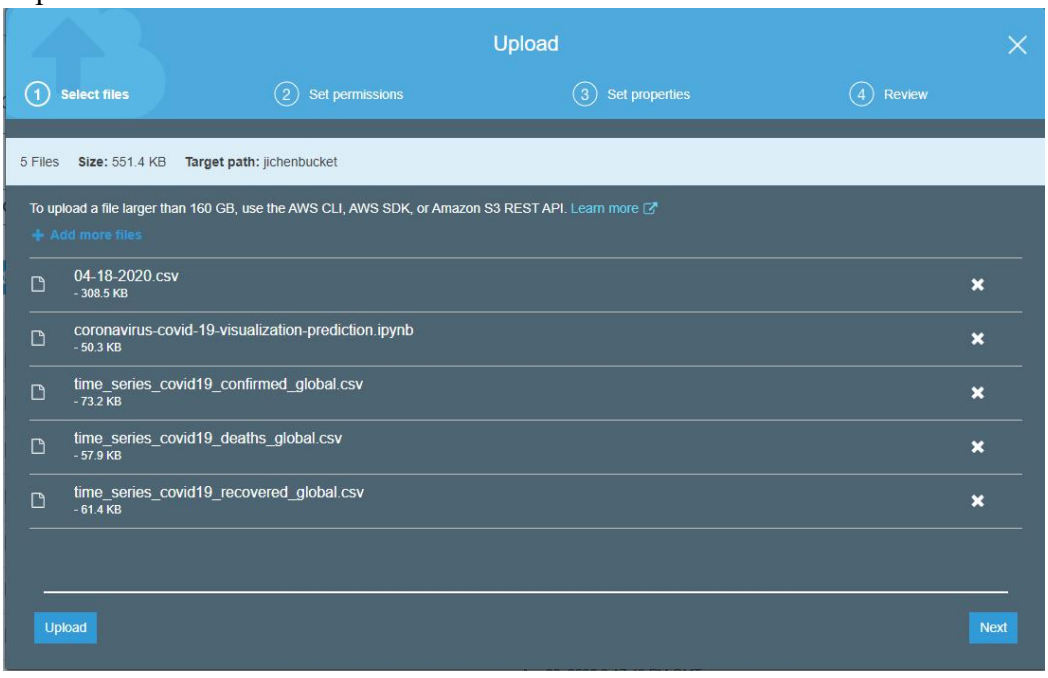


B. Create an Amazon S3 bucket and load the above data into this bucket.

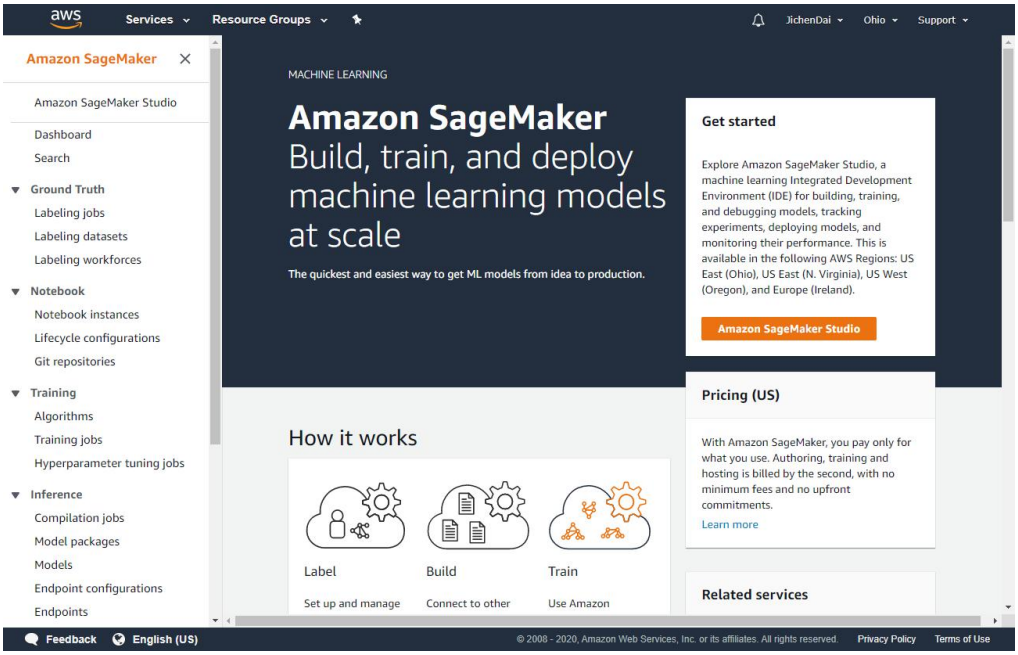
1. create 3s bucket as we did in Lab3



2. upload those five files.

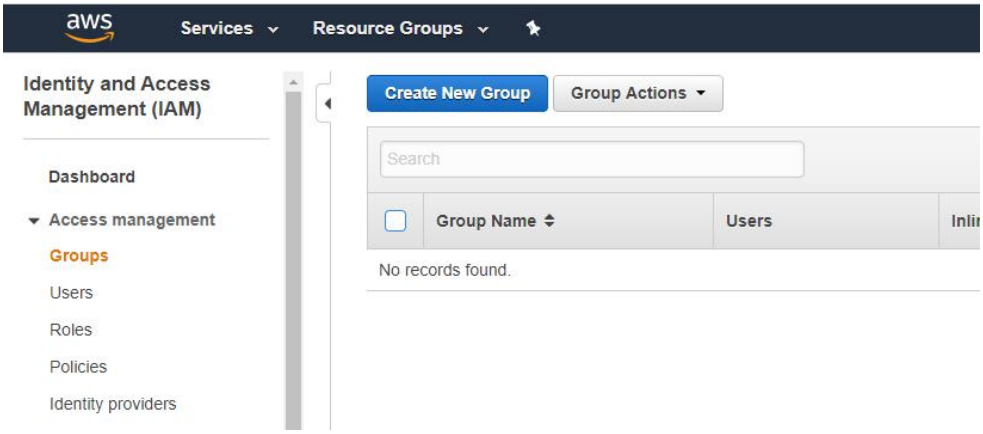


C. Open Amazon SageMaker

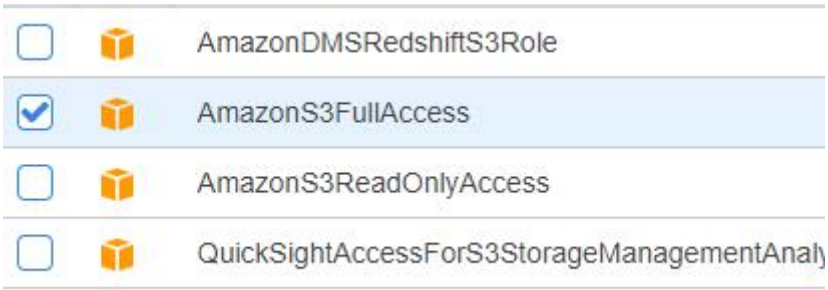


D. Launch a Notebook instance (and at the same time create the new IAM permissions for the Amazon S3 bucket you had created)

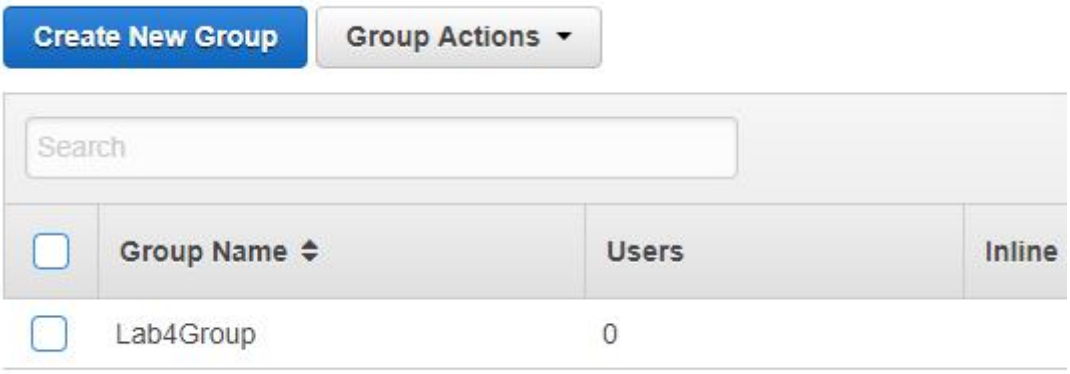
1. Create a IAM group for Amazon S3 bucket.



