AAUT: Machine learning

Project: Regression and classification

Piotr Kasprowicz (101310)

January 4, 2023

Contents

1 Regression

1.1 First problem

Goal of the first problem was to train predictor $\hat{y} = f(x)$ given a training set with 100 examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(100)}, y^{(100)})\}$, where $x^i \in \mathbb{R}^{20}$ and $y \in \mathbb{R}$. Purpose of the experiment is to compare different predictors and select one which will suit best our training set. Final evaluation of predictor will be done using a different set of test data provided by professor that cannot be used to select the predictors.

1.1.1 Methodology

In experiment 3 different model predictors will be compared: Linear Regression, Lasso and Ridge method. In order to compare results of them we will use 10-fold cross validation, as described below:

- Shuffle the dataset randomly.
- Split the dataset into 10 groups.
- For each unique group:
  a. Take the group as a hold out or validation data set.
  b. Take the remaining groups as a training data set
  c. Fit a model on the training set and evaluate it on the validation set
- Summarize the skill of the model by calculating mean of 10 MSE and standard deviation

Score of the predictors will me measured by **mean squared error(MSE)**.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

The predictor with the lowest score will be selected to predict validation set(test set provided by professor). Python library *scikit-learn* with alredy implemeted algirthms will be used to simplify the experiment.

1.1.2 Results

Table 1: Result for each of 10 folds in cross validation using different predictors.

	Model		
Iteration	Linear regression	Lasso	Ridge
1	0.01651893	3.98436997	0.02658214
2	0.01852347	10.13411738	0.02158626
3	0.01193437	8.76690991	0.0244795
4	0.02568395	8.45905951	0.02970907
5	0.01708623	9.54670603	0.0185978
6	0.0090921	9.87843893	0.01445698
7	0.01006048	7.33776117	0.0130735
8	0.02526528	18.38627736	0.03110652
9	0.01806075	6.79048614	0.01420189
10	0.01709263	4.65788934	0.01427357
Mean MSE(STD)	0.017 (0.005)	8.794 (3.769)	0.021 (0.007)

Best results were obtained by Linear regression model with average MSE of 0.017. Almost the same result, but not signiticantly worse were obtained by Ridge method. Unfortunately Lasso method doesn't fit our example. All in all Linear regression model should be chosen as a model to our training data and predict output on test data.

1.1.3 Disclaimer

Author was enrolled to course after 3 weeks and couldn't send result through Fenix platform.

1.2 Second problem

The second problem is identical to the first one but some of the training examples (less then 10%) are not generated by the model used to generate the other data. Our goal is to detect those outliers and remove them from dataset and repeat process of 1st problem to determinate which model will fit our data best to create predictor $\hat{y}$ which will score lowest MSE on validation set.

1.2.1 Methodology

Following design-choices were made:

- Isolation Forest was applied to detect outliers train data.
- The same methodology as in first problem was applied to identify best model, 10-fold cross validation.

1.2.2 Results

Table 2: Result for each of 10 folds in cross validation using different predictors.

	Model		
Iteration	Linear regression	Lasso	Ridge
1	0.837528853	5.77614983	0.81443732
2	0.995815387	6.172215578	1.03937567
3	5.2425004	13.385241261	5.29020257
4	0.295952615	5.91540862	0.29910393
5	0.33398986	8.18993576	0.31343205
6	1.13313412	7.00228021	1.08452616
7	0.31788659	5.372887647	0.30019685
8	1.72083196	16.09124997	1.81224302
9	0.317644435	2.1160592	0.31241376
10	12.39179113	12.78202716	12.25884148
Mean MSE(STD)	2.359 (3.630)	8.280 (4.144))	2.352 (3.599)

Similar situation to previous problem best result were obtained for Ridge and Linear regression model. Lasso once again doesn't fit our problem.

1.2.3 Disclaimer

Unfortunately during second part of regression author was careless when reading instructions and read that 10% of data is outliers not less than 10% and didn't check different values of contamination. Futher inspection shows that there are 2 outliers not 10, because there is a big jump in MSE when chaning value from 0.02 to 0.03. That's the reason behind of bad MSE in validation data and bad score in leaderboard (MSE for test data = 0.448098223). Removing only 2 was correct conclusion after futher investigation and would return more accurate model.

