CS 7641 Machine Learning
Assignment 1

Philip Bale
pbale3

Due February 4th, 2018 11:59pm

2 most important plots: learning curve and model complexity curve

Classification Problems

1) US Permanent Visa Applications

Overview

The first classification problem revolves around classifying whether or not a person's US permanent visa application will be accepted or denied based on the parameters of their application. Among the features used in the classifcation are:

- Job features: Industry code, job class, wage rate, wage type
- Geographic features: Country of citizen and employer location

The classes observed for this dataset are simply 'approved' and 'denied'. The dataset contains 374365 total samples.

Why is the dataset interesting?

This dataset is interesting due to its potential to aid in the visa application process from a cost and time savings potential. It could also enable confidence in those interested in applying for a US permanent visa but doubting their chances of acceptance. At the end of the day, the goal is it to try to determine the application result before time, money, and other resources are spent. As someone who has worked with a large number of first-generation visa holders and immigrants, I am extremely interested in building tools to help others to achieve the same.

From a machine learning perspective, the dataset is incredibly interesting due to its wide variety of features and the variety of values those features can take. An immense number of job types, wage rates, and citizenships alone create an extremely diverse dataset. Additionally, the number of samples available provide a comprehensive picture of historical data, lending towards greater cofidence in training and testing rates.

2) Home Sale Price Predictions

Overview

The second classification problem revolves around classifying a home's price bracket based upon the various characteristics of the home. Among the features used in the classification are :

- Subjective measurements: Exterior condition, house style, overall quality rating, and overall condition
- Objective measurements: Type of dwelling, building type, lot size, neighborhood, year built, and year sold

After an initial review of the dataset, the classes were defined as pricing brackets divided into 100k groups. I.e: 0-100k, 100k-200k, 200k-300k, etc. The dataset contains 1451 samples. An additional dataset containing another 1400 testing samples exists but was not used as it contains unclassified sale prices. It will, however, prove useful for unsupervised learning.

Why is the dataset interesting?

This dataset is interesting for two primary reasons: real-world applicability and participating in a Kaggle challenge. First, modeling home prices is both a difficult and lucrative task. If one can succesfully model home sale prices on large sets of data, he/she can make large amounts of money investing in real estate when he/she detects outliers in listed price vs. what it is expected to sell for. This applies to flipping, investing, and remodeling. Second, the dataset is part of an ongoing Kaggle competition that does not have a winning solution yet. By taking part of the competition, the dataset presents the opportunity to work towards a winning solution and advance ones algorithms over time.

Houses can have a very large amount of features–with a large amount of variety in the individual features. Similarly, housing is prone to personal taste and frequent need for upgrades/modernization. In such, I believe price estimation is an excellent problem, full of depth and complexity, that is suitable for a machine learning approach.

General Data Processing

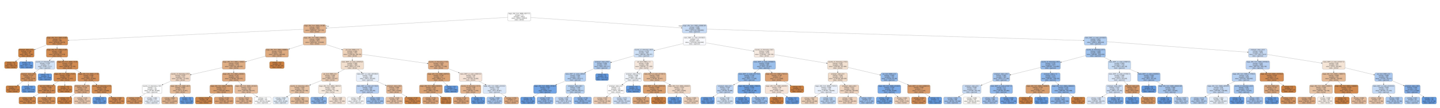
The datasets I used were both relatively clean to begin with. One small problem, however, was that a lot of my features on both datasets were text-based. To transform the features into numeric values suitable for the machine learning algorithms, I used a label encoder built into ScikitLearn.

I also did a small amount of preprocessing of the data to make it more suitable for classification. I dropped all unnecessary columns to help speed up with data processing in general–which proved immensely helpful when dealing with the more computationall intensive algorithms. In the case of home prices, I precalculated the brackets based on the 'sale price' data label. In the case of visa applications, I pregrouped the case outcome so that results such as 'certified' and 'certified withdrawn' are both concerned as 'approved' conditions whereas 'denied', 'invalidated', and 'rejected' all resolve to 'denied'.

Decision Trees

A decision tree classifier was the first algorithm applied to the datsets. Various values of max_depth were tested as a means of pruning unnecessary leaves. Similarly, a grid search was used to test whether a 'gini' or 'entropy' criterion was more effective.

US Permanent Visa



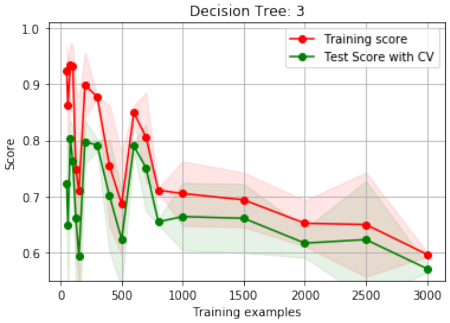
Permanent Visa Decision Tree with max_depth of 7

Depth	Criterion	Tree Size	Train %	Train Time	Test %	Test Time
1	gini	3	0.7657	0.1212	0.7680	0.0010
3	gini	15	0.6775	0.1417	0.6749	0.0009
5	gini	63	0.7136	0.2175	0.7067	0.0011
7	entropy	211	0.7256	0.2599	0.7193	0.0011
10	gini	889	0.7584	0.3192	0.7072	0.0011
15	gini	3013	0.8628	0.4199	0.7582	0.0014
20	gini	4751	0.9299	0.4517	0.7917	0.0010
25	gini	5349	0.9545	0.5284	0.8032	0.0010
30	gini	5471	0.9588	0.4390	0.7982	0.0011
40	entropy	5277	0.9592	0.4522	0.8070	0.0009
50	gini	5493	0.9592	0.4631	0.8032	0.0010

Results at multiple depths for best criterion via grid search

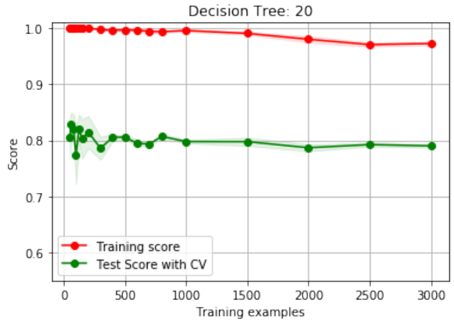
	Accepted	Denied
Accepted	96	135
Denied	217	1376

Test Data Confusion matrix



Learning Curve for max_depth = 3

Learning Curve for max_depth = 10



Learning Curve for max_depth = 20

Home Sale Prices

X	Test %	Runtime
1	2	3
1	2	3