Handwritten Image Classification

Name: Fatemeh Kiaie

Date: July 13, 2019

Classification Project

- **Data Description:**

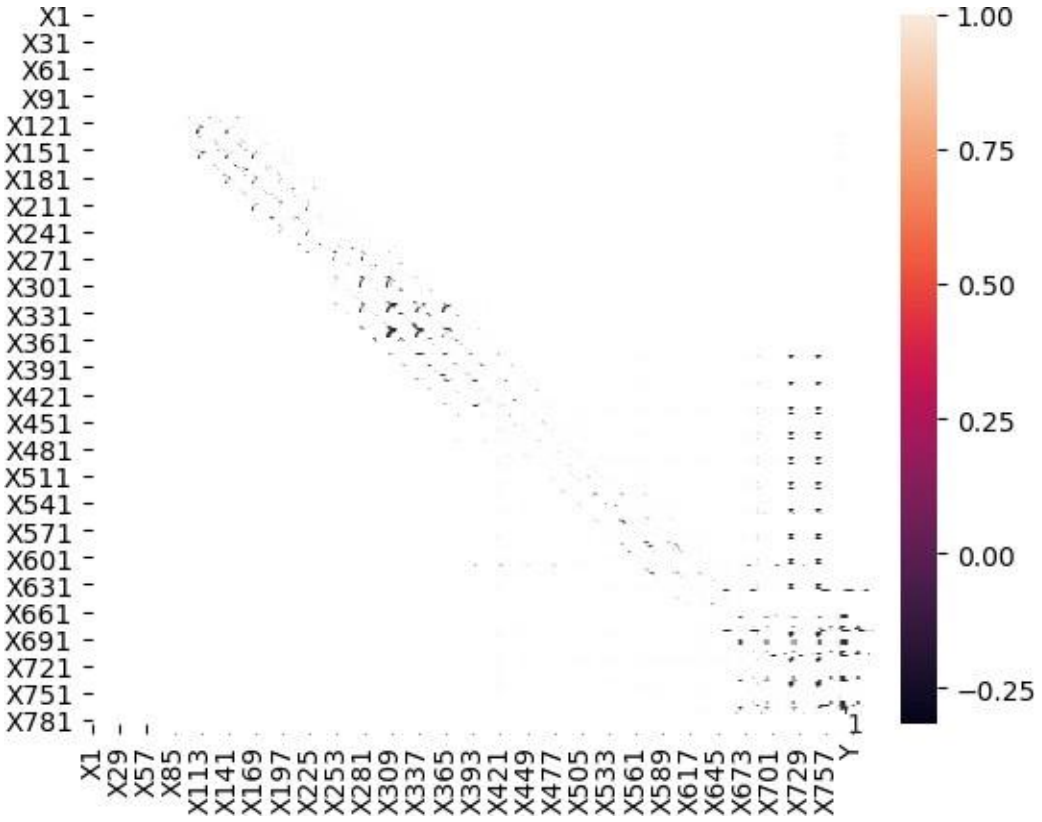
This dataset, obtained from Kaggle website, contains A-Z handwritten images. Each image is stored as a Gray-level in the size of 28*28 pixels. Therefore, dataset is a frame of 784+1 attributes (representing each image and the target column, corresponding A-Z letter) for a total number of 42240 instances. The goal is to classify the handwritten English alphabetic letter images.

- **Data preprocessing:**

1. Random sampling from data: due to the huge number of instances in dataset, a random sample of data containing 5000 instances are selected for data exploration and analysis.
2. Handling missing data: no missing value.
3. Encoding: not required because all features are numeric.
4. Dropping unnecessary feature: we do not need to drop any columns and we need to use all features and attributes as they add value to data.
5. Scaling: not required since the data are almost in the same range of values.

- **Correlation Analysis:**

1. Correlation analysis: using the spearman coefficient, heat map shows that there is a relatively weak correlation among the neighbor pixels.



2. Feature Elimination (RFE): After running RFE using Logistic Regression model, the results show that there is no dominant feature to be eliminated.

3. PCA implementation: pca implementation also confirms that there is not any specific feature that makes a high variance among the data and there is not any feature to remove in order to reduce the dimension of the dataframe.

➢ Results

As can be seen in Fig. 1, in the models with the default parameter, SVM performs better than the other models. This is evident that SVM is a good model for image classification since in its formulation it can measures how far apart two images are. It is also can be seen that the AdaBoos is a biased model. To evaluate over fitting model, I have shown the differences between the test and training score in Fig.2. As can be seen logistic regression is over fitting model and from Fig. 1 and 2, it is clear that Adaboost is under fit (or bias) model.