Table 3: Mean of MSE for different contamination

	Model		
Contamination	Linear regression	Lasso	Ridge
0.01	1.957	7.565	1.943
0.02	2.071	7.534	2.054
0.03	2.265	7.811	2.246
0.04	2.161	7.773	2.142
0.05	2.173	8.086	2.157

2 Classification

For classification a dataset of face images was provided. This dataset is an adaptation of the UTKFace dataset. Provided traning set contains grayscale images of approximately 7300 subjects, with 50x50 pixels. Two different classification tasks will be performed using these data. For each different classifier will be compared but only one chosen to evaluate validation set.

2.1 First problem

The first classification task is a binary one where we wish to create a model that predicts the gender of each subject. For this task the label is either 0 (male) or 1 (female).

2.1.1 Methodology

We will train two classifiers: Multi-layer Perceptron and Convolutional Neural Network. The train data will be split on train and validation set. In order to compare result we will calculate balanced accuracy of the predictions.

Following design-choices were made during experiment:

- Simple multi-layer Perceptron with 2 hidden layers, 128 and 256 nodes each.
- Vectors provided in traning set, where reshaped from shape $(2500, 1)$ to $(50, 50)$ in order to make data as suitable input for CNN, which compares.
- In order to increase the data set, data augmentation was applied, simple horizontal flip increases the number of input examples.
- CNN architecture based on small Xception network.
- Receptive fields in each convolution were assigned as (3,3).
- Activation function in convolution layers is ReLU.
- As a result of a binary classification problem in pooling layers "sigmoid" was chosen as a activation function.
- *Keras* and *scikit-learn* Python libraries were used to create MLP and CNN architecture.
- Final score of predicting a validaton set by model is evaluated by balanced accuracy. Callbacks on each epoch is saved so we can chose best model that does not overfit the data.

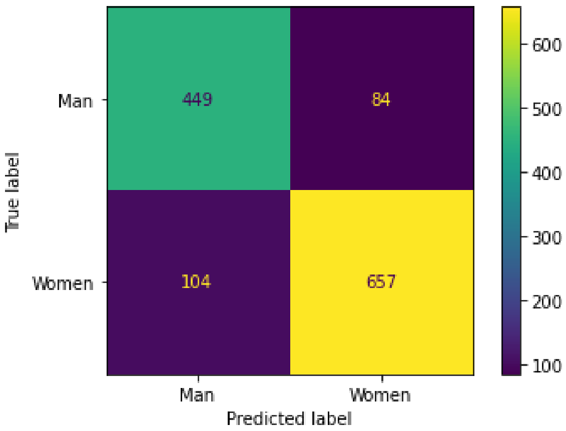
Score of the predictors will be measured by **balanced accuracy**, which is calculated by following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

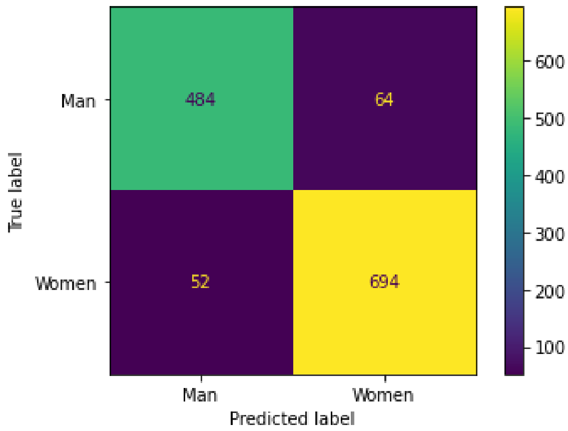
$$\text{Balanced accuracy} = \frac{\text{Precision} + \text{Recall}}{2}$$

