

Issued: May 4, 2018
Due: June 8, 2018

Project

Policy

The project report that a team hands-in must be of team's own work. Anyone suspected of copying others will be penalized. The project will take considerable amount of time so start early.

Overview

This course has introduced a wide range of classification algorithms. This project will provide students the opportunity to design a classification algorithm using real-world data, which is computationally efficient and highly accurate. Students are required to construct a classifier of his/her choice (from SVM, kernel machine, perceptron, FDA, LDA, Naive Bayes, Logistic Regression, Neural Networks) to recognize individual digits from an image with a house number (SVHN dataset). Examples of house number images are shown in Fig.1. For training and testing the classifier, a set of 128 dimensional features extracted using a well trained algorithm- capable of obtaining 84.0% accuracy- is provided: 73257 digits in the images are used for training while 26032 digits are used for testing. The number of training samples in each class range from 500 to 4800. The classifier should be implemented in matlab with minimum use of toolbox functions. For evaluation, 4,209 test data is provided without labels, and test data prediction based on students' algorithm is evaluated using the Kaggle board. The overall framework of the project is summarized as follows.

Framework

1. For training the classifier, dataset comprising of 19,483 128-dimensional feature and class label $(\mathbf{x}_i, y_i) \in \mathbb{R}^{128} \times \{1, 2, \dots, 10\}$ is provided. The dataset is divided into 18,600 training samples `{train_feat.csv, train_label.csv}` and 833 validation samples `{valid_feat.csv, valid_label.csv}`.
2. Construct a classifier among SVM, kernel machine, Perceptron, Naive Bayes, Logistic Regression, FDA, LDA, Neural Networks. Report all toolbox functions used in implementing the classifier. Students will be penalized for using critical toolbox functions. Try to implement with only built-in matlab functions.
3. Provide means for taking care of the imbalance problem. There is considerable imbalance in the training set `{train_feat.csv, train_label.csv}`. Certain class labels have many more training data than others.
4. Use the validation set `{valid_feat.csv, valid_label.csv}` for model selection.



Figure 1: The Street View House Numbers (SVHN) Dataset.

5. For testing the classifier, input test features `{test_feat.csv}` are provided. Student's prediction performance is evaluated on the Kaggle board described below. Student's relative performance to other fellow students is made public on the Kaggle board described below.
6. To help students with their implementation, MATLAB skeleton code is provided.
 - (i) `main.m`: main function of the skeleton code (1) to load training, validation and test set, (2) to train the classifier and (3) to perform the evaluation.
 - (ii) `x_train, y_train, x_valid, y_valid = createDataset(train_feat.csv, train_label.csv, valid_feat.csv, valid_label.csv)` loads both the training and validation dataset.
 - (iii) `model = algorithm(x_train, y_train)` trains the proposed classifier with the loaded training set and outputs the trained model (classifier). Students must implement their classifier using this function.
 - (iv) `[valid_p] = validation(model, x_valid)` evaluates the trained model on the validation set and outputs the predicted labels.
 - (v) `[x_test, y_test] = createDatasetTest(test_feat.csv, test_label.csv)` loads the test set for evaluating the trained model.
 - (vi) `[test_p] = validation(model, x_test)` takes as input the trained model and test features, and the predicted results are saved in `test_p` in csv file format. This file is evaluate by the Kaggle board.

Evaluation

Kaggle board is used to evaluate the performance of the trained model on the given test set `{test_feat.csv, test_label.csv}`. Classifier is evaluated based on the following criteria:

1. Rank based on accuracy [100pt].

Output Class	1	88.5% 115	0.0% 0	4.6% 7	3.3% 4	1.0% 1	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
	2	0.8% 1	31.2% 132	0.0% 0	0.8% 1	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
	3	6.2% 8	2.6% 11	78.9% 120	17.4% 21	5.8% 6	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
	4	3.8% 5	5.9% 25	15.8% 24	76.9% 93	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
	5	0.8% 1	5.9% 25	0.7% 1	1.7% 2	93.3% 97	0.0% 0	0.0% 0	0.0% 0	0.0% 0	0.0% 0
	6	0.0% 0	11.6% 49	0.0% 0	0.0% 0	0.0% 0	41.1% 99	20.4% 44	15.5% 30	21.6% 49	18.7% 36
	7	0.0% 0	11.3% 48	0.0% 0	0.0% 0	0.0% 0	18.7% 45	33.3% 72	13.5% 26	21.6% 49	23.8% 46
	8	0.0% 0	10.6% 45	0.0% 0	0.0% 0	0.0% 0	16.2% 39	18.1% 39	48.7% 94	15.0% 34	14.5% 28
	9	0.0% 0	12.5% 53	0.0% 0	0.0% 0	0.0% 0	11.6% 28	14.4% 31	8.3% 16	30.0% 68	18.7% 36
	10	0.0% 0	8.3% 35	0.0% 0	0.0% 0	0.0% 0	12.4% 30	13.9% 30	14.0% 27	11.9% 27	24.4% 47
		1	2	3	4	5	6	7	8	9	10
Target Class											

Figure 2: confusion matrix as an example.