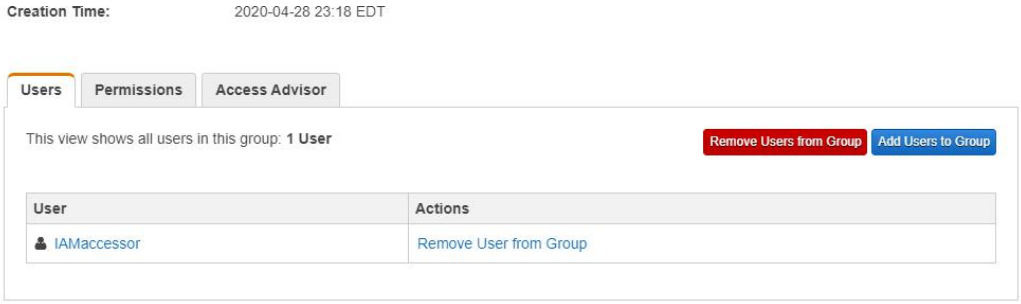
2. Attach policy choose “AmazonS3FullAccess”.



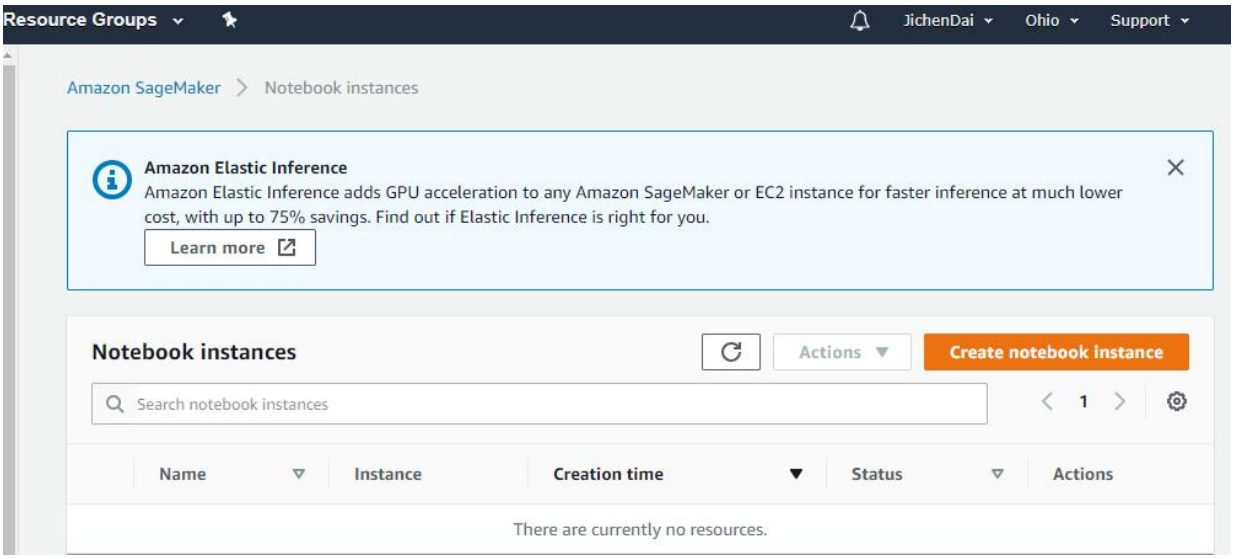
3. Click on “next” and “create group”.



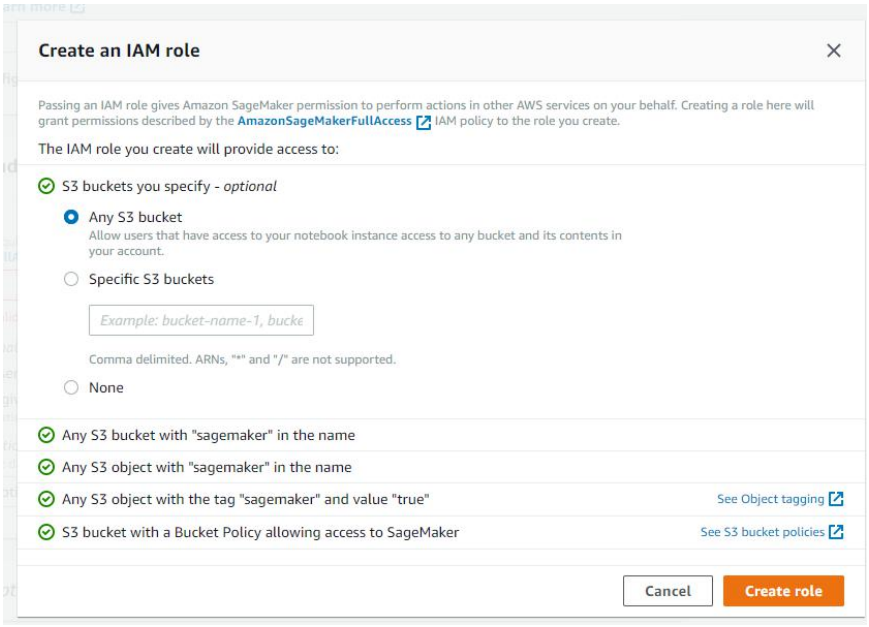
4. Go inside the group, click on “Add user to groups”. Add IAMaccessor.



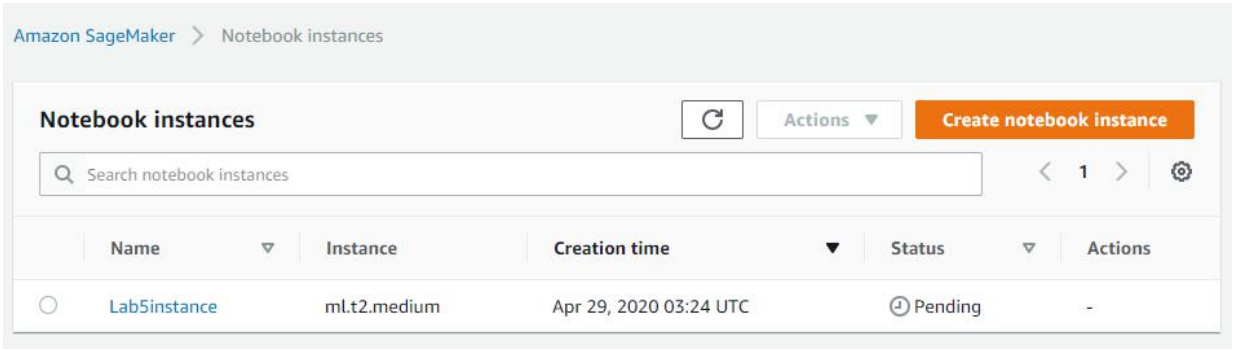
5. Go to sagemaker, click “Notebook instances” and click “create notebook instances”.



