|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Features | KFold | Best Acc | Best Acc Model Type | Second Best Acc | Second Best Acc Model Type | Gaussian Naive Bayes Acc | Trained Model Name |
| 1 | 9/9 | 5 | 56.6% | Ensemble: Subspace Discriminant | 54.2% | Linear Discriminant: Linear Discriminant | 47.8% | trainedModel\_KF\_5\_AllFeatures |
| 2 | 9/9 | 10 | 54.9% | Ensemble: Bagged Trees | 54.9% | Tree: Medium Tree | 46.5% | trainedModel\_KF\_10\_AllFeatures |
| 3 | 9/9 | 15 | 58.2% | Ensemble: Bagged Trees | 55.2% | Tree: Medium Tree | 44.8% | trainedModel\_KF\_15\_AllFeatures |
| 4 | 9/9 | 20 | 57.2% | Ensemble: Bagged Trees | 54.9% | Ensemble: Subspace Discriminant | 46.5% | trainedModel\_KF\_20\_AllFeatures |

Summery:

In this report you can see an overview of best training models for Persian LinkedIn Posts dataset with MATLAB classification apps.

As the table above shows, the best model type for this dataset is “Ensemble: Bagged Trees”, with the highest accuracy of “58.2%” in Kfold=15, all features included. In addition, the “Gaussian Naive Bayes” model seems not be an efficient training model for this dataset.

Besides, according to several tries in removing some features (for example exclude of having an image/video/hashtags in posts) during training, I have not achieved better accuracy than all features, meaning I got less accuracy as the result.

In conclusion, I think if I had more data samples, I would have had e better training model.