

Hey Guys I m Gul Mohammad jin

Today i will show you [Predict of corona patient]

LogisticRegression in Machine Learning

```
In [109]: import pandas as pd
```

```
In [85]: # Data Read &
```

```
In [86]: df = pd.read_csv('E:\\study material\\corona virus\\coron.csv')
```

```
In [87]: df.head()
```

Out[87]:

	fever	body pain	age	runny nose	diff Breath	infection prob
0	101.430404	1	4	0	-1	0
1	98.510608	1	40	1	-1	0
2	99.867925	0	26	0	0	1
3	101.539033	0	7	0	0	0
4	99.141592	0	4	1	0	0

In [88]: `df.tail()`

Out[88]:

	fever	body pain	age	runny nose	diff Breath	infection prob
995	100.156470	0	100	0	1	0
996	99.151210	0	12	0	-1	1
997	99.005036	1	45	1	-1	1
998	100.252633	0	64	1	1	0
999	98.849532	0	53	1	-1	1

In [89]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
fever          1000 non-null float64
body pain      1000 non-null int64
age            1000 non-null int64
runny nose     1000 non-null int64
diff Breath    1000 non-null int64
infection prob 1000 non-null int64
dtypes: float64(1), int64(5)
memory usage: 47.0 KB
```

In [90]: `df['diff Breath'].value_counts()`

Out[90]:

-1	370
0	317
1	313

Name: diff Breath, dtype: int64

```
In [91]: df.describe()
```

```
Out[91]:
```

	fever	body pain	age	runny nose	diff Breath	infection prob
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	99.995418	0.496000	51.141000	0.508000	-0.057000	0.504000
std	1.125076	0.500234	29.119187	0.500186	0.824883	0.500234
min	98.004185	0.000000	1.000000	0.000000	-1.000000	0.000000
25%	99.027256	0.000000	26.000000	0.000000	-1.000000	0.000000
50%	100.009925	0.000000	52.000000	1.000000	0.000000	1.000000
75%	100.901269	1.000000	76.000000	1.000000	1.000000	1.000000
max	101.995561	1.000000	100.000000	1.000000	1.000000	1.000000

```
In [92]: import numpy as np
```

```
In [93]: def data_split(data,ratio):
    np.random.seed(42)
    shuffled = np.random.permutation(len(data))
    test_set_size = int(len(data)*ratio)
    test_indices = shuffled[:test_set_size]
    train_indices = shuffled[test_set_size:]
    return data.iloc[train_indices], data.iloc[test_indices]
```

```
In [94]: np.random.permutation(5)
```

```
Out[94]: array([0, 2, 1, 3, 4])
```

```
In [95]: train , test = data_split(df , 0.2)
```

In [96]: train

Out[96]:

	fever	body pain	age	runny nose	diff Breath	infection prob
29	98.023486	0	54	1	1	0
535	98.202097	1	12	0	-1	1
695	99.833881	0	68	1	0	0
557	99.372948	0	80	1	0	1
836	98.574676	1	32	1	0	1
596	100.893933	0	14	1	-1	1
165	100.460939	1	67	1	0	1
918	101.982625	1	26	0	1	0
495	101.820307	0	93	1	0	0
824	99.531637	1	45	0	0	1
65	99.626269	0	54	1	0	0
141	98.518052	0	62	0	0	1
925	100.080126	0	41	1	-1	0
827	100.560447	0	47	1	-1	0
655	99.105027	0	79	1	-1	0
331	98.600234	1	49	1	-1	1
664	101.569355	1	79	1	1	0
249	99.315782	0	57	0	0	0
907	99.242680	0	84	1	1	0
708	98.555907	1	28	1	-1	0
305	101.464882	1	43	1	0	0
734	98.966712	1	71	1	-1	0
975	99.435719	0	24	0	-1	1
49	100.873500	1	51	0	-1	1

	fever	body pain	age	runny nose	diff Breath	infection prob
896	98.423424	1	13	1	0	1
2	99.867925	0	26	0	0	1
544	100.509471	0	89	1	-1	0
350	100.405940	1	10	0	0	0
904	99.817886	0	23	1	-1	1
536	101.227293	0	49	1	1	0
...
955	101.162042	1	27	1	1	0
191	98.153799	1	23	0	-1	1
385	101.438849	0	52	1	0	0
805	99.946429	1	36	1	-1	0
413	100.458326	1	23	0	0	1
491	100.609635	0	32	0	-1	1
343	100.147432	0	1	1	1	0
769	98.220937	1	42	1	0	1
308	99.013583	1	91	1	-1	1
661	98.279778	0	26	1	1	1
130	99.920404	1	14	0	-1	1
663	101.775001	1	3	0	-1	0
871	99.216975	0	72	1	-1	1
99	100.981200	0	64	1	0	1
372	100.433521	0	50	1	-1	1
87	99.903254	0	88	1	0	1
458	98.539105	0	68	0	0	1
330	100.197864	0	75	1	-1	1
214	99.329473	1	87	0	-1	0

	fever	body pain	age	runny nose	diff Breath	infection prob
466	101.184618	1	70	0	1	1
121	99.774325	1	11	1	-1	1
614	98.702535	1	18	0	1	1
20	99.070810	0	24	0	1	0
700	99.390375	1	31	1	1	0
71	101.319310	1	69	1	-1	0
106	99.196517	0	99	0	-1	0
270	100.428270	0	75	1	0	0
860	98.377474	0	63	1	-1	0
435	98.259869	0	38	1	1	1
102	99.555726	0	95	0	-1	0

800 rows × 6 columns

In [97]: test

Out[97]:

	fever	body pain	age	runny nose	diff Breath	infection prob
521	100.078610	0	45	0	-1	0
737	100.012449	1	75	0	0	0
740	98.917878	1	86	0	0	0
660	98.641350	1	23	1	0	1
411	98.865577	0	48	1	0	1
678	100.392687	0	57	0	0	0
626	100.872274	0	80	1	1	1
513	98.422834	1	87	0	-1	0
859	99.479727	1	36	1	1	0
136	101.679779	0	55	1	0	1
811	101.152548	0	7	0	1	1
76	100.502737	0	38	0	-1	1
636	100.826354	0	65	1	-1	0
973	100.074081	0	99	1	0	1
938	98.652307	1	43	0	0	1
899	100.051746	1	16	1	0	0
280	99.820127	0	78	0	1	0
883	99.109447	0	85	1	-1	0
761	100.684078	1	52	0	-1	1
319	101.652093	0	1	0	-1	1
549	98.946068	1	6	0	-1	0
174	100.093883	1	97	1	0	0
371	100.645795	1	20	0	-1	1
527	99.047478	1	64	0	1	0

	fever	body pain	age	runny nose	diff Breath	infection prob
210	101.825944	0	17	0	1	1
235	98.565150	0	63	1	1	0
101	100.460017	1	80	1	-1	1
986	100.092234	1	15	1	-1	0
902	99.247889	1	85	0	1	1
947	98.172237	0	83	0	-1	0
...
361	99.658999	1	93	0	-1	0
479	98.618326	1	83	1	0	1
110	100.747540	1	16	0	1	1
989	101.628722	0	91	0	-1	1
486	100.443834	1	92	0	0	1
363	98.851775	1	20	1	0	0
254	98.322625	1	66	1	0	1
259	99.015661	0	76	0	1	0
802	99.486748	0	16	0	-1	0
677	98.951247	1	57	0	-1	1
494	101.842212	0	77	0	1	1
670	100.172695	1	82	0	1	0
377	99.358409	1	91	1	-1	1
526	101.059635	1	81	1	-1	1
845	100.848644	0	43	0	0	1
137	98.788694	1	9	0	-1	0
355	98.818224	0	44	0	1	0
365	98.440752	1	100	1	-1	1
942	99.677333	1	93	0	-1	0

	fever	body pain	age	runny nose	diff Breath	infection prob
749	100.775878	1	27	0	1	0
948	99.438877	0	31	0	-1	0
829	100.140127	1	51	0	0	0
656	98.014484	0	11	0	0	0
199	100.511391	0	54	0	0	1
213	101.119810	1	29	0	-1	0
408	101.002555	0	65	0	0	1
332	100.107339	0	55	1	0	1
208	99.991814	0	83	0	1	1
613	101.557613	0	64	0	1	0
78	98.715527	0	90	0	0	1

200 rows × 6 columns