6. Crreate an IAM role

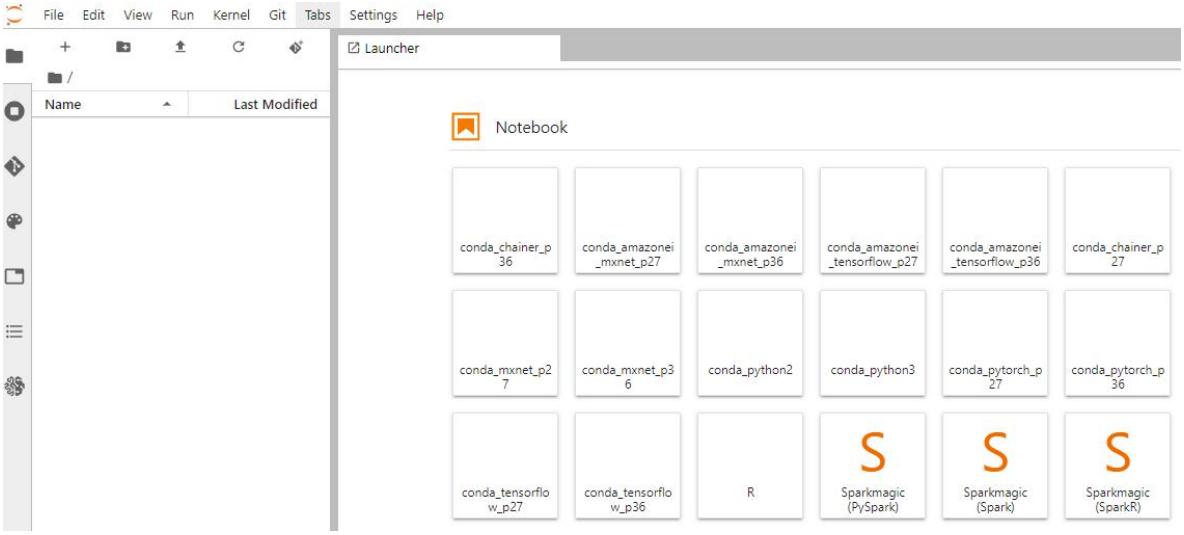


7. Then, click on create, we will got a new instance

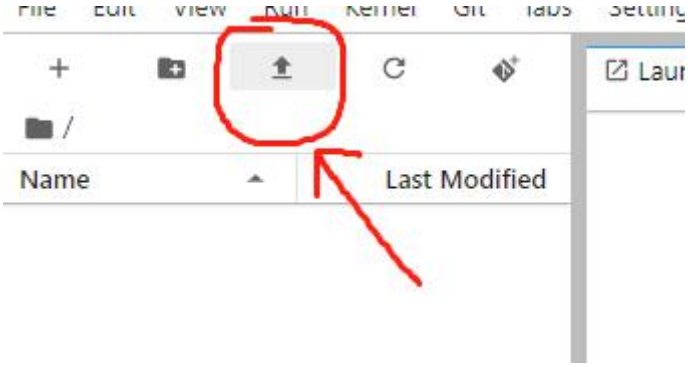


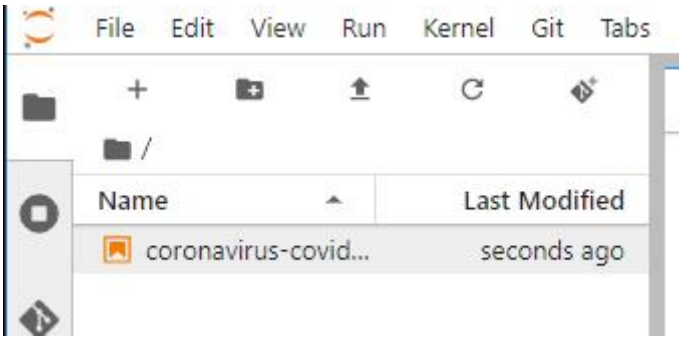
E. Load the notebook

1. Click on “open jupiter lab”

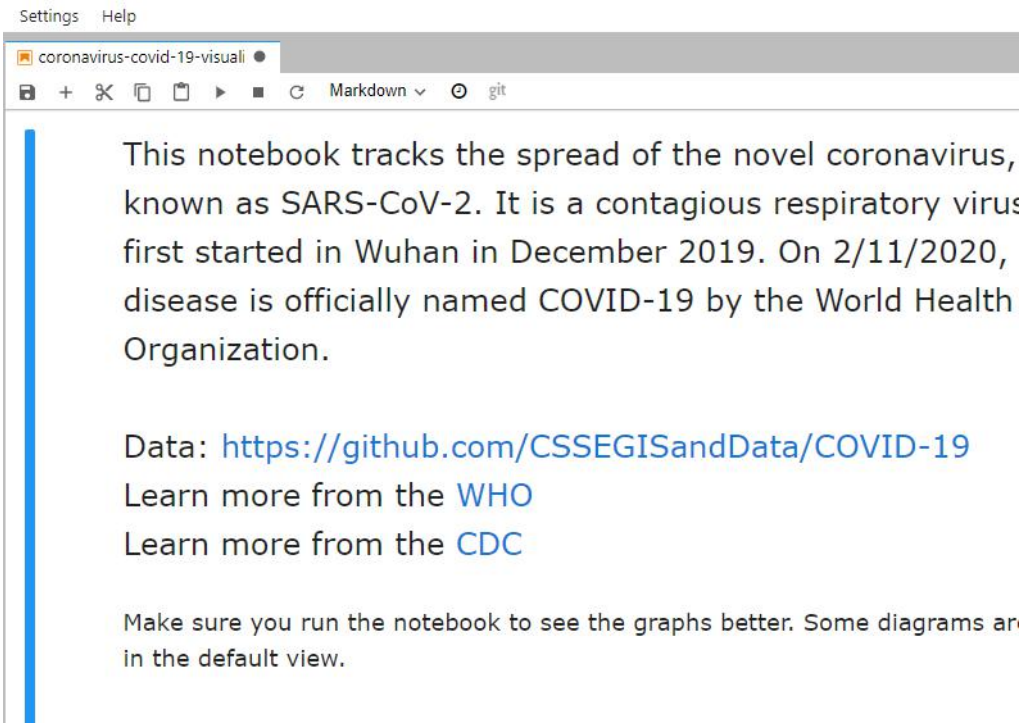


2. click on “upload” to upload the jupyter file.



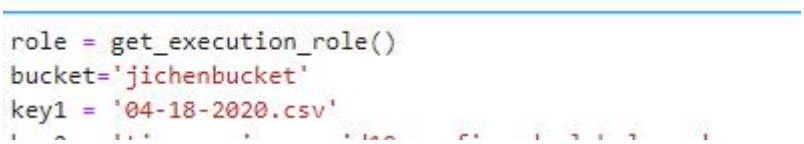


3. open the jupyter file and choose “conda_python3”



F. Edit the bucket name and run the code on each line of the notebook

1. Edit bucket name



```
role = get_execution_role()
bucket='jichenbucket'
key1 = '04-18-2020.csv'
```

2. Run the code piece by piece

```
data_location4 = 's3://{}/{}'.format(bucket, key4
```

Import the data

```
4]: latest_data=pd.read_csv(data_location1)
    confirmed_df=pd.read_csv(data_location2)
    deaths_df=pd.read_csv(data_location3)
    recoveries_df=pd.read_csv(data_location4)
```

```
]: latest_data.head()
```

```
]: confirmed_df.head()
```

```
]: cols = confirmed_df.keys()
```

Get all the dates for the outbreak

```
[7]: latest_data.head()
```

[7]:

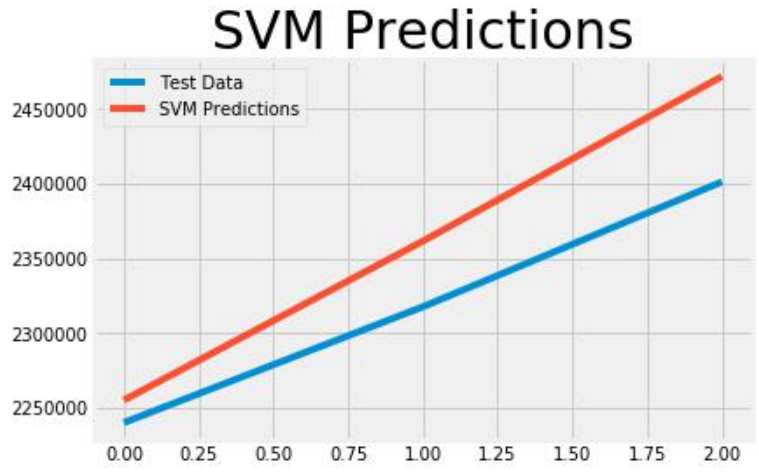
	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Coml
0	45001.0	Abbeville	South Carolina	US	2020-04-18 22:32:47	34.223334	-82.461707	15	0	0	15	Souti
1	22001.0	Acadia	Louisiana	US	2020-04-18 22:32:47	30.295065	-92.414197	110	7	0	103	Lou
2	51001.0	Accomack	Virginia	US	2020-04-18 22:32:47	37.767072	-75.632346	33	0	0	33	V
3	16001.0	Ada	Idaho	US	2020-04-18 22:32:47	43.452658	-116.241552	593	9	0	584	Ada,
4	19001.0	Adair	Iowa	US	2020-04-18 22:32:47	41.330756	-94.471059	1	0	0	1	Adai

```
[8]: confirmed_df.head()
```