2.1.2 Results

Figure 1: Confusion matrix for both classifiers



(a) MLP

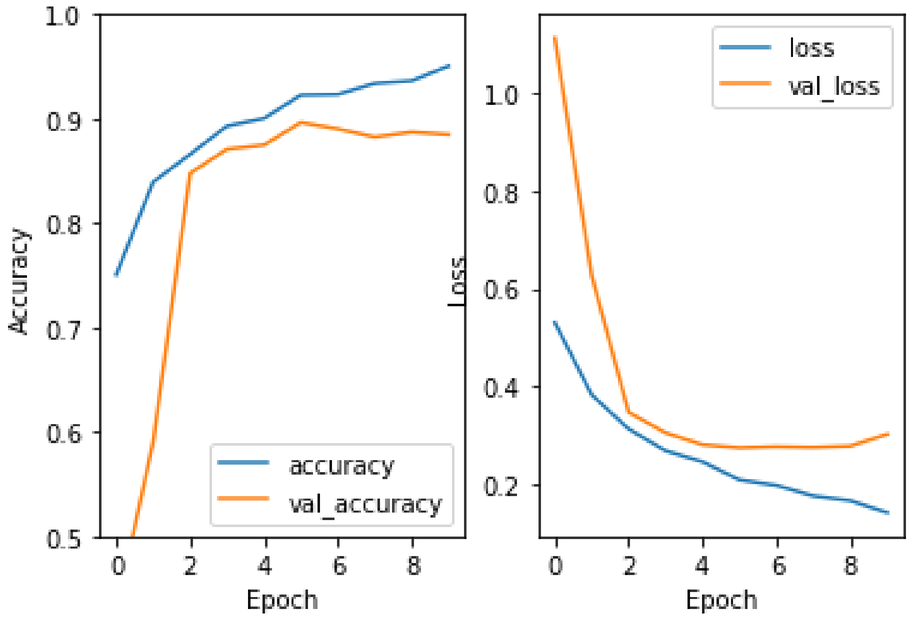


(b) CNN

Table 4: Balanced accuracy for each of classifiers

Model	BACC
MLP	0.8528696072364544
CNN	0.9067532924991684

Figure 2: Accuracy and loss for each epoch



As we can see on Figure 1 and Table 4 better result were obtained by aplying CNN, we got 5% better bACC than in MLP. In Figure 2 we can see than val_accuracy and val_loss was increasing till 5th epoch. After this point quality of classifier started decreasing. As a result this model was chosen as a final candidate to perform prediction on test data. Obtained bACC value in leader board 0.88, which almost correspond calculated value by us.

2.2 Second problem

The second classification task is a multiclass problem in which we wish to identify a person's ethnicity. For this task the label is an integer from 0 to 3, denoting Caucasian, African, Asian or Indian.

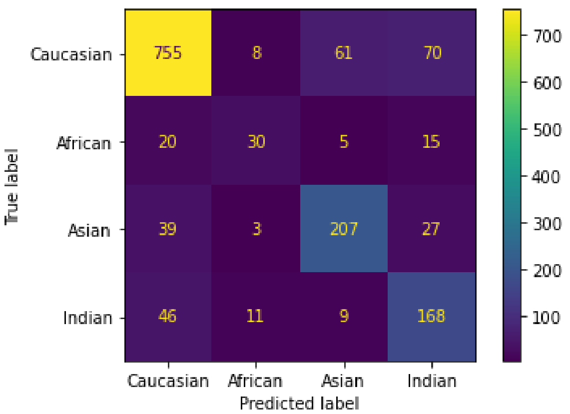
2.2.1 Methodology

Following design-choices were made:

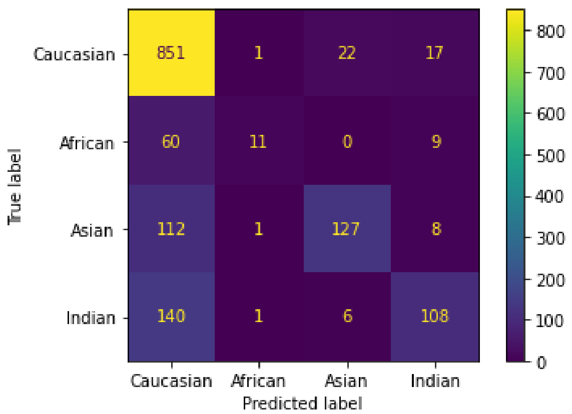
- In this case 3 different models will be compared.
- The same Multi-layer Perceptron with 2 layers described in first classification problem.
- Also Convolutional neural network with the same architecture Xception network was used.
- The only difference in CNN from previous problem was: using "softman" instead of "sigmoid", using categorical accuracy and loss function during fitness of model.
- New model tested. Random forest, which uses decision trees during training. Simple experiment how decision trees perform in comparion to MLP and CNN with image classification tasks. No additional tuning was perfoermed in this tasks.

