Build a Student Intervention System

Updated (13th Mar 2016)

Classification vs Regression

Classification; Because we're trying to classify/predict a discrete (in this case, binary) class.

Exploring the Data

Total number of students: 395
Number of students who passed: 265
Number of students who failed: 130
Number of features: 30
Graduation rate of the class: 67.09%

Preparing the Data

Identify feature and target columns

Feature column(s):-
['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences']
Target column: passed

Preprocess feature columns

Processed feature columns (48):-
['school_GP', 'school_MS', 'sex_F', 'sex_M', 'age', 'address_R', 'address_U', 'famsize_GT3', 'famsize_LE3', 'Pstatus_A', 'Pstatus_T', 'Medu', 'Fedu', 'Mjob_at_home', 'Mjob_health', 'Mjob_other', 'Mjob_services', 'Mjob_teacher', 'Fjob_at_home', 'Fjob_health', 'Fjob_other', 'Fjob_services', 'Fjob_teacher', 'reason_course', 'reason_home', 'reason_other', 'reason_reputation', 'guardian_father', 'guardian_mother', 'guardian_other', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences']

Joshua Newnham

Split data into training and test sets

Training set: 300 samples
Test set: 95 samples

Training and Evaluating Models

Logistic Regression

What is the theoretical O(n) time & space complexity in terms of input size?
Time complexity: O(N)
Space complexity: O(1)

What are the general applications of this model? What are its strengths and weaknesses?
General applications

- likelihood of a homeowner defaulting on a mortgage
- Predicting the likelihood of a 'click' for ad-serving
- Predict whether a patient has a given disease based on observed characteristics

Advantages;

- High degree of interpretation i.e. coefficients can then be interpreted in order to understand the direction and strength of the relationships between the explanatory variables and the response variable
- Fast to train
- Fast in making predictions
- Low memory requirements
- Scales well
- Can use a cost function to reduce over-fitting
- Allows for online training (incremental training)

Disadvantages;

- Disregards feature dependencies
- Does not handle categorical data
- Influenced by outliers
- Affected by imbalanced training data

Given what you know about the data so far, why did you choose this model to apply?
Binary classification problem

		Training Set		Test Set	
Training set size	**Training time**	**Prediction time**	**F1 Score**	**Prediction Time**	**F1 Score**
100	0.001	0.0	0.90780141844	0.0	0.765625
200	0.001	0.0	0.855172413793	0.0	0.779411764706
300	0.002	0.0	0.846846846847	0.0	0.805970149254

F1 score for test set: 0.805970149254

SVM

What is the theoretical O(n) time & space complexity in terms of input size?
Time complexity: O(n_samples^2 x n_features) for RBF kernel and O(n_sample x n_features) for linear SVMs
Space complexity: O(1)

What are the general applications of this model? What are its strengths and weaknesses?
General applications

- Text classification
- Image classification

Advantages;

- Works well even if your data isn't linearly separable (with the right kernel)
- High accuracy
- Good at handling high-dimensional spaces

Disadvantages;

- Memory-intensive
- Hard to interpret
- Prone to overfitting noisy data
- Difficult to tune
- Don't scale well

Given what you know about the data so far, why did you choose this model to apply?
The reason for choosing SVM was to take adjust of its kernel properties in that being able to model more complex relationships. Thus used as a comparable to Logistic Regression.

F1 score for test set: 0.783783783784

		Training Set	Test Set

Training set size	Training time	Prediction time	F1 Score	Prediction Time	F1 Score
100	0.001	0.001	0.888888888889	0.001	0.774193548387
200	0.003	0.002	0.867549668874	0.001	0.789115646259
300	0.005	0.004	0.876068376068	0.001	0.783783783784

KNeighborsClassifier

What is the theoretical O(n) time & space complexity in terms of input size?
Time complexity: O(N)
Space complexity: O(N)

What are the general applications of this model? What are its strengths and weaknesses?
General applications
- Simple classification
- Character recognition
- Clustering/Topic (text) classification

Advantages;
- Simple
- Work well with datasets with a small number of features
- 'Lazy' learner (online training)
- Nonparametric decision boundaries

Disadvantages;
- Memory intense (require a lot of memory)
- Doesn't 'learn'
- Affected by noisy data
- Slow (distanced need to be calculated across all instances)

Given what you know about the data so far, why did you choose this model to apply?
Despite the algorithm not meeting the functionality requirements (timely to reduce computational time) I choose KNN as a comparison/control for it's non-parametric properties (flexible decision boundaries).

F1 score for test set: 0.780821917808

		Training Set		Test Set	
Training set size	**Training time**	**Prediction time**	**F1 Score**	**Prediction Time**	**F1 Score**
100	0.001	0.001	1.0	0.001	0.757142857143

200	0.0	0.002	1.0	0.001	0.765957446809
300	0.00	0.006	1.0	0.002	0.772413793103

Choosing the Best Model

(1)

I have chosen Logistic Regression as the model; This model produced the best F1 score out of the 3 models I tested, it also is the most effective (least computation cost) model addressing some of the functional and business constraints of the intended service. Another important aspect (by product) is the model presents properties of 'good performers' and 'bad performers' such that effective intervention can take place using these are further insights.

