# Lecture notes on Udacity's "Introduction to Machine Learning Class"

Lauren E. Blake

June 3, 2017

*The class contains 16 lessons as well as a Final Project.*

*My code for the exercises and projects can be found on my GitHub Repo: https://github.com/Lauren-Blake/Udacity_Machine_Learning.*

## 0.1 Lesson 1: Welcome

### 0.1.1 Introduction I

- Lots of applications for machine learning across diverse fields!

### 0.1.2 Introduction II

- Keep your eyes out for applications of machine learning and data sets that you could use machine learning on.

### 0.1.3 Introduction III

- Format: Lectures with quizzes, mini-projects at the end of each lesson.
- Final project at the end that ties together different aspects of the mini-projects.

## 0.2 Lesson 2: Naive Bayes

### 0.2.1 ML in the Google Self-Driving Car

- Train the self-driving car by showing car how to drive and giving the computer examples of humans driving. This is an example of a supervised classification problem.

### 0.2.2 Acerous versus non-Acerous Quiz

- Gave example of acerous animals and non-acerous examples. Then asked whether we thought a horse was acerous or not.
- Answer: A horse is acerous (lacking horns or antlers).

### 0.2.3 Supervised Classification Examples Quiz

- Need to have training data and then making predictions, recommendations, etc. based on the training set.

### 0.2.4 Features and Labels Musical Example Quiz

- With a song, features could be tempo, intensity, gender of person singing it, etc. Labels are whether a person likes the song or not.

### 0.2.5 Features Visualization Quiz

- Answer: She likes those because they are close to the other data points representing songs that the person also likes.

### 0.2.6 Classification by Eye Quiz

- Answer: Unclear because the new data point is close to two different labels.

### 0.2.7 Intro To Stanley Terrain Classification

- Features: speed and ruggedness

### 0.2.8 Speed Scatterplot: Grade and Bumpiness Quiz

- Answer: From the picture, we can see that the terrain looks flat and smooth, particularly relative to the other pictures.

### 0.2.9 Speed Scatterplot 2

- Answer: From the picture, we can see that the terrain looks very steep and medium bumpy.

### 0.2.10 Speed Scatterplot 3

- Answer: From the picture, we can see that the terrain looks flat and very.

### 0.2.11 From Scatterplots to Predictions

- Answer: The points look closer to the blue circles than the red X's.

### 0.2.12 From Scatterplots to Predictions 2

- Answer: Unclear because the points look equally close to the blue circles than the red X's.

### 0.2.13 Scatterplots to Decision Surface Quiz

- A decision surface parses out the training data into different features so that a data point falling on one side of the decision surface has one label and on the other side, a different label.

- Answer: Red cross because it is on the same side as the training data with red crosses.

### 0.2.14 A Good Linear Decision Surface

- When the decision surface is a straight line, it is a "linear" decision surface.

- Answer: Select the line that clearly and consistently separates the red crosses from the blue circles.

### 0.2.15 Transition into Using Naive Bayes

- Naive Bayes is a common algorithm to find the decision surface.

### 0.2.16 NB Decision Boundary in Python

- Have 750 training data points and make a decision boundary.

### 0.2.17 Getting Started with sklearn

- Documentation on Naive Bayes with derivation and code
- clf = GaussianNB() # Create Gaussian classifier
- clf.fit(features, labels) # Fit Gaussian classifier
- clf.predict #Give it a point and get out a label

### 0.2.18 Gaussian NB Example

- Goes through each line of code in the example section of http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

### 0.2.19 GaussianNB Deployment on Terrain Data

- Answer: Add the following code:

```
#Specify classifier type
clf = GaussianNB()


# Fit the decision boundary
clf.fit(features_train, labels_train)
```

### 0.2.20 Finding Naive Bayes Accuracy

- accuracy = number of points classified correctly / all points in the test set
- Answer: See Lesson_2_Section_20_Quiz:Calculating_NB_Accuracy on my Udacity Machine Learning GitHub repo.

### 0.2.21 Training and Testing Data

- Important to train and test on different data sets (need to generalize to new data sets)

### 0.2.22 Unpacking NB Using Bayes Rule

- What is Naive Bayes?