2. Inference time measured by staff [100pt].
3. Quality of your report [100pt].

Report

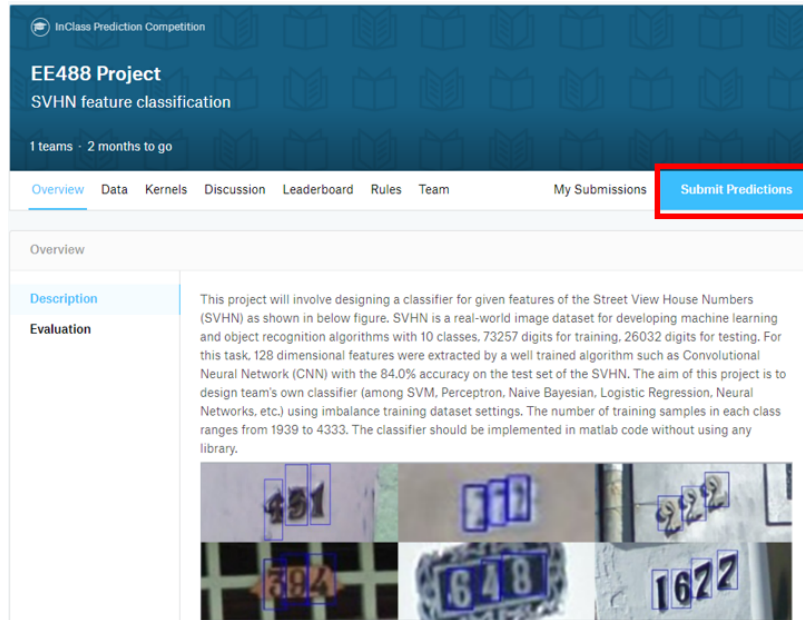
Train your classifier using the 18,600 training data described above. Provide confusion matrix of 833 validation data. Plot loss versus iteration during training. Final project report must be in pdf format [100pt]. (You may use toolbox functions to analyze your experimental results).

1. Describe your classification algorithm in terms of pseudo code. [10pt].
2. Describe key points of the algorithm to avoid over/under fitting. [10pt]
3. Describe techniques to avoid data imbalance. [10pt]
4. Describe techniques and cautions taken for improving generalization. [10pt]
5. Describe learning algorithm. [10pt]
6. Provide training / validation loss of classifier. Plot loss versus iteration. [15pt].
7. Provide confusion matrix using validation data. [10pt].
8. Describe key features of your algorithm to reduce computational cost [25pt].

Kaggle

The following steps describe the procedures for evaluating the predictions on test data and for comparing your performances to others.

1. Step1: go to the following link and sign-up : EE488 Project.
2. Step2: click submit prediction to join the competition (this step is necessary for joining an existing team).



Step 2: join the 488 competition.

3. Step3: go to team tab and change/set your team name which must be in EE488-#your team number(ex. EE488.1000) Student team number can be found from the link:team member.
4. Step 4: invite team members (remember that a team consists of less than 3 members). go to invite others and click request merge.
5. Step 5: go to data tab and download data.
6. Step 6: go to submit prediction and upload the prediction file in csv format.
7. Step 7: after submitting the appropriate csv file, rank of submission is shown in the leaderboard tab.

Submit Instructions

Please submit in .zip file to KLMS. Name of the submitted file must be “ee488_project_team_name#.zip” , for example, “ee488_project_TEAM.zip”. This file should contain two folder and one document file for results and explanation. In the matlab code, the comments explaining your code **must be** included, or points will be deducted regardless of how the code ran. Please include all files that are required for running the code in a zip file.

InClass Prediction Competition

EE488 Project

SVHN feature classification

1 teams · 2 months to go

Overview Data Kernels Discussion Leaderboard Rules **Team** My Submissions Submit Predictions

✓ You have successfully updated your team name.



