**Exam 01 Terminology**

**Types of Machine Learning Tasks**

There are three main categories of machine learning problems:

* **Supervised Learning.** The goal of a supervised learning task is to create a model that can be used to predict an output based on one or many input values. This is accomplished by providing the algorithm with many observations for which the output is known, from which it attempts to detect patterns.
* **Unsupervised Learning.** The goal of unsupervised learning is to identify structure that exists within the data but might not be perceptible to humans. Unsupervised learning can be used to cluster or group observations in an effort to identify similarities, or to detect unusual observations or outliers.
* **Reinforcement Learning.** The goal of reinforcement learning is to train an algorithm to complete a complicated task by allowing to repeat the task many times. With each new attempt, the algorithm is provided with feedback regarding its performance in the form of a score.

**General Topics Relating to Supervised Learning**

In a supervised learning task, the goal is to generate a model that can predict an output based on provided inputs.

* The output is typically called a **label**, **target value**, or **response variable**. We will generally denote the label as . We will denote predicted values of the label as  .
* The inputs are typically called **features** or **predictors**. We will typically denote the predictors as , , …, .

The algorithm is provided with a data set consisting of both features and labels. A single set of associated feature values, together with the associated label, is called an **observation** (or **instance** or **sample**). In a supervised learning task, the available collection of labelled observations is generally split into three non-overlapping sets:

* The **training set** is the set that the model or models being considered is actually trained on. An algorithm is generally provided with some type of an objective function (such as SSE or accuracy) that it attempts to either maximize or minimize on the training set.
* The **validation set** is used for model selection. This set is used to select the value of any hyper-parameters present in a particular model, or to compare models of different forms.
* The **testing set** is used after training and model selection have been completed. It is used once to compute a score for the final model. Since this data set was not seen during training or model selection, the score calculated on the training set provides a measure of how well the final model will generalize to unseen data.

**Types of Supervised Learning Tasks**

Supervised learning tasks can be grouped into two types of problems, based on the form of the desired output.

* **Regression.** A regression task is one for which the target values are continuous, real numbers.
* **Classification.** A classification task is one for which the target values are discrete classes.

**Additional Topics Relating to Supervised Learning**

* **Overfitting.** An algorithm overfits if it adapts to noise in the training set, producing an overly-complex model that performs well on the training set, but that does not generalize well to new observations. The more flexible an algorithm is, the more likely it is to overfit.
* **Underfitting.** An algorithm underfits if it fails to capture significant details within the training data, producing an overly-simple model that does not perform well on the training set, and also does not generalize well. The more rigid an algorithm is, the more likely it is to underfit.
* **Regularization.** Regularization is a technique that is used to avoid overfitting. Regularization is implemented by modifying the algorithm to make it less flexible. The details of how this is done vary from one algorithm to another.
* **Normalization.** Some ML algorithms require that features be measured in similar scales. Normalization is the process of rescaling features in order to meet this requirement. A typical normalization method is **min-max scaling**. In min-max scaling, a feature is rescaled to a scaled feature using the formula . The values of the new scaled feature will fall within the range .

**Selected Regression Algorithms**

* **Linear Regression.** In a linear regression model, predictions for the output are generated by a function with the form , were the parameters are selected to minimize the sum of squared errors objective function, which is given by , where  .

Equivalent to minimizing , we could require that the value is maximized. This value is defined by , where . The value can be interpreted as being the proportion of the variation in the label that is explained by the presence of the features in the model.

* **Lasso Regression.** Lasso regression is a modification of linear regression that performs regularization by adding a penalty for having large coefficients. The cost function to be minimized in lasso regression is given by the formula . The constant is a hyperparameter that is selected through validation. It determines the amount of weight to put on the penalty term. For sufficiently high values of , several of the model coefficients will be set to 0. As such, lasso regression also performs a type of automatic variable selection. Lasso regression generally benefits from feature normalization.
* **K-Nearest Neighbors Regression.** When predicting the output for a new observation using KNN, we first identify the training observations that are nearest to the new observation in terms of their feature values. We then average (take an arithmetic mean) of the values for these training observations. This average is used for the prediction **.** The constant is a hyper-parameter that is selected through validation by selecting the that results in the highest (or equivalently, the lowest SSE) on the validation set. Since this is a distance-based model, it is important to perform feature normalization.

**Selected Classification Models**

* **Logistic Regression.** Logistic regression is a binary classification method that results in a linear decision boundary. The model has the form . Given a labeled observation, we let   if and otherwise. The algorithm selects the model parameters by maximizing the likelihood score, . For technical reasons relating to rounding error, it is generally more feasible to maximizing the log-likelihood score instead: .
* **Multiclass Logistic Regression.** Logistic regression can be used to perform multiclass classification as follows. Assume that your dataset has possible classes. We create separate binary classification models: , , …, . An individual model is intended to make predictions of the form “Is in Class *j*” or “Is not in Class *j*”. To classify a new observation, we use each of the *k* models to generate probabilities   of the form . We then classify the observation as the class with the highest probability .
* **K-Nearest Neighbors.** When predicting the output for a new observation using KNN, we first identify the training observations that are nearest to the new observation in terms of their feature values. We classify the new observation as the class that appears most frequently in these training observations. If there is a tie, we use the total distance between the new observations and the training observations in each class as the tie breaker. The constant is a hyper-parameter that is selected through validation by selecting the that results in the greatest accuracy (or in some cases, precision or recall) on the validation set. Since this is a distance-based model, it is important to perform feature normalization.

**Precision, Recall, and Accuracy**

* **Accuracy.** A classification model’s accuracy on a labelled data set is the proportion of observations in the set for which it correctly predicts the label. Accuracy is a common metric used to score classification models, but it does have drawbacks. It is not a particularly good metric to use on a dataset with unbalanced classes, for example.
* **Precision.** A model’s precision on a labelled data set, with respect to a particular class, is equal to the number of observations **correctly** predicted to be in that class, divided by the **total** number of observations that were **predicted** to be in that class. Intuitively, it measures how likely it is that the model is correct when it predicts that an observation belongs to this particular class.
* **Recall.** A model’s recall on a labelled data set, with respect to a particular class, is equal to the number of observations **correctly** predicted to be in that class, divided by the **total** number of observations that were **actually** in that class. Intuitively, it measures how likely the model is to correctly identify instances of this particular class.