### 0.2.23 Bayes Rule

### 0.2.24 Cancer Test

- Answer: We can see in from the diagram that the probability that someone with a positive cancer test actually has the disease is approximately 8%.

### 0.2.25 Prior and Posterior

- Answers: 0.009 and 0.099

### 0.2.26 Normalizing 1

- Answer: The normalizing constant is 0.009+0.099 = 0.108.

### 0.2.27 Normalizing 2

- Answer: 0.08333. Divide the cancer joint by the normalizing constant to get the posterior.

### 0.2.28 Normalizing 3

- Answer: 0.916666. Divide the non-cancer joint by the normalizing constant to get the posterior.

### 0.2.29 Total Probability Quiz

- Answer: 1. Add the answers from Normalizing 2 and Normalizing 3 together.

### 0.2.30 Bayes Rule Diagram

$$Totalprobability = 1 = P(\frac{C \mid Pos}{Normalizingconstant}) + P(\frac{non - C \mid Pos}{Normalizingconstant}) \tag{1}$$

### 0.2.31 Bayes Rule for Classification

- Who is the person that is sending the email based on the words that they used?

### 0.2.32 Chris or Sara Quiz

- Answer 1: Sara because the email contains words that she uses with higher probability than Chris.
- Answer 2: Chris because the email contains words that she uses with higher probability than Sara.

### 0.2.33 Posterior probabilities

- Answers: 0.5714; 1-0.5714; First find

$$Prob\frac{(Chris \mid Emailcontainsthewords\text{“}lifedeal\text{”})}{Constant} \tag{2}$$

  Then, find its complement.

### 0.2.34 Bayesian probabilities on your own

- Prob (Chris | Email contains the words “love deal” ) =

  Prob (“love deal” | C )*P(C ) divided by a normalizing constant

  where the constant is Prob (“love deal” | C ) P(C) + Prob (“love deal” | S ) P(S)

  Then, take the complement.

- Answers: 0.5555; 0.4444.

### 0.2.35 Why Is Naive Bayes Naive

- Don't see underlying process (e.g. who is using the words) but get to see the outcome (e.g. the words that the person used).
- Answer: Word order is being ignored in Bayes Theorem whereas the words used and the length of the message (in terms of which words are used) are used.

### 0.2.36 Naive Bayes Strengths and Weaknesses

- Pros: Easy to implement, efficient
- Cons: Can break (e.g. phrases with distinct meanings)
- Good for text classification because can treat each word as a feature.

### 0.2.37 Congrats on Learning Naive Bayes

### 0.2.38 Lesson 2 Naive Bayes Mini-Project

- See code in "/Udacity_Machine_Learning/Lesson_2_Naive_Bayes_Mini_Project_Code.py"
- : Answer to Quiz on Author ID Accuracy: no. of Chris training emails: 7936, no. of Sara training emails: 7884. Accuracy: 0.973833902162
- Answer to Quiz on Timing Your NB Classifier: training time: 1.417 s, predicting time: 0.162 s. Training time is greater than the prediction time.

## 0.3 SVM, Support Vector Machines

### 0.3.1 Welcome to SVM

- SVM is a very popular algorithm

### 0.3.2 Quiz: Separating a Line

- SVM takes in data from 2+ classes as input and draws a line to separate the classes
- Answer: The diagonal line that separates the Xs and the Os.

### 0.3.3 Quiz: Choosing Between Separating Lines

- Answer: The vertical line is the best separator.

### 0.3.4 Quiz: Choosing Between Separating Lines

- Want to choose a line that maximizes the distances to the nearest points in either class. Margin- maximizes distance to the nearest point.
- Answer: Something else because it maximizes the distances to the nearest points in either class (to be the most robust to classification errors).

### 0.3.5 Quiz: Practice with Margins

- Answer: The middle line maximizes the margin. It is the most robust to classification errors.

### 0.3.6 Quiz: SVMs and Tricky Data Distributions

- Answer: The line that fully separates the red from the blue points (diagonally downward).
- SVM prioritizes correct classification over maximizing the margin.

