Project 7
Peter Kurtz, Kaelyn Hughes

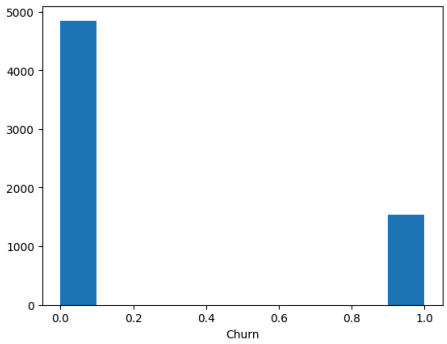
GitHub repository
Slides: Project 7

Introduction: Our goal with the first dataset was to predict whether or not a customer at a bank would leave the bank or not using their age and the balance in their bank account. The dataset we used can be found on Kaggle here. We used logistic regression and SVM. We found that SVM with a radial basis function kernel was the best at predicting.

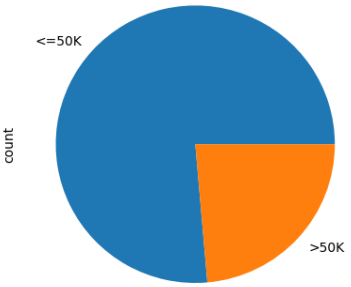
In the second dataset, we wanted to know which factors might predict whether a person makes a higher income - our dataset specifically recorded whether the person made above $50K. We especially looked at education level, age, and hours per week worked; we hoped to determine the extent to which these factor into income level. This dataset was found on UCI's Machine Learning Repository, and in our analysis we again used logistic regression and SVM.

Methods:

Dataset 1: Our target variable was churn which is a 0 if the customer stayed and a 1 if the customer left. We decided to take out customers who had zero dollars in their accounts since these customers did not participate in the bank. We standardized our predictor variables. Our data was imbalanced for our predictor variable with many churn variables as 0 as seen in the histogram of counts of churn data below. Therefore, we decided to make the model with a subsample of the data that ensured an equal amount of 1s and 0s for the churn variable. We ran multiple models to determine which was best.



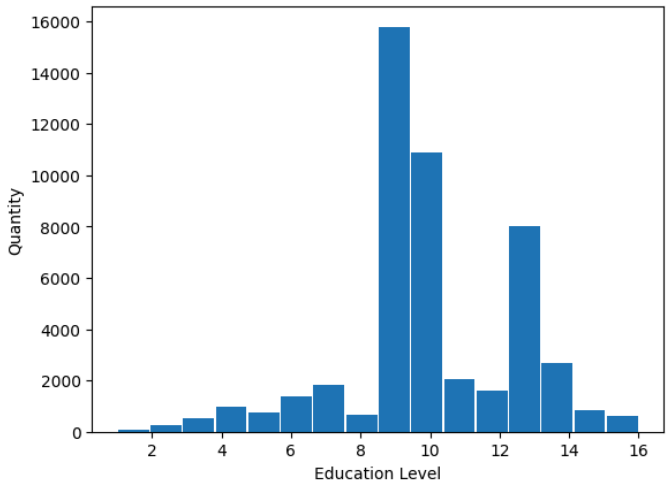
Dataset 2: For this dataset, our target variable was titled 'income', which fell into either "up to $50K" or "above $50K". We did do a bit of data cleaning here as for some reason, there was a period at the end of around a third of the values in this column which served no apparent purpose. As is evident in the pie chart, the majority of our data points fell under the $50K mark. It is also important to mention that we standardized our predictor variables.



Question 1: Which variables are predictive of the target variable?

Dataset 1: The variables that were predictive of the target variable were age and balance. Our method for finding the variables was to look at the heatmap for the variables and determine which were predictive of our target variable for the given models. We initially selected more variables but the more variables that were added, the worse the model did. Age was the best at predicting churn in the models.

