

---

# Machine Learning HW1

# COVID-19 Cases Prediction

ML TAs

[ntu-ml-2021spring-ta@googlegroups.com](mailto:ntu-ml-2021spring-ta@googlegroups.com)

---

# Outline

- Objectives
- Task Description
- Data
- Evaluation Metric
- Kaggle
- Grading
- Code Submission
- Deadlines
- Hints
- Regulations again
- Useful Links

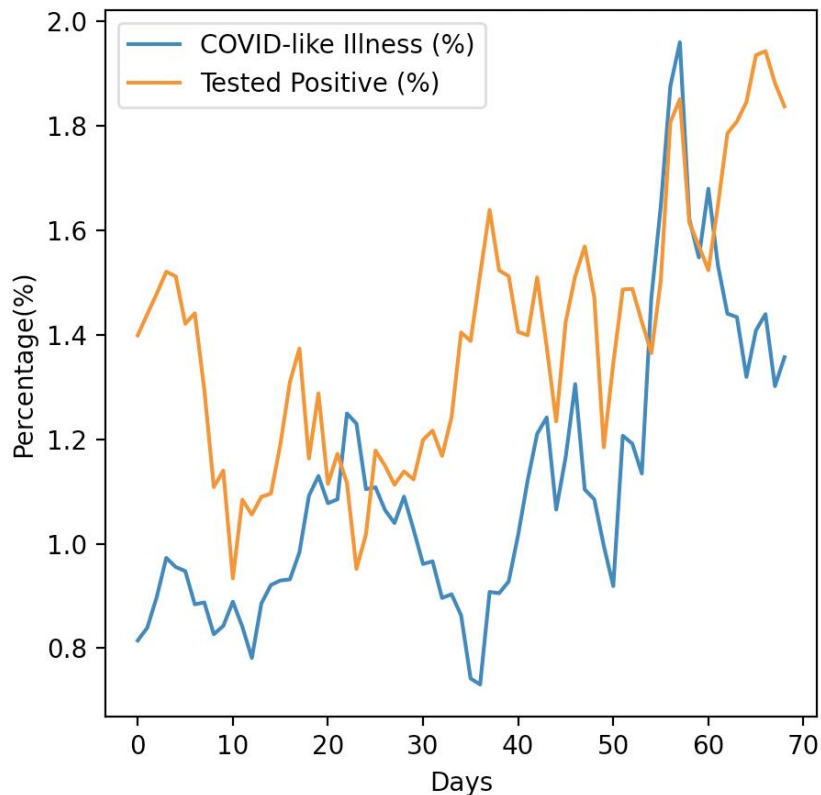
# Objectives

- Solve a **regression** problem with **deep neural networks** (DNN).
- Understand basic DNN training tips  
e.g. hyper-parameter tuning, feature selection, regularization, ...
- Get familiar with **PyTorch**.

# Task Description

- **COVID-19 Cases Prediction**
- Source: Delphi group @ CMU
  - A daily survey since April 2020 via facebook.

Do not attempt to find any related data!  
Using additional data is prohibited and  
your final grade x 0.9 !

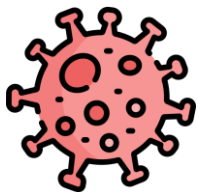


# Task Description

- Given survey results in the **past 3 days** in a specific **state** in U.S., then predict the percentage of **new tested positive cases** in the 3rd day.



