Machine Learning Practice with Python- Midterm (2 pages) **2022/10/25**

(Note: Please explain your answer as much as possible to get a good grade)

1. (1) Please explain "*Machine Learning*" in your own words (5%).

   (2) Please explain and give an example for each of the following *supervised learning tasks*: *Classification, Regression* (5%, 5%).

2. (1) Please explain the purpose of the *validation set* (5%).

   (2) Please explain how to calculate the *validation error* for *Cross Validation with K*=3 (5%).

3. In order to guarantee that *the test set* is representative of the overall population, the technique of "*stratified sampling*" is usually used, in which the population is divided into homogeneous subgroups called *strata* according to the most important feature in the task, and the right number of instances are sampled from each stratum. Please given an example for stratified sampling in preparing the test set. (10%)

4. The confusion matrix for binary classification is defined as in the following illustration, in which some performance metrics are also defined.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Positive (TP)

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP}$$

$$Precision = \frac{TP}{FP + TP}$$

$$Recall = \frac{TP}{FN + TP}$$

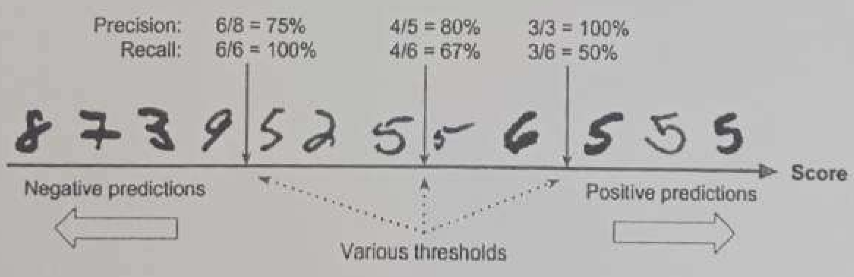
   (1) A *binary classifier* has the confusion matrix as the result, please depict the confusion matrix of a perfect classifier for the task (5%).

	N	P
N	53,057 (TN)	1,522 (FP)
P	1,325 (FN)	4,096 (TP)

   (2) Please depict the confusion matrix for a *dumb classifier* which classifies every instance in the negative class for the task (5%).

   (3) Please explain why *Accuracy* is not a good measure for comparing the classifiers in (1)&(2), and *Precision/Recall* are better for performance comparison (5%). data unbalance

   (4) Please refer to the following illustration and explain *Precision/Recall Trade-off* (i.e. increasing precision reduces recall, and vice versa) (5%).



5. (1) Please explain (with illustration) the idea behind *Gradient Decent* (5%).

   (2) Please explain (with illustration) why *feature scaling* speeds up Gradient Decent (5%).

6. Considering *Linear Regression* with *3-D input feature* (x_1, x_2, x_3) as follows:

   $\mathbf{x}^{(i)}$: the i^{th} instance, $\mathbf{x}^{(i)} = \left(x_0^{(i)}, x_1^{(i)}, x_2^{(i)}, x_3^{(i)}\right)$, $x_0^{(i)} = 1$ for the bias term

   $y^{(i)}$: the label of $\mathbf{x}^{(i)}$

   $h_\theta(\mathbf{x}^{(i)})$: the hypothesis function $= \theta \cdot \mathbf{x}^{(i)} = \theta^T \mathbf{x}^{(i)}$, $\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$

   Cost Function: *Mean Squared Error* (*MSE*) $= \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(\mathbf{x}^{(i)}) - y^{(i)}\right)^2$

   (m: the total # of instances for training)

   (1) Please list the weight update step(s) in *Batch Gradient Decent* (5%).

   (2) Please revise the step(s) in (1) if *Ridge Regression* (ℓ_2 norm) is adopted for regularization (5%).

   Ref. *Ridge Regression Cost Function* = $MSE(\theta) + \alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2$, ($n = 3$ in our case)

   (3) Please explain why *feature scaling* is important when using *Ridge Regression* for regularization (5%).

7. Considering *3-D input feature* (x_1, x_2, x_3) and Linear Regression model. Please write down the hypothesis function if we add more features by the method of "*Polynomial Feature*" with *degree* =*2* (Don't forget the *bias* term) (10%).

8. Please refer to the following graph and explain (1) *Underfitting*, (2) *Overfitting*, and (3) *Bias/Variance Trade-off* (3%, 3%, 4%).