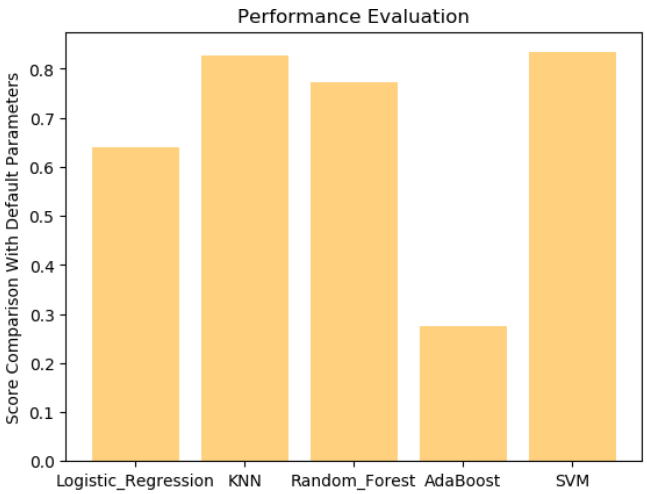


Fig. 1 Models' score with the default parameter

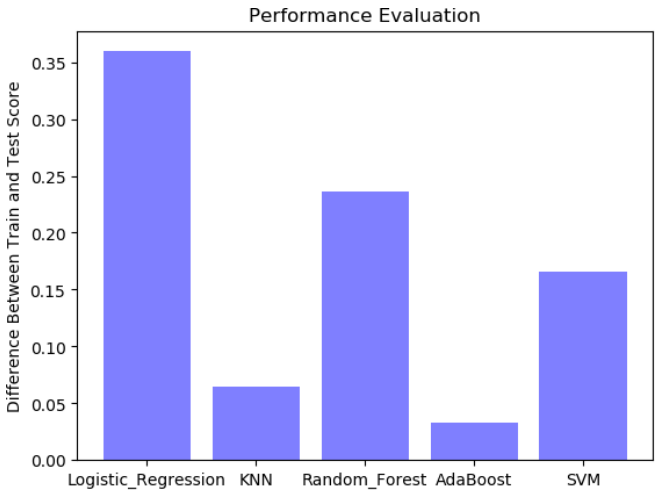


Fig. 2. Over-fitting values comparison

To get a better performance from the models, 4fold cross validation and the grid search cv methods are implemented and the results are shown in Fig. 4 and Fig. 5.