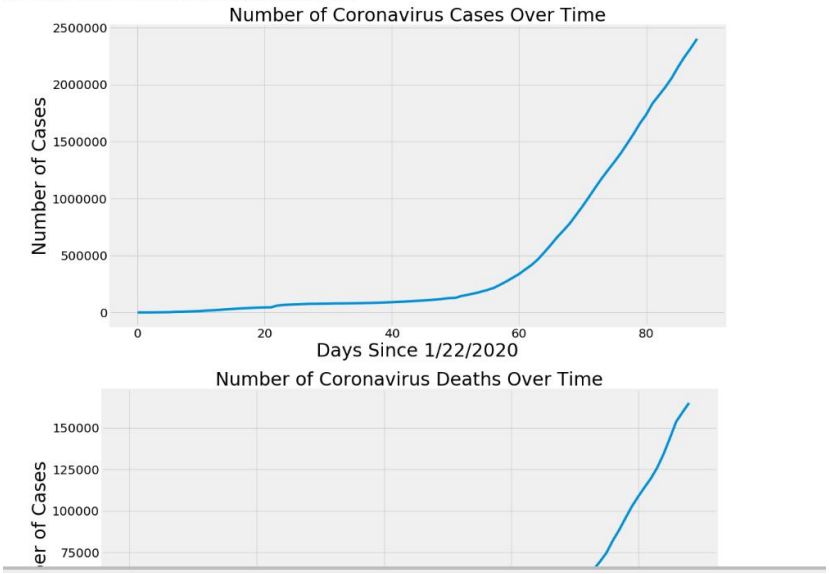
[8]:

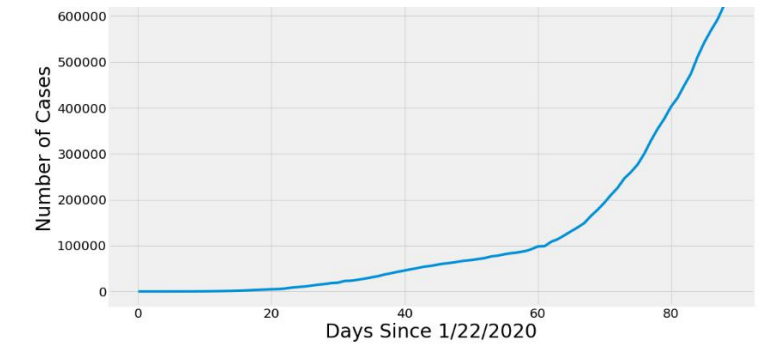
	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	4/10/20	4/11/20	4/1
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	521	555	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	416	433	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	1761	1825	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	601	601	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	19	19	

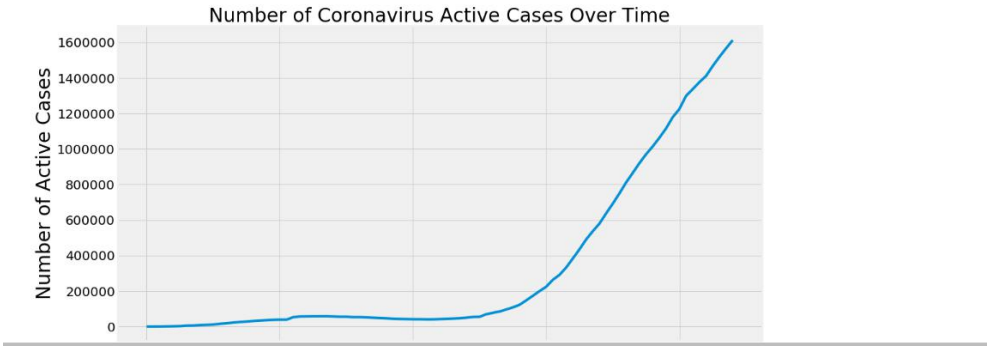
```
MSE: 2375651382.185125
[21]: s3.Object(bucket_name='jichenbucket', key='Output/SVM')
```

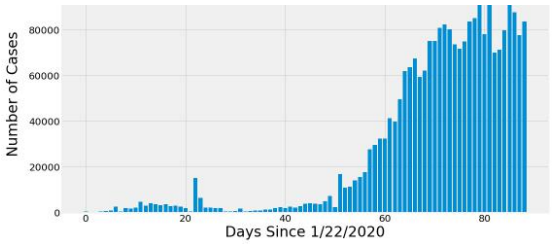


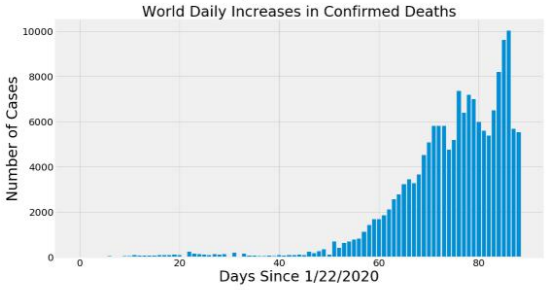
[29]: s3.Object(bucket_name='jichenbucket', key='Output/Active cases over time')

