# EECS461/ECE523
# MACHINE LEARNING
# Fall 2018

# ASSIGNMENT 2

***Due Date:*** *Sunday, November 25th, 2018, 23:59*

***Assignment Submission:*** *Turn in your assignment by the due date through LMS. Prepare a single Jupyter Notebook (.ipynb) with the answers to all questions.* ***Name the file as <your first name>_<your last name>_ assignment2.ipynb.*** *Make sure to* ***use the sample Jupyter Notebook file provided to you as template****.*

***All work in questions must be your own; you must neither copy from nor provide assistance to anybody else****. If you need guidance for any question, talk to the instructor or TA in office hours. You can also reach to TA Hacer Tilbeç at hacertilbec@std.sehir.edu.tr*

***Late Assignment Policy:*** You have a total of ***4 days of late assignment*** turn-in allowance throughout this semester. For a single assignment, you can use ***a maximum of 2 late-days.*** You decide which assignments you are going to use your 4 late-days. After assignment due date/time, each 24-hours period is counted as one late date (i.e., if you submit late 1 hour or 23 hours, you use 1 late-date). It is your responsibility to keep track of your late days. If you are late more than 2 days for any assignment or you exhausted your late days, you get 0 from the late assignment (No exceptions)

**DATA SET**

In this assignment, you are going to perform some basic image recognition tasks. The image dataset we have provided you contains images (in jpeg format) of dimensions 120x128, with the training set consisting of 315 images and the test set consisting of 90 images.

In each image, the subject has the following characteristics:

- Name – name of the subject
- Direction Faced – left, right, straight, up
- Emotion – happy, sad, neutral, angry
- Eyewear – open, sunglasses

Each image follows the naming convention "name_directionFaced_emotion_eyewear.jpg"

**DATA EXPLORATION & PREPARATION (15 points)**

Create X_train using images in the *TrainingSet* folder. You will use Pillow library to open jpeg files. If Pillow is not installed, install it by `pip install Pillow` .

```
from PIL import Image
image_array = np.array(Image.open('sample_image.jpeg').convert('L'))
```

Each image has shape of 120x128. Flatten each image array to a vector of dimensions 1x15360. Put all all image vectors in one array called **X_train**.

Labels of the images will be extracted from the file name. Create the label vector of **y_train_direction** using the *directionFaced* field of the file names of images. For instance, if the file name is aaa_right_neutral_eyewear.jpg, then the label of the image is 'right'. Use the following dictionary to encode directions into a numerical format:

```
direction_encode = {'right': 0, 'left': 1, 'up': 2, 'straight': 3}
```

You will also use eyewear as a label in a classification model later. For instance, if the file name is aaa_right_emotion_sunglasses.jpg, then the label of the image is 'sunglasses'. Create **y_train_eyewear** according to the eyewear label. Use the following dictionary to encode eyewears into a numerical format:

```
eyewear_encode = {'open': 0, "sunglasses":1}
```

At the end, **X_train** will be a numpy array that contains 315 images of dimensions 1x15360, and **y_train_direction** and **y_train_eyewear** arrays will contain 315 encoded image labels.
Using same steps you followed for training data, create **X_test**, **y_test_direction** and **y_test_eyewear** arrays using the *TestSet* folder.

**CLASSIFICATION MODEL TO PREDICT DIRECTION FACED (40 points)**

**(a) (10 points)** In this part, you will use Support Vector Classifier (SVC) to predict the direct the subject in the image faced and will try to find the best parameters for the model. Use SVC with random_state=0 as your classifier and try to find best parameters using *5-fold grid search cross validation* (GridSearchCV with scoring= "accuracy") with X_train and y_train_direction. Use following parameters below during grid search:

| **Parameter name** | **Parameter values** |
|---|---|
| kernel | linear, rbf, poly |
| C | 0.1, 0.5, 1.0 |
| tol | 0.0001,0.001 |
| decision function shape | ovo, ovr |

Report best parameters, training accuracy and validation accuracy of the best model.