survey

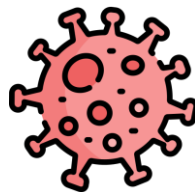


positive  
cases

Day 1



survey



positive  
cases

Day 2



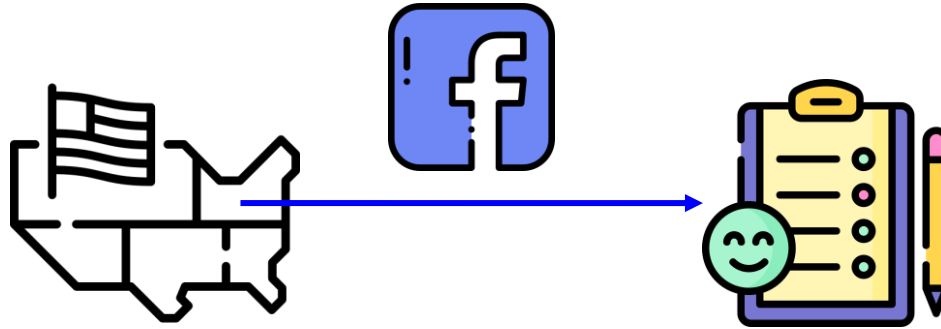
survey



**positive  
cases**

Day 3

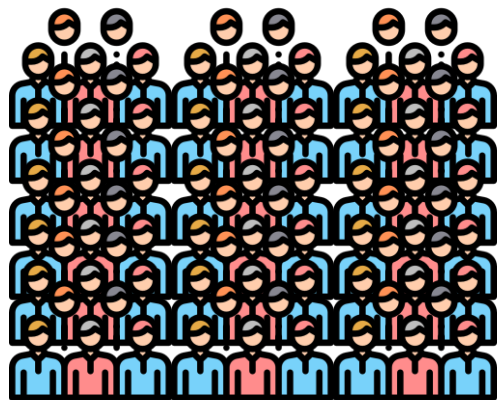
# Data -- Delphi's COVID-19 Surveys



Conducted surveys via facebook (every day & every state)

Survey: symptoms, COVID-19 testing, social distancing, mental health, demographics, economic effects, ...

# Data -- Delphi's COVID-19 Surveys



All population in a  
certain state of the U.S.



some samples



survey



**estimation for all  
population in that  
state  
(data we are using)**

# Data -- Delphi's COVID-19 Surveys

- **States** (40, encoded to **one-hot** vectors)
  - e.g. AL, AK, AZ, ...
- **COVID-like illness** (4)
  - e.g. cli, ili (influenza-like illness), ...
- **Behavior Indicators** (8)
  - e.g. wearing\_mask, travel\_outside\_state, ...
- **Mental Health Indicators** (5)
  - e.g. anxious, depressed, ...
- **Tested Positive Cases** (1)
  - **tested\_positive** (this is what we want to predict)

} Percentage



# Data -- One-hot Vector

- **One-hot vectors:**

Vectors with **only one element equals to one** while others are zero.

Usually used to encode discrete values.

If state code = AZ  
(Arizona)

**one-hot encoding**



0	AL (Alabama)
0	AK (Alaska)
1	AZ (Arizona)
0	AR (Arkansas)
⋮	
0	WI (Wisconsin)

# Data -- Training

covid.train.csv (2700 samples)

**state one-hot  
encoding (40)**

## Day 1 features (18)

## Day 2 features (18)

## Day 3 features (18)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

**tested  
positive**

## 1 row = 1 sample

# Data -- Testing

covid.test.csv (893 samples)

## state one-hot encoding (40)

## Day 1 features (18)

## Day 2 features (18)

## Day 3 features (17)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

## 1 row = 1 sample

# Evaluation Metric

- Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^N (f(\mathbf{x}^n) - \hat{y}^n)^2}$$

Diagram illustrating the components of the RMSE formula:

- $f(\mathbf{x}^n)$ : your model
- $\mathbf{x}^n$ : input features (testing data)
- $\hat{y}^n$ : ground truth label (correct answer)

# Kaggle

- Link: TBA
- Displayed name: **<student ID>\_<anything>**
  - e.g. b06901020\_puipui
  - For auditing, don't put student ID in your displayed name.
- Submission format: **.csv** file
  - See sample code

```
1 id,tested_positive
2 0,0.0
3 1,0.0
4 2,0.0
5 3,0.0
6 4,0.0
```

# Kaggle -- Submission

- You may submit up to **5** results each day (UTC).
- Up to **2** submissions will be considered for the private leaderboard.

<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large3_684_compressed.pth, size = 201KB, params: 93139 (rabbit ensemble)	0.65059	0.66341	<input checked="" type="checkbox"/>
<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large3_676_compressed.pth, size = 201KB, params: 93139 (rabbit ensemble)	0.65282	0.65422	<input type="checkbox"/>
<a href="#">prediction_large.csv</a> 2 years ago by <a href="#">ntuee_jizz</a> model_large2_669_compressed.pth, size = 222KB, params: 103623	0.65394	0.65254	<input checked="" type="checkbox"/>

remember to select **2**  
results for your final  
scores before the  
competition ends!