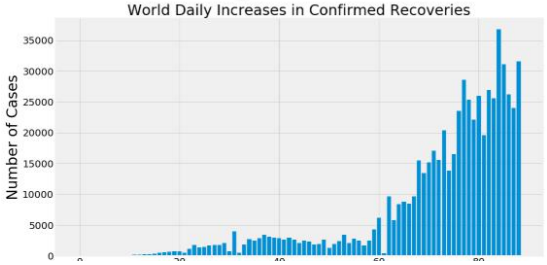


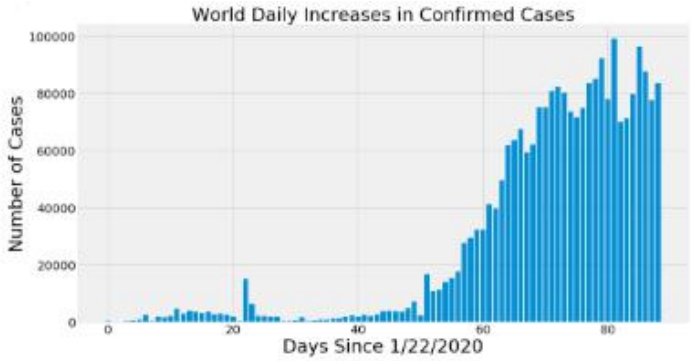


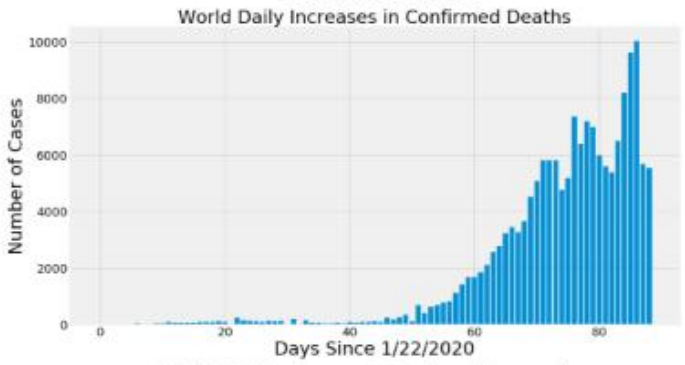


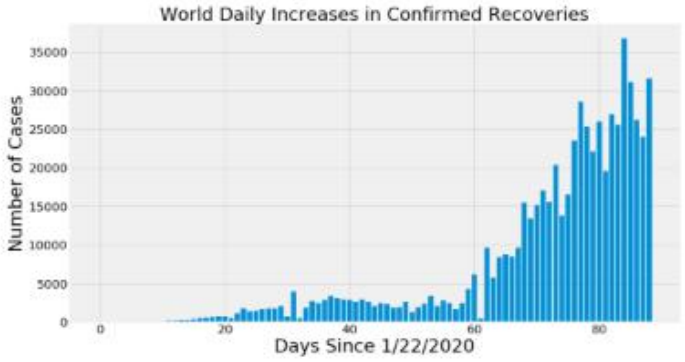

Number of Cases
Days Since 1/22/2020

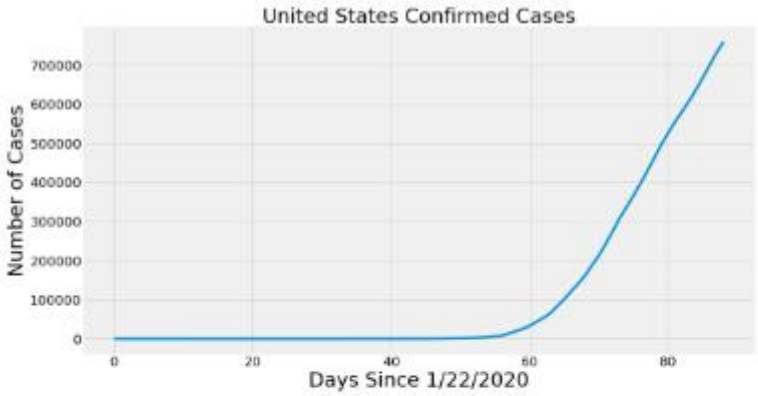

World Daily Increases in Confirmed Deaths
Number of Cases
Days Since 1/22/2020

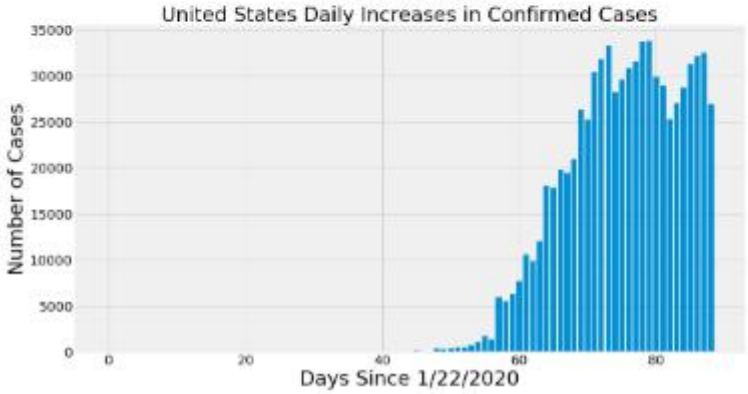

World Daily Increases in Confirmed Recoveries
Number of Cases

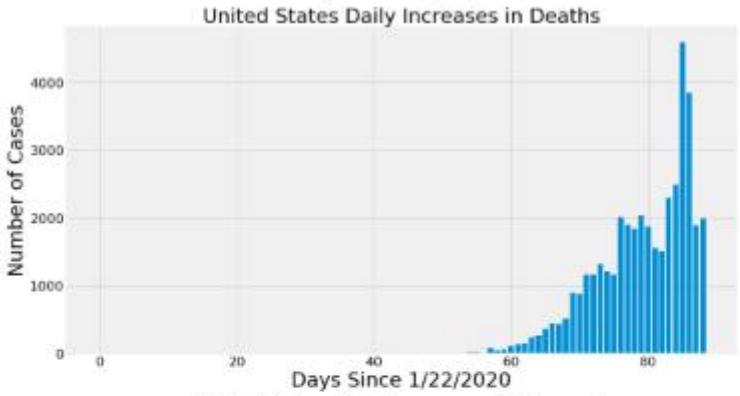

World Daily Increases in Confirmed Cases
Number of Cases
Days Since 1/22/2020

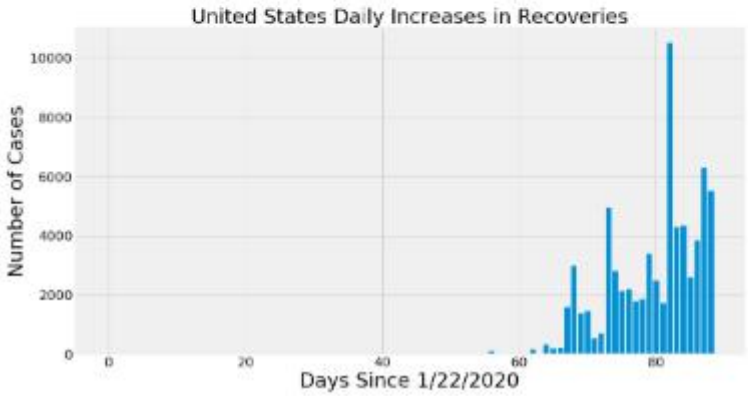

World Daily Increases in Confirmed Deaths
Number of Cases
Days Since 1/22/2020

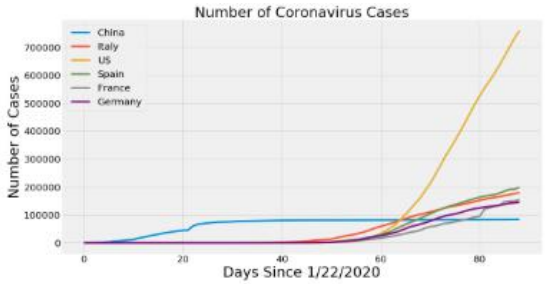

World Daily Increases in Confirmed Recoveries
Number of Cases
Days Since 1/22/2020

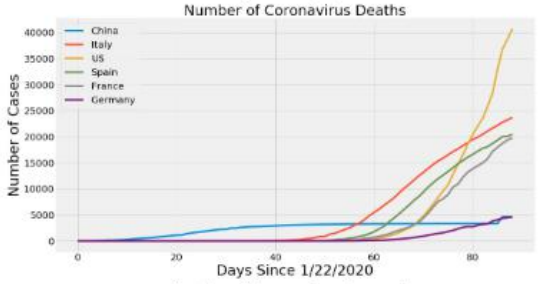

United States Confirmed Cases
Number of Cases
Days Since 1/22/2020

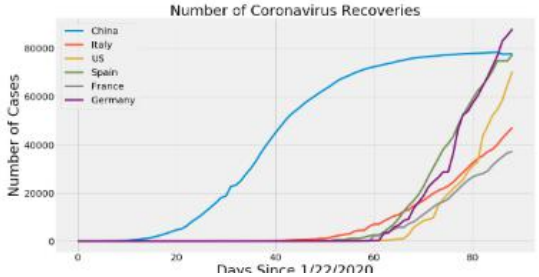

United States Daily Increases in Confirmed Cases
Number of Cases
Days Since 1/22/2020

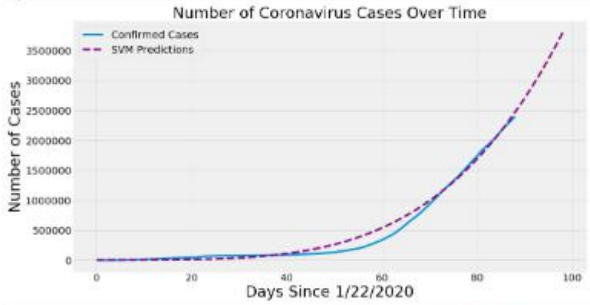

United States Daily Increases in Deaths
Number of Cases
Days Since 1/22/2020


United States Daily Increases in Recoveries
Number of Cases
Days Since 1/22/2020

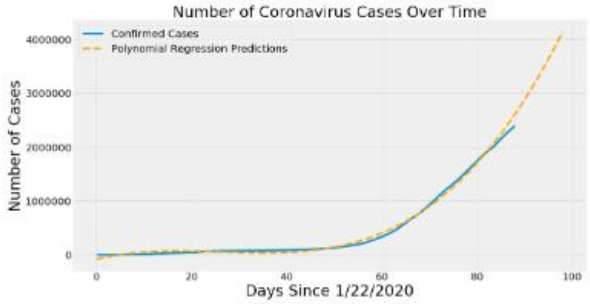


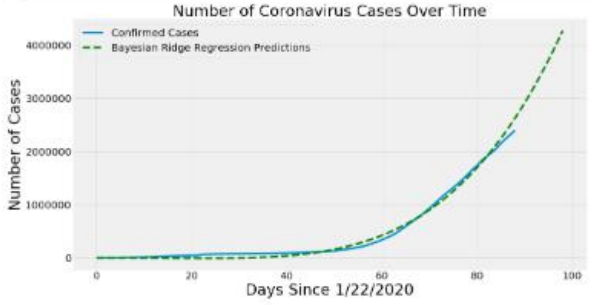







```
plot_predictions(adjusted_dates, world_cases, linear_pred, 'Polynomial Regression Predictions', 'orange')
```





```
# Future predictions using SVM
svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'SVM Predicted Number of Confirmed Cases Worldwide': np.round(svm_pred[-10:])})
svm_df
```