```
In [98]: X_train = train[['fever','body pain','runny nose','diff Breath' , 'age']]
```

In [99]: X_train

372	100.433521	0	1	-1	50
87	99.903254	0	1	0	88
458	98.539105	0	0	0	68
330	100.197864	0	1	-1	75
214	99.329473	1	0	-1	87
466	101.184618	1	0	1	70
121	99.774325	1	1	-1	11
614	98.702535	1	0	1	18
20	99.070810	0	0	1	24
700	99.390375	1	1	1	31
71	101.319310	1	1	-1	69

In [100]: X_test

```
[ 99.35840936,  1.      ,  1.      , -1.      ,
   91.      ],
[101.0596351 ,  1.      ,  1.      , -1.      ,
   81.      ],
[100.8486443 ,  0.      ,  0.      ,  0.      ,
   43.      ],
[ 98.78869397,  1.      ,  0.      , -1.      ,
   9.       ],
[ 98.81822399,  0.      ,  0.      ,  1.      ,
   44.      ],
[ 98.44075245,  1.      ,  1.      , -1.      ,
  100.      ],
[ 99.67733349,  1.      ,  0.      , -1.      ,
   93.      ],
[100.7758782 ,  1.      ,  0.      ,  1.      ,
   27.      ],
[ 99.43887679,  0.      ,  0.      , -1.      ,
   31.      ],
[100.1401266 ,  1.      ,  0.      ,  0.      ,
   51.      ],
```

```
In [101]: X_train = train[['fever','body pain','runny nose','diff Breath','age']].to_numpy()
X_test = test[['fever','body pain','runny nose','diff Breath','age']].to_numpy()
```

```
In [102]: Y_train = train[['infection prob']].to_numpy().reshape(800 )
Y_test = test[['infection prob']].to_numpy().reshape(200 )
```

```
In [103]: Y_train
```

[illegible]

```
In [104]: from sklearn.linear_model import LogisticRegression
```

```
In [105]: clf=LogisticRegression()  
clf.fit(X_train,Y_train)
```

C:\Users\JIN USTAD\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
FutureWarning)

```
Out[105]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,  
                             intercept_scaling=1, max_iter=100, multi_class='warn',  
                             n_jobs=None, penalty='l2', random_state=None, solver='warn',  
                             tol=0.0001, verbose=0, warm_start=False)
```

```
In [106]: clf.predict([[100,1,22,1,1]])
```

```
Out[106]: array([0], dtype=int64)
```

```
In [107]: clf.predict_proba([[100,1,22,1,1]])
```

```
Out[107]: array([[0.9989401, 0.0010599]])
```

```
In [108]: clf.predict_proba([[100,1,22,1,1]])[0][1]
```

```
Out[108]: 0.0010599044754561774
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```