Build a Student Intervention System

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: 31
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']

Split data into training and test sets

Training set: 300 samples
Test set: 95 samples

Joshua Newnham

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;

- Decision boundary must be linear
- 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

F1 score for test set: 0.805970149254

Joshua Newnham

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

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

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)

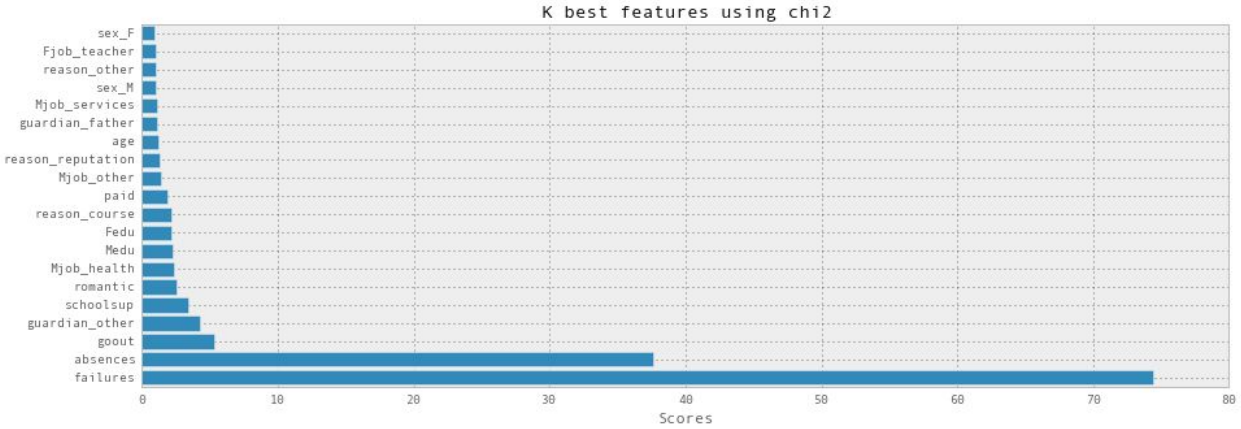
Logistic Regression is one of the simplest Machine Learning algorithms and is based on the premise that the data can be split linearly by a form of function applied to the independent variables (features).

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. Here the independent variable (feature) is 'study time (hours)' and dependent variable (class) is 'passed (y/n)'. The job of the Logistic Regression model (aka function) is to find a line that best splits the data between those who passed and those who failed.

The algorithm follows the following steps:

1. Randomly assign a coefficient to the independent variable (study time)
2. Measure how well this line separates the dependent variable (passed or not).
   Logistic Regression uses statistical deviance for the goodness-of-fit measurement.
3. Adjust coefficient and goto 2, repeat this process until a threshold is met.

(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 **0.802816901408**
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='l2', random_state=None, tol=0.0001)

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