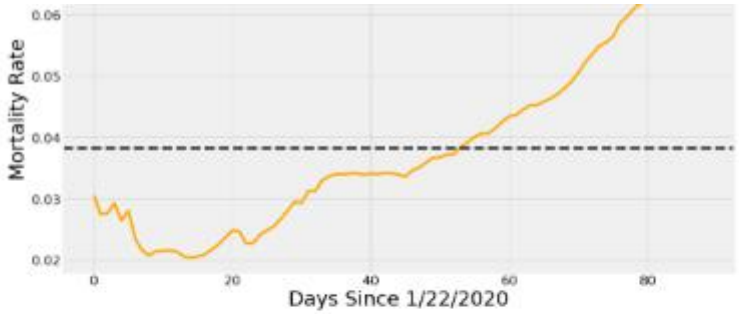
	Date	SVM Predicted Number of Confirmed Cases Worldwide
0	04/20/2020	2585685.0
1	04/21/2020	2703831.0
2	04/22/2020	2825754.0
3	04/23/2020	2951752.0
4	04/24/2020	3081887.0
5	04/25/2020	3216330.0
6	04/26/2020	3355332.0
7	04/27/2020	3498887.0
8	04/28/2020	3646992.0
9	04/29/2020	3798636.0

```
linear_pred = linear_pred.reshape(1,-1)[0]
svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'Polynomial Predicted Number of Confirmed Cases Worldwide': np.round(linear_pred[-10:])})
svm_df
```

	Date	Polynomial Predicted Number of Confirmed Cases Worldwide
0	04/20/2020	2746776.0
1	04/21/2020	2883303.0
2	04/22/2020	3024271.0
3	04/23/2020	3169752.0
4	04/24/2020	3319814.0
5	04/25/2020	3474527.0
6	04/26/2020	3633960.0
7	04/27/2020	3798185.0
8	04/28/2020	3967269.0
9	04/29/2020	4141283.0

```
# Future predictions using Bayesian Ridge
svm_df = pd.DataFrame({'Date': future_forcast_dates[-10:], 'Bayesian Ridge Predicted Number of Confirmed Cases Worldwide': np.round(bayesian_pred[-10:])})
svm_df
```

	Date	Bayesian Ridge Predicted Number of Confirmed Cases Worldwide
0	04/20/2020	2769856.0
1	04/21/2020	2913950.0
2	04/22/2020	3063326.0
3	04/23/2020	3218337.0
4	04/24/2020	3379081.0
5	04/25/2020	3545696.0
6	04/26/2020	3718321.0
7	04/27/2020	3897099.0
8	04/28/2020	4082171.0
9	04/29/2020	4273683.0



Recovery Rate (worldwide) *susceptible to change *

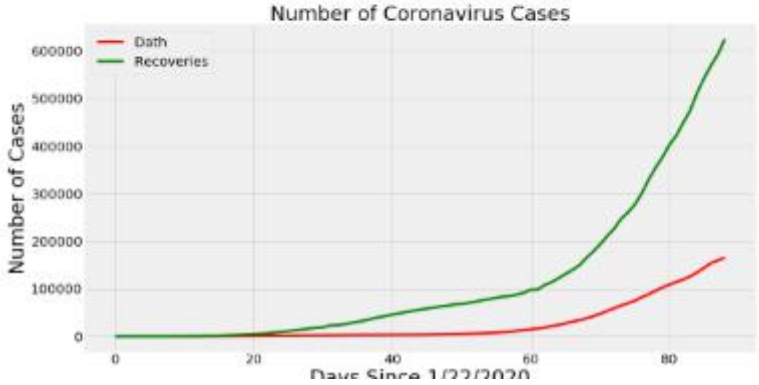
```
[43]: mean_recovery_rate = np.mean(recovery_rate)
plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, recovery_rate, color='blue')
plt.axhline(y = mean_recovery_rate,linestyle='--', color='black')
plt.title('Recovery Rate of Coronavirus Over Time', size=30)
plt.legend(['recovery rate', 'y='+str(mean_recovery_rate)], prop={'size': 20})
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Recovery Rate', size=30)
plt.xticks(size=30)
plt.yticks(size=30)
img_data = io.BytesIO()
plt.savefig(img_data, format='png')
img_data.seek(0)
s3 = boto3.resource('s3')
bucket1 = s3.Bucket(bucket)
```

```
[48]: s3.Object(bucket_name='jichenbucket', key='Output/Recovery_Rate')
```

Graphing deaths against recoveries

```
[49]: plt.plot(adjusted_dates, total_deaths, color='r')
plt.plot(adjusted_dates, total_recovered, color='green')
plt.legend(['Dath', 'Recoveries'], loc='best', fontsize=20)
plt.title('Number of Coronavirus Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
img_data = io.BytesIO()
plt.savefig(img_data, format='png')
img_data.seek(0)
s3 = boto3.resource('s3')
bucket1 = s3.Bucket(bucket)
bucket1.put_object(Body=img_data, ContentType='image/png', Key="Output/Death_vs_Recoveries")
```

```
[49]: s3.Object(bucket_name='jichenbucket', key='Output/Death_vs_Recoveries')
```



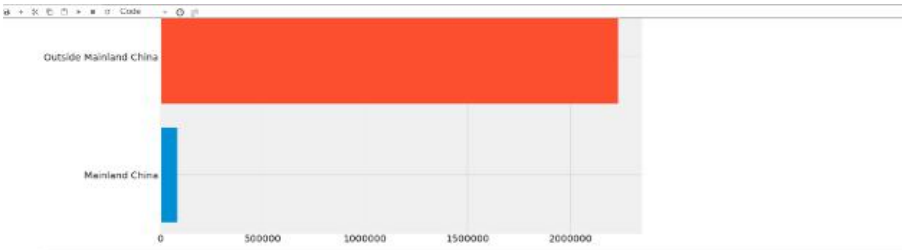
[94]:

