

# phoneme\_Shivani

October 17, 2021

```
[1]: %load_ext autoreload
      %autoreload 2
      %matplotlib inline
```

```
[2]: import sys
      from pathlib import Path
      import matplotlib.pyplot as plt
```

```
[3]: if 'source' not in sys.path:
      sys.path.append('source')
```

```
[4]: import phoneme
      from phoneme import main
```

```
[23]: main()
```

Perceptron accuracy score

0.967

Logistic Regression accuracy score

0.950

classifier |  $\mu$  | 

without preprossesing

P	0.899	0.014
---	-------	-------

L	0.897	0.007
---	-------	-------

R	0.907	0.010
---	-------	-------

With Data standardizatin

P	0.884	0.007
---	-------	-------

L	0.887	0.009
---	-------	-------

R	0.904	0.010
---	-------	-------

1. Yes the training data linerly seprable, has we are getting good accuracy score. Both perceptron and logistic regression are good at classifying the outputs that are +1 and -1, and since its trying to fit to the data and finding a good line its not surprising
2. (without data preprossesing) R = logistic regression, with offset, with L1-regularization ( $C = 1$ ), failed to converge, so I had to limit it with the solver = "lbfgs" and Max iterations to 1000 (based on ths discussion on stackOverflow : <https://stackoverflow.com/a/66560912/16838276>). R model seems to perform better than

the rest, that is with the offset and l1 regularization but it was supposed to be taken extra care.

3. (with standardization) The problem of lbfgs error did not arrive after standardization with feature standardization method. R model perform the best with stndardization. Overall the mean value is decreassing after data standardization, but standard devivatio is decreasing only for P which is good. Now if we compare with and without preprossing R will be left, but since we had to limit the iterations for R before preprocessing. I would say R(after standerdisation), performs best.

[ ]: