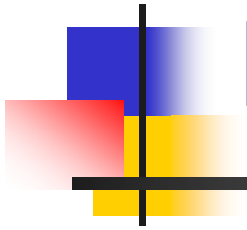
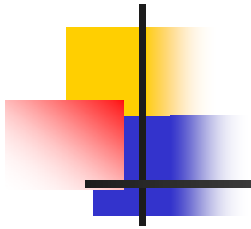


Machine Learning

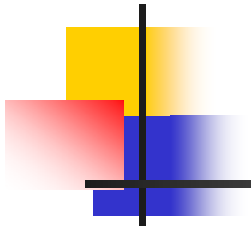
Lab 1

Fall, 2020



In-class Lab

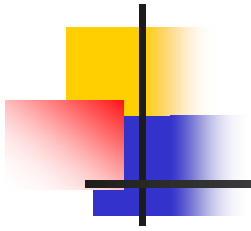
- Goal
  - Build classifier models on the same dataset
    - Choose one of two given datasets – car acceptability, heart disease
    - Build three models – random forest, logistic regression, SVM
  - Compare accuracy of the models
    - Compute the scores for various parameter values for each model
    - Compare the maximum scores of each model



Datasets (1/3)

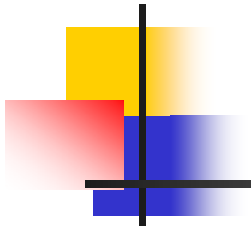
- Dataset: car acceptability
  - 1728 records in car.csv
  - https://archive.ics.uci.edu/ml/datasets/Car+Evaluation
  - https://www.kaggle.com/elikplim/car-evaluation-data-set
  - Variables
    - Set car = 'acc' for rows with car != 'unacc'
    - You need to use a label encoder for categorical attributes

Variable	Description
buying	buying price
maint	price of maintenance
doors	number of doors
persons	capacity in terms of persons to carry
lug_boot	the size of luggage boot (trunk)
safety	estimated safety of the car
car	car acceptability (target variable)



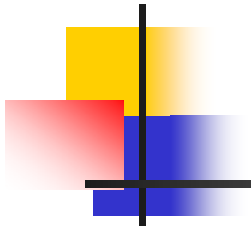
Datasets (2/3)

- Dataset: heart disease
  - 303 records in heart.csv
  - https://www.kaggle.com/ronitf/heart-disease-uci
  - Variables
    - 14 predictor variables: age, sex, cp (chest pain), chol (serum cholesterol), etc.
    - Response variable: target (1 or 0)



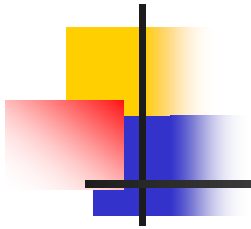
Datasets (3/3)

- Preprocessing
  - Use sklearn.preprocessing
  - Encoding
    - Categorical values need to be converted to numeric ones
    - LabelEncoder, OrdinalEncoder, OneHotEncoder, etc.
  - Scaling
    - Numeric values may need to be scaled into a given range
    - MinMaxScaler, Normalizer, StandardScaler, etc.
  - Data cleansing
    - Missing data, wrong data, outliers, etc.
    - Two given datasets have little need for cleansing
  - **TIP:** Concentrate on building classifiers in this class, although preprocessing is essential for efficient classifiers.



What to Do (1/2)

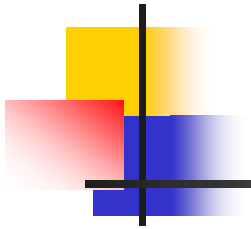
- Build three classifier models
  - Random forest, logistic regression, and SVM models
  - *(each as a binary* classifier)
- For each model,
  - Do *k*-fold cross validation (*k* = 10)
  - For each fold, split the dataset, fit & test the model, and compute the score using Scikit-learn as studied in class
  - Compute the average score



What to Do (2/2)

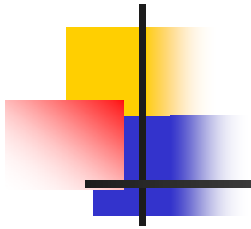
- Try various parameter values
  - For each combination of parameter values,
    - Print a confusion matrix using Seaborn (with margins)
    - Display the accuracy score in the form of 3D bar chart
      - ex) "Title": RandomForest(gini), "X": n_estimators, "Y": max_depth, "Z": accuracy
  - Find the maximum score and the corresponding parameter values for each classifier (see example below)

Classifier	Parameters
Random forest	criterion="gini"(gini index), "entropy"(information gain); n_estimators=1, 10, 100; max_depth=1, 10, 100
Logistic regression	C=0.1, 1.0, 10.0; solver="liblinear", "lbfgs", "sag"; max_iter=50, 100, 200
SVM	C=0.1, 1.0, 10.0; kernel="linear", "poly", "rbf", "sigmoid"; gamma=0.01, 0.1, 1.0, 10.0



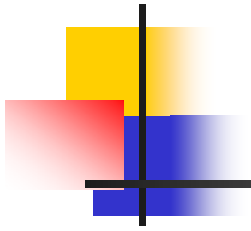
Programming Homework

- Goal: *Ensemble Learning*
  - In our class:
    - Improve classification accuracy by using three classifiers
    - random forest, logistic regression, and SVM models
- Dataset: MNIST original
  - 70,000 images of hand-written digits (0 ~ 9) → requires *multinomial* classifier
  - Widely used as a basic dataset for studying and training classifiers
  - https://www.kaggle.com/avnishnish/mnist-original



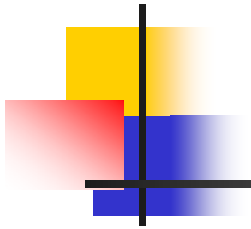
What to Code (1/2)

- Split the dataset into two sub-datasets
  - D1 (train & test), D2 (ensemble-test)
- Build three classifier models using D1
  - In the same manner as done in the Lab (page 7)
    - random forest: {criterion: “gini”, “entropy”}
    - logistic regression
    - SVM_linear
  - For each model, find the parameter combination with the highest score



What to Code (2/2)

- Test the ensemble classifier model using D2
  - For an image in D2, the classification result is decided as the digit that the majority of three classifiers predicted
  - If all classifiers have different predictions, choose the one returned by the classifier with the highest score
  - Display a confusion matrix using Seaborn (with margins), and compare with those obtained using each of the three classifiers
- Note:
  - You may use only a portion of original data for D1 and D2 (e.g., 10%)
  - Try two approaches with and without sklearn.ensemble.VotingClassifier



End of lab