### 0.3.7 Quiz: SVM Response to Outliers

- What happens when no decision surface exists that completely separates the classes of data?
- Answer: Do the best it can

### 0.3.8 Quiz: SVM Outlier Practice

- SVM ignores extreme outliers.
- Answer: The line on the right is the best separator.

### 0.3.9 Handoff to Katie

- Making your own SVM

### 0.3.10 SVM in SKlearn

- Code: Import statement, training data, training features, create classifier, fit classifier, do a prediction.
- Important: import svm, classifier is svm.SVC()

### 0.3.11 SVM Decision Boundary

- Unlike with our Naive Bayes classifier, the SVM decision boundary will be a straight line.

### 0.3.12 SVM Coding Up the SVM

- Answer: See "Lesson_3_Section_12_Coding_Up_the_SVM for code. Also, the accuracy is 0.92, so it does better than the Naive Bayes classifier.

### 0.3.13 Nonlinear SVMs

- SVM can do complicated shapes in the decision boundary.

### 0.3.14 Quiz: Nonlinear Data

- Answer: No. Given our definition so far, SVMs will not separate this dataset.

### 0.3.15 Quiz: A New Feature

- Can put x squared and y squared into SVM in addition to x and y inputs.
- Answer: Yes, this is now separable.

### 0.3.16 Visualizing A New Feature

- Can do a linear transformation, which means we can look at it in a new coordinate system.

### 0.3.17 Quiz: Separating with the New Feature

- Answer: Yes, the data classes are now separable.

### 0.3.18 Quiz: Practice Making a New Feature

- Answer: Using the absolute value of X makes the data classes separable.

### 0.3.19 Kernel Trick

- Use kernels to change x,y input space to a much larger input (high dimensional) space. And this can lead to a non-linear separation for x, y

### 0.3.20 Quiz: Playing Around with Kernel Trick

- How to create an SVC and specify a kernel type
- Answer: All + more. There are many kernel types.

### 0.3.21 Quiz: Kernel and Gamma

- Parameters in machine learning- arguments passed when you create your classifier (prior to fitting)
- Parameters for an SVM- kernel (e.g. linear, rbf); C; gamma
- Answer: The plot on the far right best represents an SVM with a linear kernel and a gamma = 1.0.

### 0.3.22 Quiz: SVM C Parameter

- C- controls the tradeoff between a smooth decision boundary and classifying training points correctly.
- Answer: A smaller value of C will cause the optimizer to look for a larger margin (even if this means that the decision boundary will misclassify more points. Therefore, a large C means that you expect that you will get more training points correct.

### 0.3.23 Quiz: Gamma Parameter

- Gamma- defines how far the influence of a single training example reaches. Low values = far reach (even the points far from the potential decision boundary get taken into account when evaluating where to put the decision boundary) and high values = close.

### 0.3.24 Quiz: Overfitting

- Overfitting is a common problem in machine learning
- Answer: C, Gamma, and the Kernel type are all parameters that can impact fit/overfitting.

### 0.3.25 SVM Strengths and Weaknesses

- SVM works well in complicated domains where there is a clear margin of separation.
- SVM doesn't work well in large data sets or data sets with lots of noise (e.g. overlapping classes)

### 0.3.26 Lesson 2 Naive Bayes Mini-Project

- See code in "/Udacity_Machine_Learning/Lesson_3_SVM_Mini_Project_Code.py"
- Answer to Quiz on SVM Author ID Accuracy: The accuracy is 0.984072810011.
- Answer to Quiz on SVM Author ID Timing: The training time is 144.617 s and the predicting time: 14.203 s. Therefore, SVM is slower than Naive Bayes in this case.