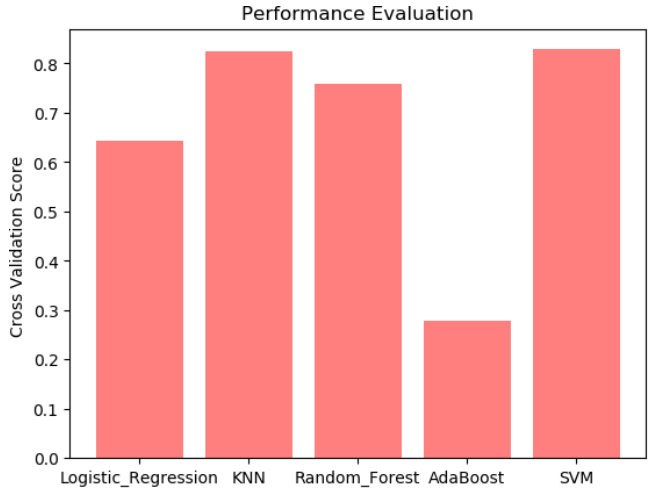


Fig. 4 Models' score with cross validation method

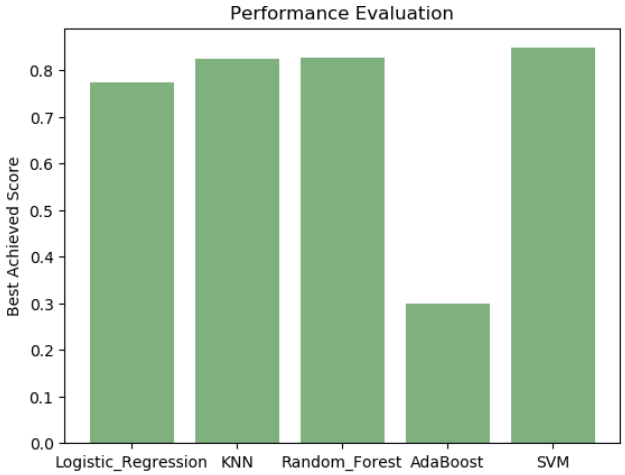
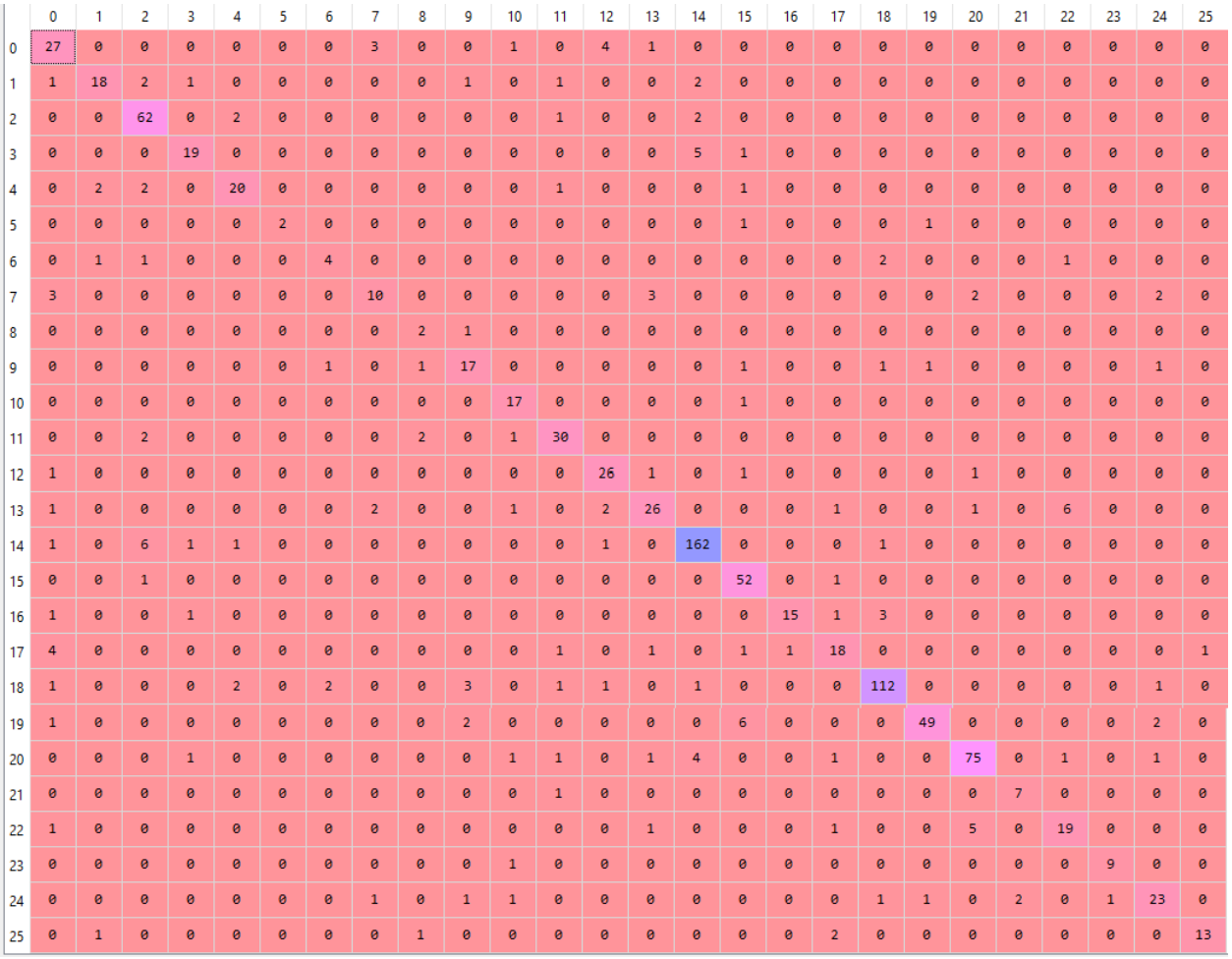


Fig. 5. Best Score of the models from gridsearch



	0	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
0	27	0	0	0	0	0	0	3	0	0	1	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	18	2	1	0	0	0	0	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0
2	0	0	62	0	2	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	19	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0
4	0	2	2	0	20	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
6	0	1	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0
7	3	0	0	0	0	0	0	10	0	0	0	0	0	3	0	0	0	0	0	0	2	0	0	0	2	0
8	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	1	17	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11	0	0	2	0	0	0	0	0	2	0	1	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	26	1	0	1	0	0	0	0	1	0	0	0	0	0
13	1	0	0	0	0	0	0	2	0	0	1	0	2	26	0	0	0	1	0	0	1	0	6	0	0	0
14	1	0	6	1	1	0	0	0	0	0	0	0	1	0	162	0	0	0	1	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	52	0	1	0	0	0	0	0	0	0	0
16	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	1	3	0	0	0	0	0	0	0
17	4	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	18	0	0	0	0	0	0	0	1
18	1	0	0	0	2	0	2	0	0	3	0	1	1	0	1	0	0	0	112	0	0	0	0	0	1	0
19	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	6	0	0	0	49	0	0	0	0	2	0
20	0	0	0	1	0	0	0	0	0	0	1	1	0	1	4	0	0	1	0	0	75	0	1	0	1	0
21	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	7	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	5	0	19	0	0	0
23	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
24	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	1	0	2	0	1	23	0
25	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	13

Fig. 3. Confusion Matrix

As can be seen in Fig. 5, other than Adaboost, the models performing similarly with a slightly better performance in SVM. Although with trying more range of the parameters in the grid search we may be able to get more improvement in the models performance. The best parameters of the models from the selected parameter are listed below

- Best Logistic regression performance obtained with {'C': 0.001, 'penalty': 'l1'}
- Best KNN performance is obtained with {'n_neighbors': 5}
- Best Random Forest performance is obtained with {'max_depth': 15, 'n_estimators': 50}
- Best Adaboost performance is obtained with {'max_depth': 15, 'n_estimators': 50}
- Best SVM performance is obtained with {'C': 1.0, 'kernel': 'poly'}