Dataset 2: Education was unsurprisingly quite predictive of income level, as were age and hours per week worked to an extent; very few of those who made a higher salary worked less than 40 hours a week or were under 35 years old. We did also note that several levels of education were by far the most popular (distribution shown in the graph below); we specifically wanted to point out level 9, which represents graduation from high school and nothing past; level 10, which represents some college but no degree; and level 13, which represents a bachelor's degree.



Question 2: Can logistic regression or a linear SVM predict well?

Dataset 1: Both logistic regression and linear SVM were not very good at predicting. The f-scores for the logistic regression model when trying to predict on a random sample of the dataset were 0.781 and 0.550 for labels 0 and 1 respectively. The low f-score for label 1 was due to the precision which was a score of 0.439. Linear SVM had f-scores of 0.798 and 0.563 for labels 0 and 1 respectively with a low f-score for label 1 also due to a low precision score.

Dataset 2: Logistic regression had consistently good precision and recall when predicting points below the $50K line, and consistently terrible precision and recall when predicting points above it. When predicting solely off of age and hours worked per week, we had a recall score of 0.0899 for those with an income of above $50K! Our best overall results came when we took all three features into account, which gave us a precision score of 0.61, recall of 0.31, and f-score of 0.41 for high-earning data points, along with a precision score of 0.81, recall of 0.94, and f-score of 0.87 for lower-earning points.

Question 3: What do plots of selected pairs of variables look like? Where is the decision boundary in those plots?

Dataset 1: Figure 3.1 is a plot of Age and Balance with red signifying a 0 for churn and blue a 1. Figure 3.2 is the decision boundary made from the RBF support vector machine.

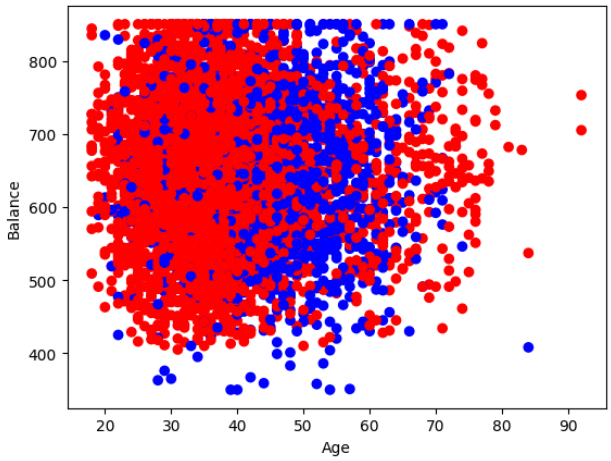


Figure 3.1

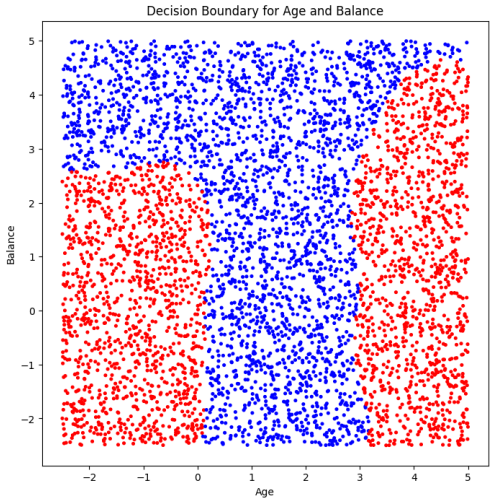
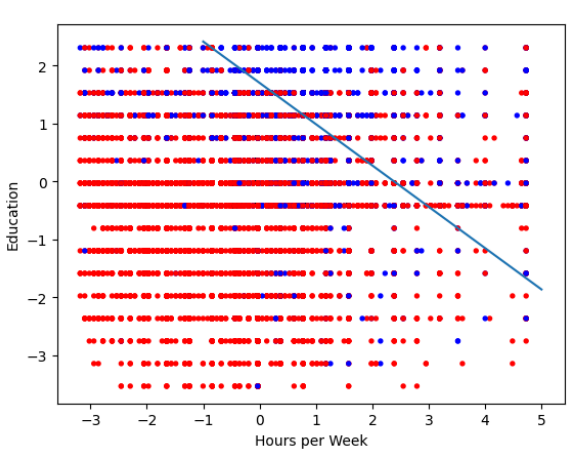
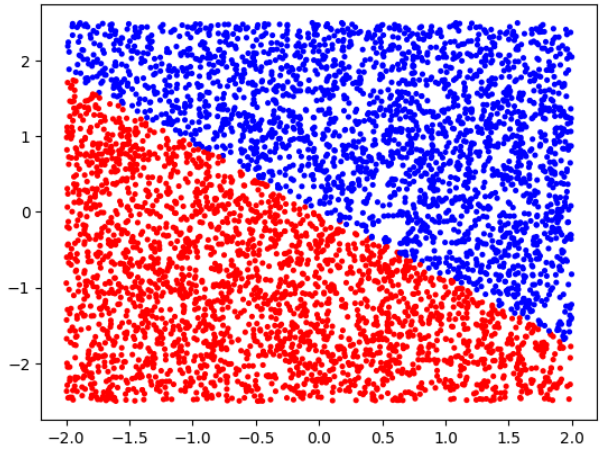