- Answer to Quiz on A Smaller Training Set: The accuracy is 0.884527872582.
- Answer to Quiz on Speed-Accuracy Tradeoff: The two cases that happen in real time, flagging credit card fraud and voice recognition.
- Answer to Quiz on Deploy an RBF Kernel: Interestingly, with this more complex kernel, our accuracy is lower, 0.616040955631.
- Answer to Quiz on Optimize C Parameter: The accuracy when C = 10 is 0.616040955631, C = 100 is 0.616040955631, when C = 1000 is 0.821387940842, when C = 10000 is 0.892491467577. This means that the accuracy is greatest when C is highest (10000).
- Answer to Quiz on Accuracy after Optimizing C: The accuracy when C = 10000 is 0.892491467577. Based on the definition of C given earlier in the lesson, greater C equals more complex decision boundaries.
- Answer to Quiz on Optimized RBF vs. Linear: The accuracy of the optimized RBF is 0.990898748578.
- Answer to Quiz on Extracting Predictions from An SVM: The SVM predicts 1 for element 10, 0 for element 26, and 1 for element 50.
- Answer to Quiz on How Many Chris Emails Predicted: He is expected to have authored X emails. Note: when the partial training set is used, this answer is 1018 emails.

### 0.3.27 Final Thoughts on Deploying SVM

- Generally, Naive Bayes classifiers are better for text than SVM.

## 0.4 Lesson 4: Decision Trees

### 0.4.1 Welcome to Decision Trees

- Decision trees are the 3rd supervised classification algorithm that we are going to cover in this course.
- Decision trees are used frequently in machine learning.

### 0.4.2 Quiz: Linearly Separable Data

- The data in the plot are not linearly separable (would need 2 lines).

### 0.4.3 Quiz: Multiple Linear Questions

- Decision trees allow you to ask multiple linear questions (one after another).
- Answer: No, if it is not windy, then you stop.

### 0.4.4 Quiz: Constructing a Decision Tree First Split

- Answer: The initial split should be a X less than 3.

### 0.4.5 Quiz: Constructing a Decision Tree 2nd Split

- Answer: The second split should be on Y less than 2.

### 0.4.6 Quiz: Class Labels After Second Split

- Answer: If Y is less than 2, then the class label is the red cross.

### 0.4.7 Quiz: Constructing a Decision Tree/ 3rd Split

- Answer: The best split when X is less than 3 is Y = 4.
- We will use an algorithm to find the decision tree boundary(ies).

### 0.4.8 Quiz: Coding a Decision Tree

- Answer: See "Lesson_4_Section_8_Coding_A_Decision_Tree for code to produce a plot with the data and decision boundary.

### 0.4.9 Quiz: Decision Tree Accuracy

- See "Lesson_4_Section_9_Decision_Tree_Accuracy.py" code.
- Answer: The accuracy is 0.908.

### 0.4.10 Quiz: Decision Tree Parameters

- Many parameters that we can try to tune.
- We are first going to look at the min_sample_split. This might be helpful for our overfitting problem because we can control the amount of splits on the tree. This asks the question about how many samples can be in the node before we stop splitting. The default is 2 samples.
- Answer: 1. You would not be able to split a node (or leaf) with 1 sample when using the min_sample_split default of 2.

### 0.4.11 Quiz: Min Samples Split

- Answer: The plot on the left (with a more complicated decision boundary) has min_sample_split = 2 and the plot on the right (with a "cleaner" decision boundary) has a min_sample_split = 50.

### 0.4.12 Quiz: Decision Tree Accuracy

- See "Lesson_4_Section_12_Decision_Tree_Accuracy.py" code.
- Answer: The accuracy when min_sample_split = 2 is 0.908 and when min_sample_split = 50 is 0.912.

### 0.4.13 Data Impurity and Entropy

- Entropy- controls how a decision tree decides where to split the data (measure of impurity in a bunch of examples). For example, in this data set, we are trying to determine whether the car should go fast or slow. However, there might also be a speed limit that is enforced at certain places, so even if the terrain warrants that we go fast, if there is an enforced speed limit, we will go slow.

### 0.4.14 Quiz: Minimizing Impurity in Split

- Answer: The plot with speed limit on the x-axis has a higher purity than that plot with bumpiness on the x-axis.

### 0.4.15 Formula of Entropy

- When all examples are in the same class, the entropy is 0. When all examples are spread across all classes, the entropy is 1.

### 0.4.16 Quiz: Entropy Calculation Part 1

- Answer: There are 2 slow observations.

### 0.4.17 Quiz: Entropy Calculation Part 2

- Answer: There are 4 total observations.

### 0.4.18 Quiz: Entropy Calculation Part 3

- Answer: 0.5 observations are slow.