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
0	US	732197	38664	64940	628693	0.052805...
1	Spain	191726	26043	74797	96886	0.10454
2	Italy	175925	23227	44927	107771	0.132030...
3	France	149749	19185	36587	93217	0.128103...
4	Germany	143342	4459	85400	53483	0.031104...
5	United Kingdom	115514	15498	414	99402	0.134180...
6	China	83787	4636	77614	1537	0.055330...
7	Turkey	82329	1890	10453	69986	0.022956...
8	Iran	80868	5031	55987	19850	0.062212...
9	Belgium	37183	5453	8348	23382	0.146651...
10	Russia	36793	313	3057	33423	0.008507...
11	Brazil	36658	2354	14026	20278	0.064215...
12	Canada	34356	1480	10964	21992	0.043074...
13	Netherlands	31766	3613	317	27836	0.113738...
14	Switzerland	27404	1368	17100	8936	0.049919...
15	Portugal	19185	887	610	18388	0.046209...
16	India	15722	521	2463	12738	0.033138...
17	Ireland	14758	571	77	14110	0.038709...
18	Austria	14671	443	10214	4014	0.030196...
19	Peru	14420	348	6684	7388	0.024133...
20	Sweden	13822	1511	550	11761	0.109378...
21	Israel	13265	164	3456	9645	0.012363...
22	Korea, South	10653	232	7937	2484	0.021777...
23	Japan	10296	222	1060	9005	0.021561...
24	Chile	9130	126	4035	5569	0.013804...
25	Ecuador	9022	456	1008	7558	0.050543...
26	Poland	8742	347	981	7414	0.039694...
27	Romania	8418	421	1730	6267	0.050011...
28	Saudi Arabia	8274	92	1329	6853	0.011119...
29	Pakistan	7638	143	1832	5663	0.018722...
30	Denmark	7437	346	4031	3060	0.046524...
31	Norway	7036	164	32	6840	0.023308...
32	Mexico	6875	546	2125	4204	0.079418...
33	Czechia	6606	181	1227	5198	0.027399...
34	Australia	6547	67	4124	2356	0.010233...
35	United Arab Emirates	6302	37	1188	5077	0.005871...
36	Indonesia	6248	535	631	5082	0.085627...
37	Philippines	6087	397	516	5174	0.065221...
38	Serbia	5934	117	657	5240	0.019719...
39	Singapore	5992	11	740	5241	0.001835...
40	Malaysia	5305	88	3102	2115	0.016588...
41	Ukraine	5106	133	275	4698	0.026047...
42	Qatar	5008	8	510	4490	0.001597...
43	Belarus	4779	45	342	4392	0.009416...
44	Dominican Republic	4335	217	312	3806	0.050057...
45	Panama	4210	116	122	3972	0.027556...
46	Finland	3681	90	1700	1891	0.024449...
47	Luxembourg	3537	72	601	2864	0.020356...
48	Colombia	3439	153	634	2652	0.044490...
49	South Africa	3054	52	983	2019	0.017029...
50	Egypt	3052	224	701	2127	0.073387...
51	Argentina	2758	129	685	1944	0.046773...
52	Thailand	2733	47	1787	899	0.017197...
53	Morocco	2685	137	314	2234	0.051024...
54	Algeria	2534	367	894	1273	0.144831...
55	Moldova	2378	57	391	1930	0.023969...
56	Greece	2235	110	269	1856	0.049217...
57	Bangladesh	2144	84	66	1994	0.039179...
58	Hungary	1834	172	231	1431	0.093784...
59	Croatia	1832	39	615	1178	0.021288...
60	Bahrain	1773	7	755	1011	0.003948...
61	Iceland	1760	9	1291	460	0.005113...

```
province_df = pd.DataFrame({'Province/State Name': unique_provinces, 'Country': province_country, 'Number of Confirmed Cases': pr
                            'Number of Deaths': province_death_cases, 'Number of Recoveries' : province_recovery_cases,
                            'Mortality Rate': province_mortality_rate})
# number of cases per country/region
```

[7]:

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
0	New York	US	241712	17671	0	0.073107
1	New Jersey	US	81420	4070	0	0.049988
2	Hubei	China	68128	4512	63494	0.066228
3	Massachusetts	US	36372	1404	0	0.038601
4	Pennsylvania	US	31652	1042	0	0.032920
5	Michigan	US	30791	2308	0	0.074957
6	California	US	30491	1140	0	0.037388
7	Illinois	US	29160	1259	0	0.043176
8	Florida	US	25492	748	0	0.029345
9	Louisiana	US	23580	1267	0	0.053732
10	Texas	US	18704	476	0	0.025449
11	Georgia	US	17669	673	0	0.038093
12	Connecticut	US	17550	1096	0	0.062449
13	Quebec	Canada	17521	668	0	0.038127
14	Maryland	US	12326	421	0	0.034155
15	Washington	US	11776	613	0	0.052055
16	Ontario	Canada	11013	564	0	0.051212
17	Indiana	US	10641	545	0	0.051217
18	Ohio	US	10222	451	0	0.044120
19	Colorado	US	9047	389	0	0.042997
20	Virginia	US	8053	258	0	0.032037
21	Tennessee	US	6589	142	0	0.021551
22	North Carolina	US	6528	187	0	0.028615
23	Missouri	US	5570	197	0	0.035311
24	Arizona	US	4724	180	0	0.038103
25	Alabama	US	4712	153	0	0.032470
26	Rhode Island	US	4491	137	0	0.030505
27	South Carolina	US	4248	119	0	0.028013
28	Wisconsin	US	4199	212	0	0.050488
29	Mississippi	US	3974	152	0	0.038248
30	Nevada	US	3626	151	0	0.041643
31	New South Wales	Australia	2926	26	1379	0.008885
32	Utah	US	2917	25	0	0.008570
33	Kentucky	US	2707	144	0	0.053195
34	District of Columbia	US	2666	91	0	0.034133
35	Alberta	Canada	2562	51	0	0.019906
36	Delaware	US	2538	67	0	0.026398
37	Iowa	US	2513	74	0	0.029446
38	Oklahoma	US	2465	136	0	0.055172
39	Minnesota	US	2209	121	0	0.054775
40	Oregon	US	1844	72	0	0.039045
41	Kansas	US	1821	85	0	0.046677
42	New Mexico	US	1758	53	0	0.029577
43	Arkansas	US	1744	38	0	0.021789
44	Idaho	US	1655	43	0	0.025981




```
print('Outside Mainland China {} cases:'.format(outside_mainland_china_confirmed))
print('Mainland China: {} cases'.format(china_confirmed))
```

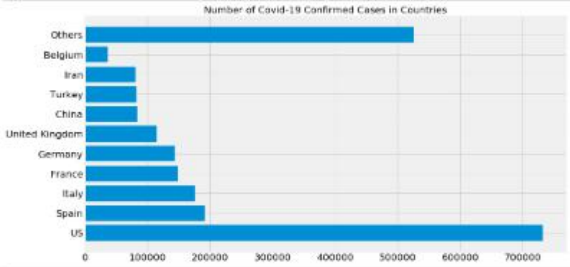
```
Outside Mainland China 2233972 cases:
Mainland China: 83787 cases
Total: 2317759 cases
```

```
visual_unique_countries = []
visual_confirmed_cases = []
others = np.sum(country_confirmed_cases[10:])

for i in range(len(country_confirmed_cases[:10])):
    visual_unique_countries.append(unique_countries[i])
    visual_confirmed_cases.append(country_confirmed_cases[i])

visual_unique_countries.append('Others')
visual_confirmed_cases.append(others)
```

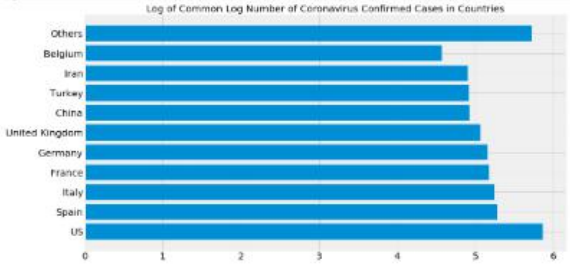
Visual Representations (bar charts and pie charts)

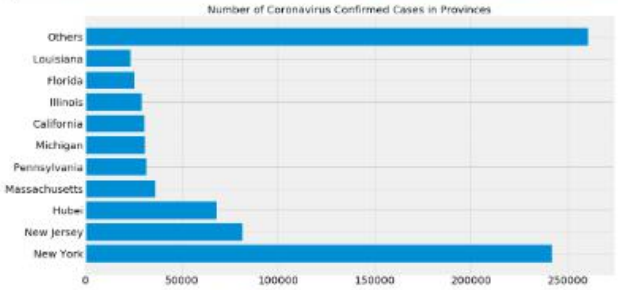
```
def plot_bar_graphs(x, y, title):
    plt.figure(figsize=(16, 9))
    plt.barh(x, y)
    plt.title(title, size=20)
    plt.xticks(size=20)
    plt.yticks(size=20)
    img_data = io.BytesIO()
    plt.savefig(img_data, format='png')
    img_data.seek(0)
    s3 = boto3.resource('s3')
    bucket1 = s3.Bucket(bucket)
```

```
plot_bar_graphs(visual_unique_countries, visual_confirmed_cases, 'Number of Covid-19 Confirmed Cases in Countries')
```