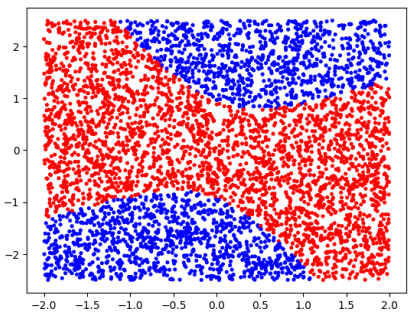
Figure 3.2

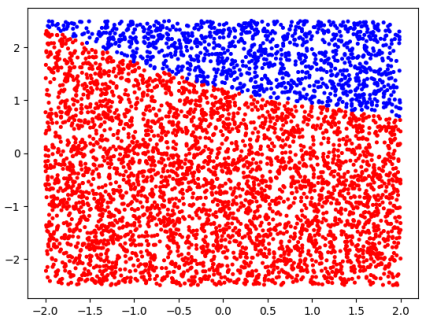
Dataset 2: We looked at several visualizations of the data. One was our logistic regression model, which performed poorly; as we can see in the graph below and to the left, lots of data falls to the wrong side of the line:



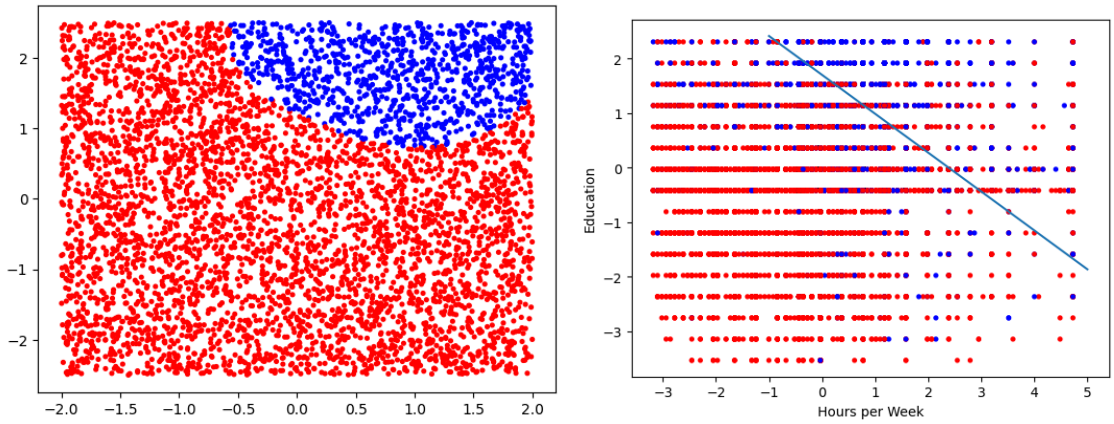


Above and to the right is the decision boundary used by the linear SVM; this performed even worse than the logistic regression analysis. Below we see the decision boundary used by the polynomial kernel SVM and a degree of 2, which had f-scores of 0.81 and 0.41 for high- and low-salary data, respectively. Next to it is the decision boundary from the polynomial kernel again, but with a degree of 3; we found this to be a bit more accurate with scores of 0.86 and 0.43.