# Grading

- Simple baseline (public) +1 pt (sample code)
- Simple baseline (private) +1 pt (sample code)
- Medium baseline (public) +1 pt
- Medium baseline (private) +1 pt
- Strong baseline (public) +1 pt
- Strong baseline (private) +1 pt
- Upload code to NTU COOL +4 pts

Total: 10 pts

# Grading -- Kaggle

- The baselines might be modified.

#	Team Name	Notebook	Team Members	Score ?	Entries	Last
📍	----- strong baseline -----			0.88017		
📍	----- medium baseline -----			1.08443		
📍	----- simple baseline -----			2.03004		



# Grading -- Bonus

- If you got 10 points, we make your code **public** to the whole class.
- In this case, if you also submit **a PDF report briefly describing your methods** (<100 words in English), you get a bonus of **0.5 pt.**  
(your report will also be available to all students)
- [Report template](#)

# Code Submission

- **NTU COOL** (4pts)

- Compress your code and report into

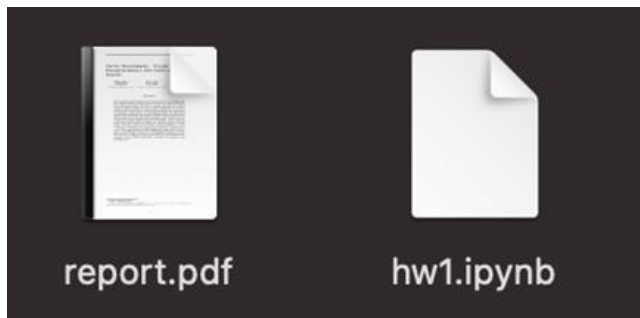
**<student ID>\_hw1\_v<version>.zip**

**e.g. b06901020\_hw1\_v1.zip (versions: 1, 2, 3, ...)**

- Do not submit your model or dataset.
- If your code is not reasonable, your semester grade x 0.9.

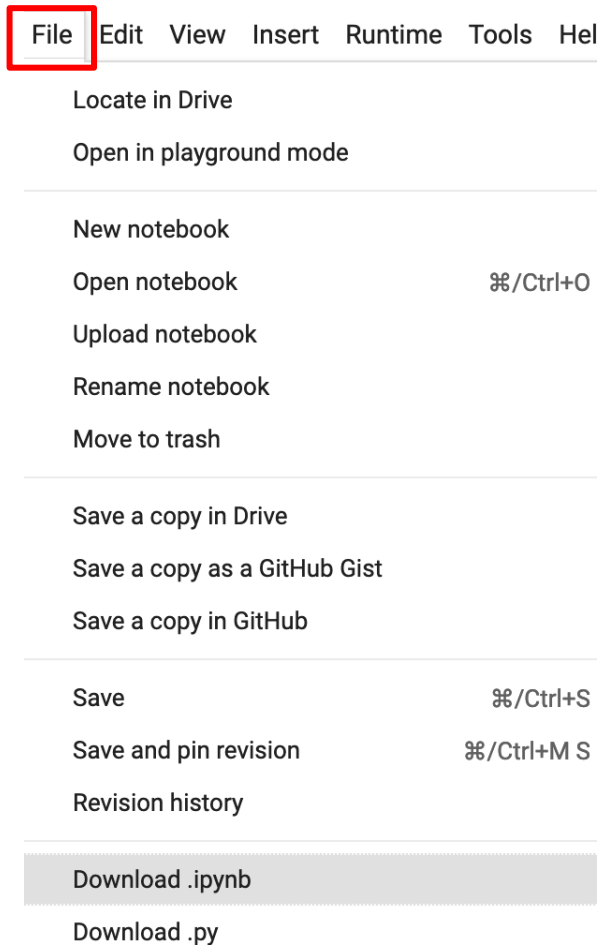
# Code Submission

- Your .zip file should include only
  - **Code:** either .py or .ipynb
  - **Report:** .pdf (only for those who passed **all** baselines)
- Example:



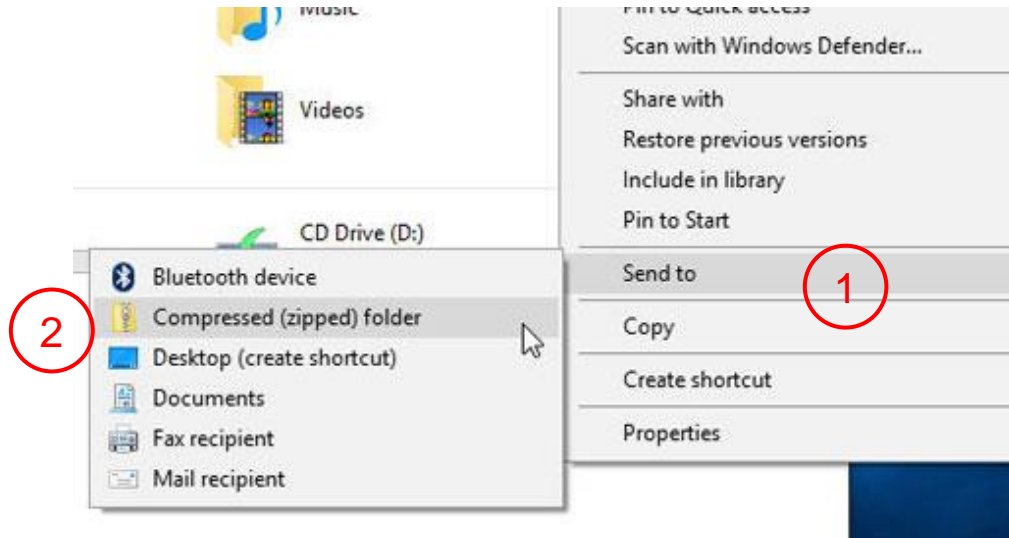
# Code Submission

- How to download your code from Google Colab?



# Code Submission

- How to compress your folder?
- Method 1 (for Windows users)
  - <https://support.microsoft.com/en-us/windows/zip-and-unzip-files-f6dde0a7-0fec-8294-e1d3-703ed85e7ebc>



# Code Submission

- How to compress your folder?
- Method 2 (for Mac users)
  - <https://support.apple.com/guide/mac-help/mchlp2528/mac>



Compress “b06901020\_hw1\_v1”

# Code Submission

- How to compress your folder?
- Method 3 (command line)

```
zip -r <name>.zip <directory name>
```

e.g.

```
zip -r b06901020_hw1_v1.zip b06901020_hw1_v1
```

# Deadlines

- Kaggle

**2021/03/26 23:59 (UTC+8)**

- Code Submission (NTU COOL)

**2021/03/28 23:59 (UTC+8)**

**No late submission!**  
**Submit early!**



# Hints

- **Simple Baseline**

- Sample code (TBA)

- **Medium Baseline**

- Feature selection: 40 states + 2 tested\_positive  
(will be demonstrated in class)

- **Strong Baseline**

- Feature selection (what other features are useful?)
- DNN architecture (layers? dimension? activation function?)
- Training (mini-batch? optimizer? learning rate?)
- L2 regularization
- There are some mistakes in the sample code, can you find them?

# Regulations Again

- You should finish your homework on your own.
- You should NOT modify your prediction files manually.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- Do NOT search or use additional data or pre-trained models.
- Your **final grade x 0.9** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

# If any questions, you can ask us via...

- NTU COOL (recommended)
  - <https://cool.ntu.edu.tw/courses/4793>
- Email
  - [ntu-ml-2021spring-ta@googlegroups.com](mailto:ntu-ml-2021spring-ta@googlegroups.com)
  - Title should begin with “[hw1]”
- TA hour
  - Each Friday during class

# Useful Links

- Hung-yi Lee, Regression & Gradient Descent (Mandarin)
  - ([link1](#), [link2](#), [link3](#), [link4](#), [link5](#), [link6](#))
- Hung-yi Lee, Tips for Training Deep Networks (Mandarin)
  - ([link1](#), [link2](#))
- Google Machine Learning Crash Course (English)
  - ([Regularization](#), [NN Training](#))
- <https://pytorch.org/docs/stable/index.html>
- <https://www.google.com/>

(If Google or Stackoverflow can solve your problems, you may take advantage of them before asking TAs.)

Have fun and wish you good luck!