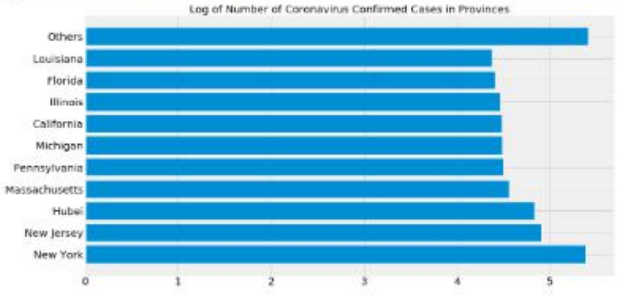


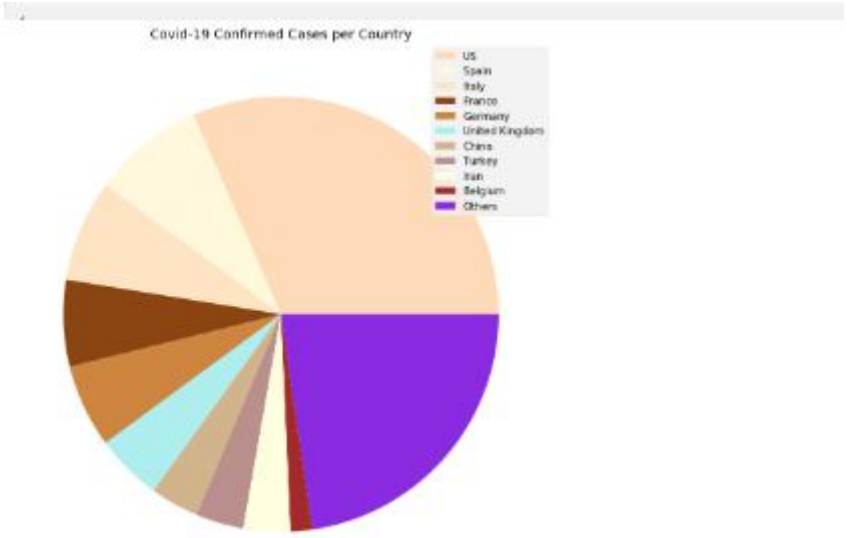

```
log_country_confirmed_cases = [math.log10(i) for i in visual_confirmed_cases]
```



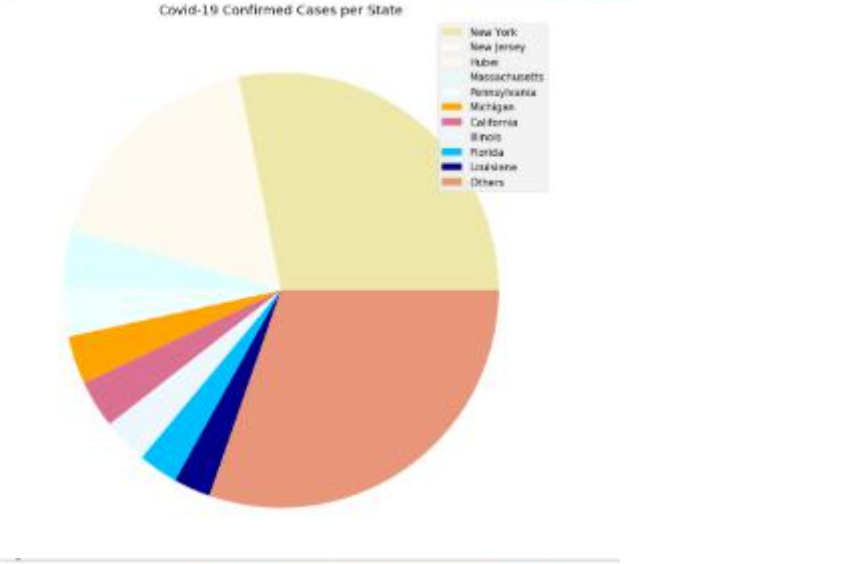



```
log_province_confirmed_cases = [math.log10(i) for i in visual_confirmed_cases2]
```

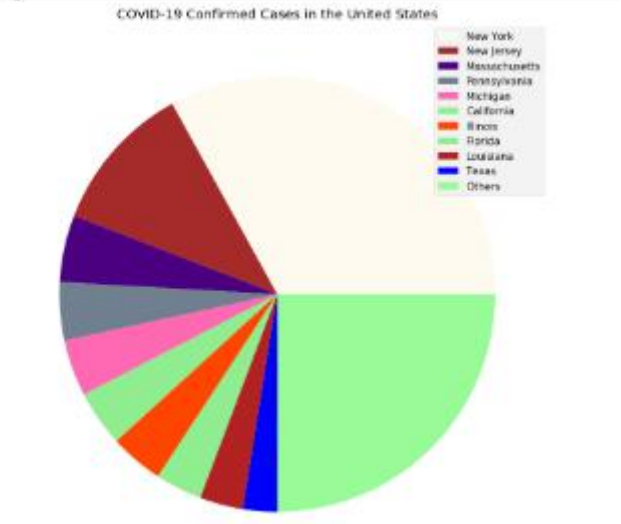





```
[73]: plot_pie_charts(visual_unique_provinces, visual_confirmed_cases2, 'Covid-19 Confirmed Cases per State')
```

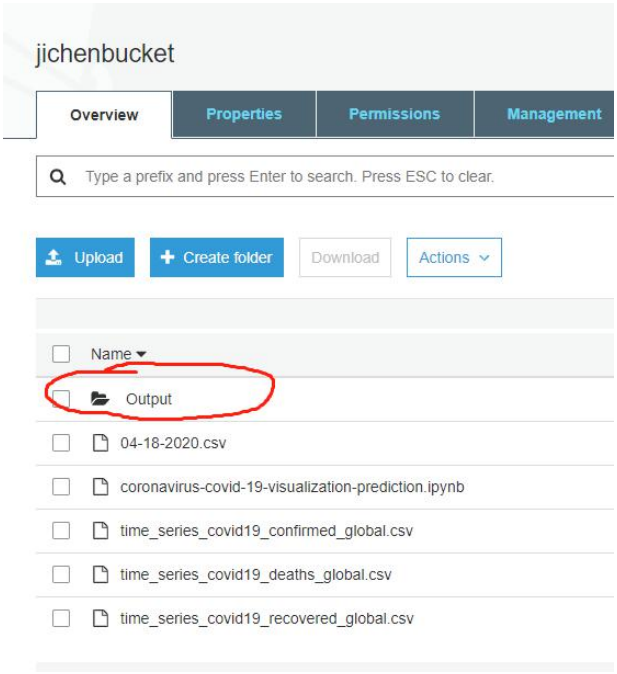



```
[74]: plot_pie_country_with_regions('US', 'COVID-19 Confirmed Cases in the United States')
```

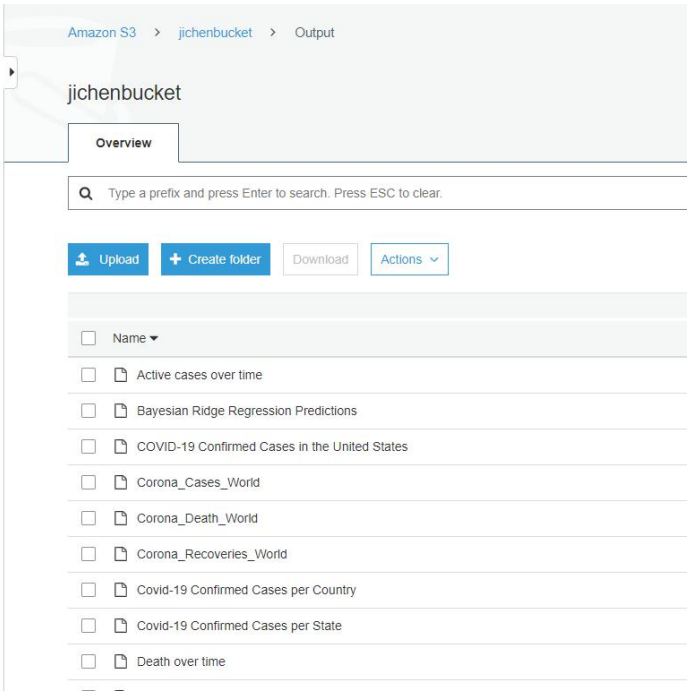


G. Check the results in the *output folde*r in the Amazon S3 bucket you had created.

1. Go to the bucket, there is a output folder



Output folder includes all output files we generated.



Cloclusion:

As we can see, the SageMaker provides a powerful platform for us to deploy our projects. And it cooperates with S3 storage and IAM to give us a great experience. We don't need to consider about things such as environment, thus we can concentrate more on our project.