The RBF kernel was the first that didn't need the higher-salary data to be weighted! This gave us our best results so far with this dataset; we got respective f-scores of 0.87 and 0.45. When we weighted the higher-salary data, we found that to be a bit more accurate (0.85 and 0.53), so we chose to go with that. We noted that for the RBF model, precision, recall, and f-score are all nearly the same for each label. The decision boundary for this model is below, next to the original for comparison.



Question 4: How generalizable are the different models on your data? How does the bias-variance tradeoff affect which model you might choose?

Dataset 1: The models were not very generalizable. The classification scores for the data that was trained were very good but classifying on a random set of data from the original dataset resulted in a low precision score for when churn is 1. For example, the RBF SVM model resulted in the following classification scores: Precision: [0.892, 0.458], Recall: [0.722, 0.730], f-score = [0.79824561, 0.56329114], Support: [378, 122] with the first value being for label 0 and the second value being for label 1. The classification scores for the training dataset were all very good. This example is characteristic of most models that were created where the precision for label 1 being poor. Therefore, we had a model that was high in variance since it did well on data it was trained on but poor on test data.

Dataset 2: Our favorite model (the RBF SVM) actually turned out to be very generalizable! By splitting the data into train and test sets, we found very similar if not better f-scores - 0.85 for lower-salary labels and 0.55 for higher-salary ones.

Question 5: Is there a difference between the polynomial and RBF SVMs?

Dataset 1: The RBF SVM did better than the polynomial SVM. The polynomial f-scores for a random sample of the dataset for labels 0 and 1 were [0.831, 0.195]. Both precision and recall were poor for label 1. The RBF SVM f-scores were [0.798, 0.563] as discussed in question 4.

Dataset 2: We found that there was both a difference between degrees of the polynomial SVM and between the polynomial and RBF SVMs. Using a higher degree in the polynomial dataset got us to our peak f-score for the lower-income label but had little effect on accuracy with higher-income labels, and using the RBF kernel kept a similar level of accuracy in the former while significantly improving accuracy in the latter.

Question 6: What effect does changing the class_weight in an SVM have on your data? How might this be important for this data?
Dataset 1: Changing the weight was very bad for this dataset. Changing the weight for one category would cause the other category's classification statistics to be 0. For example, setting 0 as weight 1 and 1 as weight 2 for RBF SVM resulted in a precision for category 0 to be 0 and a recall to be 0.

Dataset 2: Changing the weight absolutely made a difference! When using a linear model SVM, using evenly weighted classes really struggled to predict higher-salary data points at all. When we weighted the higher-salary class heavier, our model was able to give us the confusion matrix scores we were looking for, but as we continued to weight that class more heavily, we found that there was ultimately an overall decrease in accuracy based on f-score. When using the polynomial kernel, we moved that up to a weight of 3 to get similar results.

Question 7: Is there a difference in runtime performance?
Dataset 1: The RBF SVM ran longer than logistic regression and polynomial SVM. RBF SVM took 7.1 seconds, the polynomial SVM took 3.8 seconds, and logistic regression took 0 seconds.

Dataset 2: Definitely! Some of this comes down to machine speed, but logistic regression on this dataset took under half a second. In contrast, all SVM analyses took at least twenty seconds, and the RBF SVM took around a minute.

Question 8: Logistic regression and LinearSVC use one-vs-rest (OVR) for multi-class classification. SVC uses one-vs-one (OVO). Where n is the number of classes, OVR learns n models, whereas OVO learns n(n-1)/2 (n choose 2) models. What effect does this have on performance?
OVR has a faster runtime than OVO. This is because the n^2 in the OVO will grow much quicker than n in OVR.