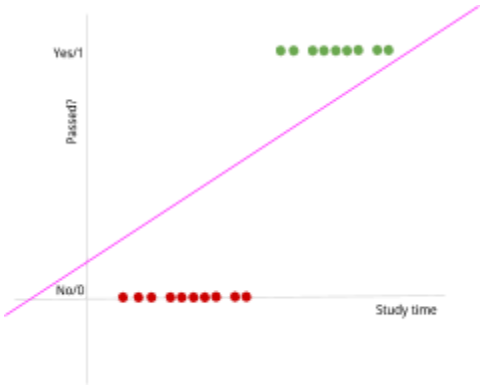
(2)

Logistic Regression is one of the simplest Machine Learning algorithms and is based on the premise that the data can be split linearly by a form of function applied to the independent variables (features).

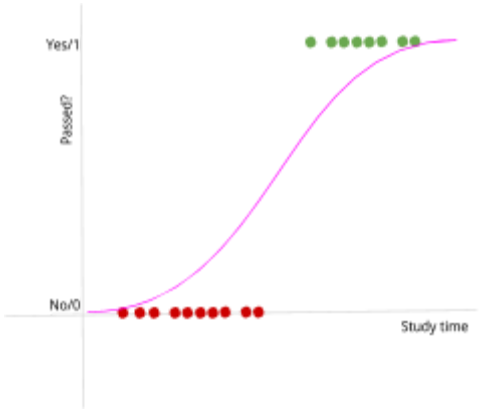
This is probably best illustrated through the use of a simple example; Imagine a dataset that has recorded student **study time** and whether they **passed or not** (i.e. binary classification).

Here the feature is **'study time (hours)'** and class is a binary output of **y/1 (passed) or n/0 (failed)**. The job of the Logistic Regression is to model the probability of the student passing based on their ‘study time’ (at least in context of this example).

Similar to Linear Regression, we find a line that best fits the data by deriving a constant and weights for each of the feature variables (i.e. study time) - a best fit line might look similar to the figure shown below..



But unlike Linear Regression, our goal is to predict the probability of a student passing given their study time. To achieve this (constrain the output between 0 and 1 aka probability) we pass the function (constant and weights/coefficients) through the logistic function, which will return a value between 0 and 1 (our probability). The following figure illustrates how it differs (visually) from above.

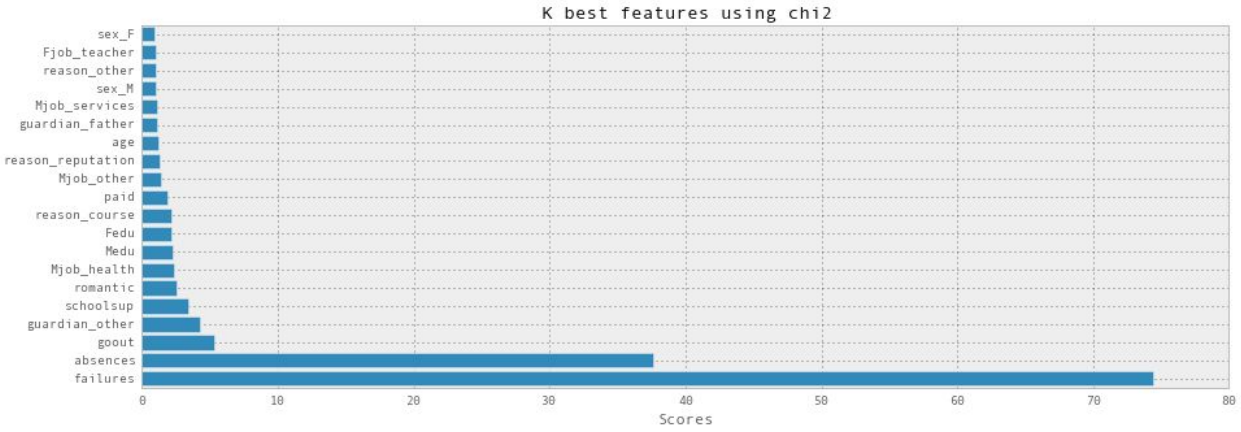


Once the model has been trained, predictions are made by passing the feature variables through the model (using the derived constant, feature weights and logistic function) to obtain the probability, classification is then derived.

The weights are calculated using maximum likelihood estimate method, which returns the parameters that result in the minimum error between the trained predicted value and actual value (from the training dataset).

(3)
Tuning
1 - Feature Selection

2 - Parameter selection using GridSearch (k and C)



The above graph displays the top 20 features using SelectKBest (from SKLearn) with their associated scores using chi2.

Best Classifier

Best f1 score **0.805555555556**

Best SelectKBest SelectKBest(k=10, score_func=<function chi2 at 0x10adc9f50>)

Best Estimator LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, penalty='l1', random_state=None, tol=0.0001)

Admittedly a little complexed why my test f1 score is less than when I began.

Changes (2016-03-13)

- Updated feature count
- Fixed ‘bug’ (import dependency)
- Added performance summary tables for each of the ML algorithms (including training and prediction time and F1 score)
- Elaborated on the discussion describing the chosen algorithm (Logistic Regression) - including:
    - Role of the logistic function
    - How predictions are made
- Set f1_score as the scoring function of the GridSearch when tuning the model.

Joshua Newnham