2.2.2 Results

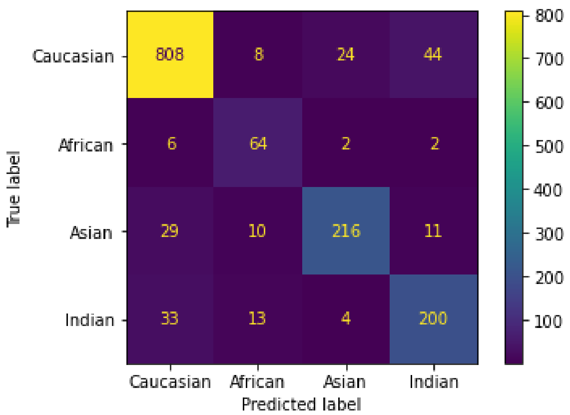
Figure 3: Confusion matrix for each of classifiers



(a) MLP



(b) Random Forest



(c) CNN

Figure 4: Categorical accuracy and categorical loss for each epoch

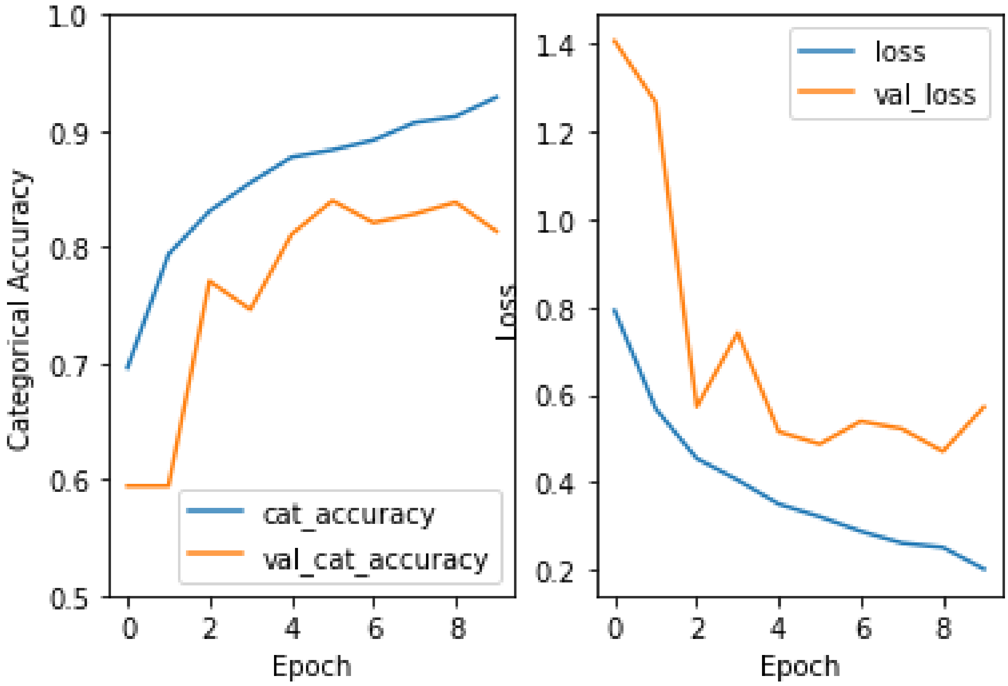


Table 5: Balanced accuracy for each of classifiers

Model	BACC
MLP	0.6852597905450255
Random Forest	0.5070582019328856
CNN	0.8407091419989934

As we can see in Figure 3 and Table 5 Random forest almost completely failed obtaining only 0.50 balanced accuracy. We can conclude that decision model are not the best in image recognition problems, at least in tested in experiment form. MLP was again significantly worse than CNN obtaining 16% worse balanced accuracy. Once again accuracy and loss was calculated at each epoch. This time using categorical metric. Validation categorical accuracy and validation categorical loss obtained maximum at 5 epoch. Once more 5th epoch was chosen as a final classifier for test data. Calculated bACC for validation set is a bit different than for test data. Validation set obtaining 0.84 and for final test data provided by professor 0.75.