Manage Team

Team Name

EE488_1000 [Save Team Name](#)

This name will appear on your team's leaderboard position.

Team Members

	 EE488 (you)	Leader
---	---	--------

Step 3: change your team name

EE488 Project

SVHN feature classification

1 teams · 2 months to go

Overview Data Kernels Discussion Leaderboard Rules **Team** My Submissions Submit Predictions



Manage Team

Team Name


EE488_1000 [Save Team Name](#)

This name will appear on your team's leaderboard position.

Team Members

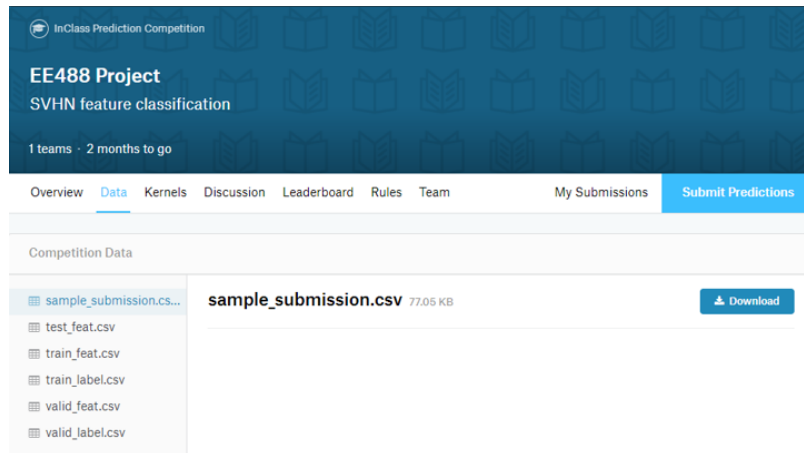
	 EE488 (you)	Leader
---	---	--------

Invite Others

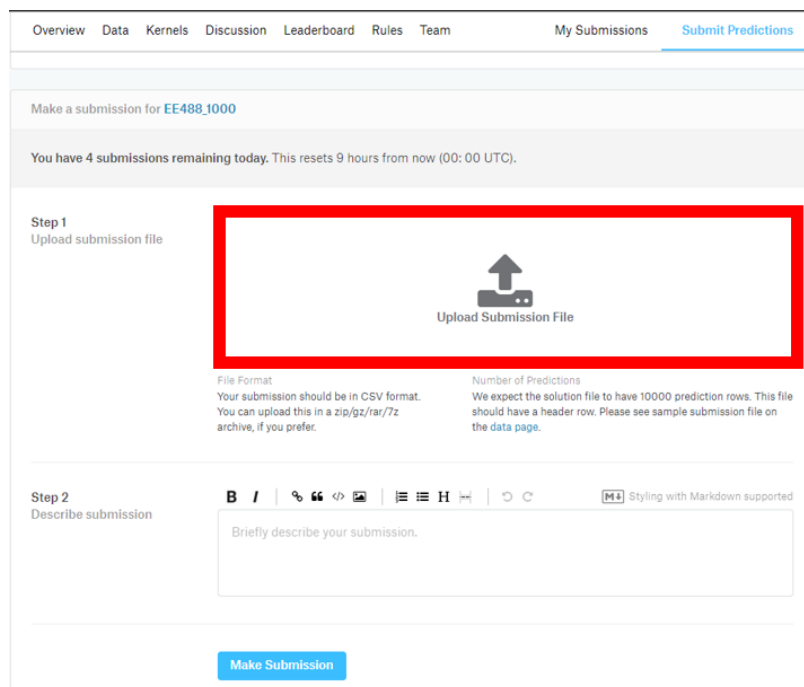
 Merge with other teams or invite users to your team by their team name

[Request Merge](#)

Step 4: invitation of your team member.



Step 5: downloading data.



Step 6: evaluation of your prediction.


OverviewDataKernelsDiscussionLeaderboardRulesTeamMy SubmissionsSubmit Predictions

Make a submission for [EE488_1000](#)

You have 4 submissions remaining today. This resets 9 hours from now (00:00 UTC).

Step 1

Upload submission file



Upload Submission File

File Format











Your submission should be in CSV format. You can upload this in a zip/gz/rar/7z archive, if you prefer.

Number of Predictions

We expect the solution file to have 10000 prediction rows. This file should have a header row. Please see sample submission file on the [data page](#).

Step 2

Describe submission

B**I**

Styling with Markdown supported

Briefly describe your submission.

Make Submission

Step 7: check submission.

7