**(b) (10 points)** Predict labels for X_test by using the model with the best parameters you found above. Report test accuracy, precision and recall.
Did your model perform well on both training and test data? If no, does it underfit or overfit? Write your answer the corresponding cell in the template with a precise explanation.
(Note: For precision, recall and f1 score calculations, set average parameter to 'weighted')

**(c) (20 points)** In this part, you will plot learning curve to analyze the effect of training set size on the performance of the model. Use SVC with best parameters (you found in part a) and random_state=0 as your classifier. Train the classifier repeatedly while increasing training set size with additional 15 images at each iteration (first iteration=15 images, second iteration=30 images, third iteration=45 images,...). In each iteration, calculate the rate of misclassified samples in X_train and X_test. (Note: the rate of misclassified samples equals to 1-accuracy score). Plot the learning curve (how the rate of misclassified samples changes while training set size increase).

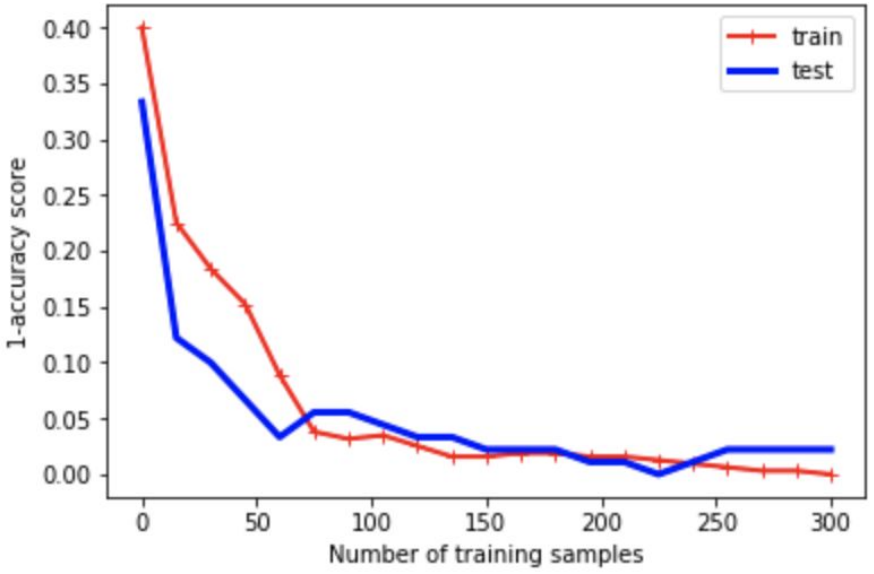


Fig 1. Number of training samples vs. Rate of misclassified samples

## CLASSIFICATION MODEL TO PREDICT EYEWEAR (25 points)

**(d) (15 points)** In this part, you will predict eyewear category. Try to find best parameters for both *SVC* (random_state=0) and *LogisticRegression* (random_state=0, solver="saga") using *5-fold grid search cross validation* (with scoring= "accuracy") with X_train and y_train_eyewear. Use following parameters below during grid search:

| SVC Parameters | | LogisticRegression Parameters | |
|---|---|---|---|
| **Parameter name** | **Parameter type** | **Parameter name** | **Parameter type** |
| kernel | linear, rbf, poly | penalty | l1, l2 |
| C | 0.1, 0.5, 1.0 | C | 0.001, 0.01, 0.1 |
| tol | 0.0001,0.001 | | |
| decision function shape | ovo, ovr | | |

Report best parameters for SVC and Logistic Regression. Also, report training accuracy and validation test accuracy of the best SVC and Logistic Regression models.

**(e) (5 points)** Predict labels for X_test using LR model with best parameters and SVC model with best parameters that you found above. Report each model's accuracy, precision, and recall scores. Which model's performance is the best?

**(f) (5 points)** Do Logistic Regression and SVC models perform well on both training and test data? If no, do they underfit or overfit? Write your answer the corresponding cell in the template with a precise explanation.

## MODEL EXPLORATION - EYEWEAR PREDICTION (20 points)

In this part, you will try to increase accuracy on X_test (Remember that you should never use test data for training and model tuning. Do the model selection/tuning with cross validation on training data). You may use different models or parameters (Notice that, so far, you tried only a small list of values for parameters such C in the grid search). You may also try to extract new features from images. Your aim is to get a higher test accuracy than the one you got in part (f).

**IMPORTANT NOTES**

- Prepare and upload one Jupyter notebook file which should be named as <your first name>_<your last name>_ assignment2.ipynb.
- A sample Jupyter notebook file provided to you. Follow the template's structure.
- **Explain your code with comments. If you don't explain your code with reasonable comments, you will get 0 from corresponding part.**

| | |
|---|---|
| Wrong file name format | -10 points |
| Not using template | -10 points |