University of Glasgow | School of Computing Science

# ***CSC3005 Laboratory/Tutorial 4: Classification Analysis II***

Using back tutorial/Lab 3 Question 1 dataset two-dimensional dataset that comprises of 10,000 labelled instances, each of which is assigned to one of two classes, 0 or 1. Instances from each class are generated as follows:

1. Instances from class 1 are generated from a mixture of 4 Gaussian distributions, cantered at [5,15], [15,15], and [15,5], [5, 5] with covariance of 2 with zero mean respectively.
2. Instances from class 0 are generated from a uniform distribution in a square region, whose sides have a length equals to 20.

## *1.* K Nearest Neighbour(KNN)

In KNN, the class label of a test instance is predicted based on the majority class of its $k$ closest training instances. The number of nearest neighbors, $k$, is a hyperparameter that must be provided by the user, along with the distance metric. By default, we can use Euclidean distance (which is equivalent to Minkowski distance with an exponent factor equals to r=2):

**Step 1:** Create the same data set as in Tutorial/Lab 3 Question 1

**Step 2**: Split the Data into Training and Test in the ratio of 70:30 using sklearn.model_selection.train_test_split() function as in Tutorial 3 Question 1

**Step 3**: Create KNN Classifier and test the performance for various neighbours,$k$

from $k$= [1,5,10,20,25] .

Perform the three step approach similar in Tutorial/Lab3

a) Model fit the training data from step 2
b) predict with both train and test data.
c) Compare the accuracy using sklearn.metrics.accuracy_score()

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.fit

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier.predict

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

**Step 4 :** Plot the performance of training accuracy and test accuracy versus the number of neighbour, $k$. What can you observe?

## 2. Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable $y$ and dependent feature vector $x_1$ through $x_n$ :

$$P(y|x_{1\ldots}x_n) = \frac{P(y)P(x_1 \cdots x_n|y)}{P(x_1 \cdots x_n)}$$

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^{n} P(x_i|y)$$

**Step :** Repeat the same process as in Question 1 by importing the sklearn.naive_bayes.GaussianNB() . We assume the features is gaussian distributed.

https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

What is the training and testing accuracy? Is it good or bad and why?

## 3. Linear Prediction using Logistic Regression and Linear Support Vector Machine

Linear classifiers such as logistic regression and support vector machine (SVM) constructs a linear separating hyperplane to distinguish instances from different classes.

For logistic regression, the model can be described by the following equation:

$$P(y=1|x) = \frac{1}{1+e^{-(\boldsymbol{w}^T x+b)}} = \sigma(\boldsymbol{w}^T x + b)$$

The model parameters $(w, b)$ are estimated by optimizing the following regularized negative log-likelihood function:

$$(\boldsymbol{w}^*, b) = \underset{\boldsymbol{w},b}{\operatorname{argmin}} -\sum_{i=1}^{N} y_i \log\big(\sigma(w^T x + b)\big) + (1-y_i)\log\big(1-\sigma(w^T x+b)\big) + \frac{1}{C}\Omega(\boldsymbol{w}.b)$$

where $C$ is the hyperparameter that controls the inverse of model complexity(small values mean strong regularization) while $\Omega(.)$ is the regularization term which by default is an $l_2$ norm in sklearn.

For support vector machine, the model parameters are estimated by solving the constrained optimization problem

$$(\boldsymbol{w}^*, b) = \underset{\boldsymbol{w},b,\xi_i}{\operatorname{argmin}} \frac{\|\boldsymbol{w}\|^2}{2} + \frac{1}{C}\sum_{i=1}^{N} \xi_i$$

$$s.t. \forall_i : y_i[\boldsymbol{w}^T \phi(x_i) + b] \geq 1 - \xi_i, \xi_i \geq 0$$

**Step 1:** Import linear model.LogisticRegression from sklearn and SVC from sklearn.svm

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

**Step 2:** Create both the logistic regression and svm classifier with the hyperparameter $C$ ranges from [0.01,0.1,0.2,0.5,0.8,1,5,10,20,50]

**Step 3:** Repeat the same process of fitting and predicting with training and test data as in Question 1 and 2. Hence obtain the training and testing accuracy as usual and plot both accuracies versus the hyperparameter $C$

What can you observe from the plot?

## 4. Non-Linear Support Vector Machines

Let try nonlinear support vector machine with a Gaussian radial basis function kernel to fit the 2-dimensional dataset.

**Step : Repeat the whole process in Question 3 on SVM but setting kernel='rbf' and gamma='auto'**

What can you observe from the plot?

## 5. Visualization of Decision Boundary on various Classifier

Let look at the decision boundary of all classifiers that we have done through to match the intuition understanding on the accuracy performance. Perform visualization using contourf() method from matplotlib.pyplot on the following classifiers as done above

1)KNN(K=30)
2)Navie Bayes

3)Logistic Regression (C=50)
4)Linear SVM(C=50)
5)Non Linear SVM (RBF)
6)Decision Tree (Depth =10)

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.contourf.html

What is your conclusion? What is your final insight and understanding how to choose an appropriate classifier?

## 6. Naïve Bayes Theorem

a) Suppose the fraction of CS undergraduate students who are female is 15% and the fraction of CS graduate students who are female is 23%. If one-fifth of the university CS students are graduate students and the rest are undergraduates, what is the probability that a student who are female is a graduate student?

b) Given the information in part (a), is a randomly chosen university student more likely to be a graduate or undergraduate student?

c) Repeat part (b) assuming that the student is a female.

d) Suppose 30% of the graduate students live in a hostel but only 10% of the undergraduate students live in a hostel. If a student is a female and lives in the hostel, is she more likely to be a graduate or undergraduate student? You can assume independence between students who live in a hostel and those who is female.

## 7. K Nearest Neighbor Theory

The nearest-neighbor algorithm can be extended to handle nominal attributes. A variant of the algorithm called PEBLS (Parallel Examplar-Based Learning System) by Cost and Salzberg measures the distance between two values of a nominal attribute using the modified value difference metric (MVDM). Given a pair of nominal attribute values, $V_1$ and $V_2$, the distance between them is defined as follows:

$$d(V_1, V_2) = \sum_{i=1}^{k} \left| \frac{n_{i1}}{n_1} - \frac{n_{i2}}{n_2} \right|$$

where $n_{ij}$ is the number of examples from class $i$ with attribute value $V_j$ and $n_j$ is the number of examples with attribute value $V_j$

Consider the training set for the loan classification problem as shown below. Use the MVDM measure to compute the distance between every pair of attribute values for the Home Owner and Marital Status attributes.

| | categorical | categorical | continuous | class |
|---|---|---|---|---|
| ID | Home Owner | Marital Status | Annual Income | Defaulted Borrower |
| 1 | Yes | Single | 125k | No |
| 2 | No | Married | 100k | No |
| 3 | No | Single | 70k | No |
| 4 | Yes | Married | 120k | No |
| 5 | No | Divorced | 95k | Yes |
| 6 | No | Married | 60k | No |
| 7 | Yes | Divorced | 220k | No |
| 8 | No | Single | 85k | Yes |
| 9 | No | Married | 75k | No |
| 10 | No | Single | 90k | Yes |

8. **Artificial Neural Network (ANN) theory**

a) Demonstrate how the perceptron model can be used to represent the AND and OR functions between a pair of Boolean variables.

b) Comment on the disadvantage of using linear functions as activation functions for multilayer neural networks.