### 0.4.19 Quiz: Entropy Calculation Part 4

- Answer: 0.5 observations are fast.

### 0.4.20 Quiz: Entropy Calculation Part 5

- The entropy is -1(0.5*log2(0.5)*2) = 1.

### 0.4.21 Information Gain

- Information gain is defined as the entropy(parent) - [weighted average]entropy(children)
- The decision tree algorithm maximizes the information gain

### 0.4.22 Quiz: Information Gain Calculation Part 1

- Answer: 3 observations have a grade of "steep".

### 0.4.23 Quiz: Information Gain Calculation Part 2

- Answer: The entropy of this node is 0 because all of the observations are in the same class (fast).

### 0.4.24 Quiz: Information Gain Calculation Part 3

- Answer: p_slow = 2/3 because 2/3 of the observations are slow.

### 0.4.25 Quiz: Information Gain Calculation Part 4

- Answer: p_fast = 1/3 because 1/3 of the observations are fast.

### 0.4.26 Quiz: Information Gain Calculation Part 5

- Answer: Entropy = -1((2/3)*log2(2/3) + (1/3)*log2(1/3)) = 0.91829583405.

### 0.4.27 Quiz: Information Gain Calculation

- Entropy of parent = 1
- Entropy of children = (3/4)*(0.9184) + (1/4)*(0)
- Information gain = Entropy(parent) - [weighted average]*entropy(children) = 1 - 0.6888 = 0.3112.
- Answer: The information gain that we get if we split based on grade = 0.3112.

### 0.4.28 Quiz: Information Gain Calculation Part 7

- Entropy of parent = -1*((1/2)*log2(1/2)*2) = 1
- Answer: When you sort based on bumpiness, there is still one slow and one fast observation in the category of "bumpy" and one slow and one fast observation in the category of "smooth". The observations are still spread evenly throughout the classes. Therefore, the entropy of bumpy is 1.

### 0.4.29 Quiz: Information Gain Calculation Part 8

- Answer: When you sort based on bumpiness, there is still one slow and one fast observation in the category of "bumpy" and one slow and one fast observation in the category of "smooth". The observations are still spread evenly throughout the classes. Therefore, the entropy of flat is 1.

### 0.4.30 Quiz: Information Gain Calculation Part 9

- Entropy of the parent = 1
- Weighted average of the entropy of the children = 1
- Answer: The information gain is 0. (1-1).

### 0.4.31 Quiz: Information Gain Calculation Part 10

- Answer: When you sort based on speed limit, there are 2 slow observations when there is an enforced speed limit and 2 fast observations when there is not an enforced speed limit. The observations are separated by class and therefore, the entropy is 0. The information gain is 1-0=1.

### 0.4.32 Tuning Criterion Parameter

- Default criterion for the Decision Tree Classifer in sklearn is "gini" (the Gini impurity) but it can also support "entropy" for information gain.

### 0.4.33 Bias-Variance Dilemma

- High bias- the machine learning algorithm doesn't learn/isn't highly affected by the training
- High variance- the algorithm learns a lot on the data, to the point that it can't generalize to novel data input types.
- Ideally, you want something that learns from the training, but isn't highly limited by only the data types in the training.

### 0.4.34 DT Strengths and Weaknesses

- Strengths- easy to use, results are clearly interpretable. Can build larger classifiers (ensemble methods).
- Limitations- prone to overfitting (so have to be careful with parameter tunes).

### 0.4.35 Decision Tree Mini-Project

- See code in "/Udacity_Machine_Learning/Lesson_4_Decision_Tree_Mini_Project_Code.py"
- Answer to Quiz on Your First Email DT: The accuracy is 0.978953356086.
- Answer to Quiz on Speeding Up Via Feature Selection: There are 3,785 features in the training data.
- Answer to Quiz on Changing the Number of Features: When you change the percentile of training data from 10 to 1, the number of features is 379.
- Answer to Quiz on SelectPercentile and Comparing: A large value for percentile lead to a more complex decision tree (because you now have to classify many more features).
- Answer to Quiz on Accuracy Using 1% of Features: The accuracy is 0.966439135381.