Task 1: Comparing Machine Learning Algorithms

You run a large company that receives tens of thousands of job applications each year for positions in your company. You have a database of the hundreds of thousands of applications that you have received over the past decade as well as information about who, from this pool, you have phone interviewed, interviewed in person, and hired. For those people that you hired, you have annual performance evaluation data as well. You are interested in a machine learning algorithm to determine, from the tens of thousands of applications that you receive each year, which small subset you should interview. What machine learning algorithm might be appropriate for this task (circle all that apply)?

- Decision tree
- \( k \) nearest neighbors
- Linear regression
- Logistic regression

For what algorithms is it important to use feature scaling (circle all that apply)?

- Decision tree
- \( k \) nearest neighbors
- Linear regression
- Logistic regression
Suppose we have data, such as in the two examples illustrated below, that we have reason to believe is not linearly separable.

Assuming we do not change or add features, which algorithms might be appropriate to use to classify such data?

- Decision tree
- $k$ nearest neighbors
- Perceptron
- Logistic regression

Suppose we have data, such as in the two examples illustrated below, that we have reason to believe is not linearly separable.

Assuming we can change and add features, which algorithms might be appropriate to use to classify such data?

- Decision tree
- $k$ nearest neighbors
- Perceptron
- Logistic regression
Task 2: Logistic Regression

For gradient descent, on the surface the gradient for logistic regression looks the same as that for linear regression:

\[ w_j = w_j - \alpha \frac{1}{n} \sum_{i=1}^{n} (h(x^{(i)}) - y^{(i)}) x_j^{(i)} \]

However, the gradient calculation for logistic regression is different from that of the gradient calculation for linear regression. Why?

Suppose you want to classify data that you believe is not linearly separable and you wish to explore using non-linear logistic regression for this purpose. You experiment with adding features to your data that are polynomial combinations of your original data. You add features that correspond to increasingly complex polynomials, i.e., polynomials of increasing degree. The degree of the polynomial may be viewed as a hyperparameter, so you can evaluate polynomials of different degree using validation data. Which of the following figures represents a reasonable expectation for how the cost associated with training data and with validation data might vary as the degree of the polynomial increases?
Task 3: Confusion Matrix

Thus far, when evaluating the performance of a classifier, our measure of accuracy has been the percentage of classifications that the classifier makes correctly, i.e., if it classifies 35 out of 50 data points correctly then we say that the accuracy is 70%. This notion of accuracy may be reasonable when we have approximately the same number of examples in each class, but it can break down when we have skewed classes. For example, imagine we are classify dollar bills as authentic (0) or counterfeit (1). Suppose we encounter many more authentic bills than counterfeit bills, e.g., 99% of bills are authentic and 1% are counterfeit. If we used a naïve algorithm for classification that always classified every bill as authentic, then it would have an accuracy of 99% because it would correctly classify all authentic bills (which account for 99% of all bills) even though it would incorrectly classify all counterfeit bills (which account for 1% of all bills). Such a naïve classifier achieves a high accuracy, 99%, even though it is not particularly useful. This is one motivation for using other measures (besides accuracy) of a classifier’s performance, particularly when the data is skewed.

A confusion matrix uses a table format to keep track of different aspects of a classifier’s performance. For instance, in the binary classification problem of identifying authentic and counterfeit bills, the confusion matrix would keep track of:

- True Positives (TP) - the number of bills predicted to be counterfeit that are actually counterfeit
- False Positives (FP) - the number of bills predicted to be counterfeit that are actually not counterfeit
- True Negatives (TN) - the number of bills predicted not to be counterfeit that are actually not counterfeit
- False Negatives (FN) - the number of bills predicted not to be counterfeit that are actually counterfeit

Confusion matrix

<table>
<thead>
<tr>
<th>Predicted by classifier to be counterfeit</th>
<th>Actually Counterfeit</th>
<th>Actually Not Counterfeit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted by classifier not to be counterfeit</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td></td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

Suppose for one thousand bills, a classifier predicts 30 to be counterfeit and 970 to be authentic. Of the 30 counterfeit predictions, 20 are actually counterfeit. Of the 970 authentic predictions, 965 are actually authentic. Fill in a confusion matrix below indicating the number of true positive, false positive, false negative, and true negative classifications.
Two common measures of a classifier’s performance are \textit{recall} (a.k.a., \textit{sensitivity}) and \textit{precision}. \textit{Recall} is the percentage of actual positives (counterfeit bills in our example) that were correctly predicted to be positives (counterfeit). \textit{Precision} is the percentage of predicted positives (predicted counterfeit bills) that are actual positives (counterfeit bills).

\begin{align*}
\text{recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\
\text{precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}
\end{align*}

In our example, high recall means that we are identifying a lot of the actual counterfeit bills. High precision means that when we predict a bill as counterfeit, our prediction is likely correct.

What is the recall and what is the precision associated with our counterfeit bill classifier?

It is desirable for a classifier to have high recall and high precision. However, there is often a trade-off between the two, where a classifier’s recall can be improved at a cost to its precision, or the classifier’s precision can be improved at a cost to its recall. For a given classifier, if you tweak its parameters so as to increase the number of positive predictions that the classifier makes (in our example, you cause the classifier to predict a larger number of bills as counterfeit), then which of the following are likely:

- Recall will increase and precision will decrease
- Precision will increase and recall will decrease

It is often helpful to have a single measure, rather than two measures (recall and precision), to evaluate a classifier. The $F_1$ score is a measure of a classifier’s performance that combines recall and precision into one number. The $F_1$ score is defined as follows:

\[ F_1 \text{ score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \]

What is the $F_1$ score for our counterfeit bill classifier above?

Instead, suppose we use a naïve classifier that predicts all one thousand bills are counterfeit even though only 30 are actually counterfeit. What is the $F_1$ score for this naïve classifier?
Your favorite college is considering a change to its admission process. Rather than having the Admission Office staff read the thousands of applications from aspiring students that the college receives each year, a machine learning algorithm will use the applications as input to determine who will be offered admittance to the fine institution. Opponents of this admission process change argue that machines cannot adequately evaluate the diverse qualities and contributions valued by the college for both individual applicants and the entire class of applicants in aggregate. Proponents of this admission process change argue that it will provide a fairer and more transparent system without the opaque biases, conscious and unconscious, associated with different people reading different subsets of applications, and that it will dramatically reduce costs thereby enabling the college to charge lower tuition and offer greater financial aid packages.

Do you support changing the admission process to use machine learning algorithms to determine admittance? What aspects of the change are you comfortable with and uncomfortable with? Are there admission practices/policies/properties that you would insist on? (minimum 100 word response)
Name(s): ____________________________________________

In the TIME column, please estimate the time you spent on this exercise. Please try to be as accurate as possible; this information will help us to design future exercises.

<table>
<thead>
<tr>
<th>PART</th>
<th>TIME</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you or your partner is using a late coupon, please indicate who is using the coupon and how many coupons.
Late coupons: ____________________________________________