Analyzing the Effects of Non-Academic Features on Student Performance

Gabriela Castillo-Rivera  Jeffrey Hooper  Austin Leverette  James Scruggs

Why You Should Care

- Secondary education is related to unemployment and incarceration rates.
- More than a trillion U.S. dollars spent on education.
  - 5.62% of GDP (average for a developed country)
- Are high schoolers still dropping out in the U.S?
  - Over 1.2 million students drop out per year.

What is Our Data?

School, sex, age, address

First, second, and final grade period

Parents live together?

Family Size

Extra-curricular activities?

Mother's education

Family educational support?

Father's education

Father's job

Mother's education

Extra educational support?

Paid extra for classes?

Weekly study time

Preschool attended?

Free time?

Romantic?

Reason to choose this school

Travel time to school

Want to go to college?

Drink alcohol? Even during the week?

Guardian

Internet access?

Home dramas?

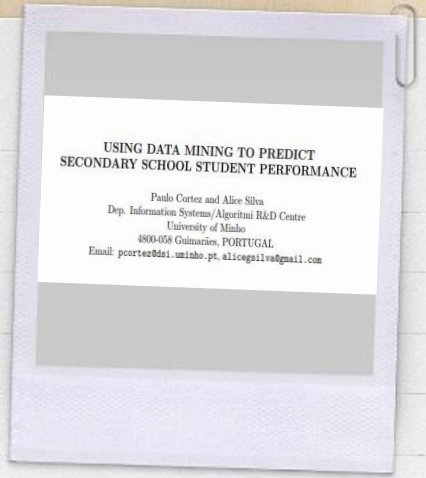
# of past class Failures

Absences

Go out w/ friends?

Healthy?

The Original Study


USING DATA MINING TO PREDICT
SECONDARY SCHOOL STUDENT PERFORMANCE

Paulo Cortez and Alice Silva
Dep. Information Systems/Algoritmi R&D Centre
University of Minho
4800-058 Guimarães, PORTUGAL
Email: pcortez@dsi.uminho.pt, alicegsilva@gmail.com

This data was collected by Paulo Cortez. Past grades and attendance of students from two Portuguese schools were collected along with their answers to a questionnaire that asked them about things such as how much free time they have after school and their parent's highest level of education.

The researchers concluded that in order to best predict student performance, previous grades needed to be included. We wanted to take it a step further. Could removing certain unimportant features improve the neural network's ability to predict student grades? How do the most important features affect student performance?

Let's find out!

Pre-Processing Our Data

Non-Numeric Data

- What about the data that wasn't numeric like mother's job whose values included:
  - teacher, health care related, civil services (e.g. administrative or police), at home or other
- We converted them to numerical data using LabelEncoder from SciKit-Learn

```
# Need this for LabelEncoder
from sklearn import preprocessing

# Label Encoder
le = preprocessing.LabelEncoder()

# Columns that hold non-numeric data
indices = np.array([0,1,3,4,5,8,9,10,11,15,16,17,18,19,20,21,22])

# Transform the non-numeric data in these columns to integers
for i in range(len(indices)):
    # COMMENT THIS, PLEASE
    column = indices[i]
    le.fit(student_data[:,column])
    student_data[:,column] = le.transform(student_data[:,column])
```

Binary Data

- To ensure proper weight updates, we converted all 0's to -1's

```
# Columns that hold binomial data
indices = np.array([0,1,3,4,5,15,16,17,18,19,20,21,22])

# Change 0's to -1's
for i in range(len(indices)):
    column = indices[i]

    # values of current feature
    feature = student_data[:,column]

    # change values to -1 if equal to 0
    feature = np.where(feature==0, -1, feature)
    student_data[:,column] = feature
```

Standardizing Our Data

- To ensure all input is treated equally, we needed to standardize our data.
  - Transform our data to have a mean of 0 with a standard deviation of 1.

```
scaler = preprocessing.StandardScaler()
temp = student_data[:,[2,6,7,8,9,10,11,12,13,14,23,24,25,26,27,28,29,30,31]]
Standardized = scaler.fit_transform(temp)
student_data[:,[2,6,7,8,9,10,11,12,13,14,23,24,25,26,27,28,29,30,31]] = Standardized
```

Encoding Our Labels

- Output data consisted of values ranging from 0 - 20.
  - Portugal scoring system for secondary and post-secondary education.
- Transformed it to represent U.S. conventional A - F grade scale.
  - A = >18; B = >16; C = >14; D = >12; F = <=12
- Followed by one-hot encoding
  - To give us binarization of categorical outputs.

Our Model

The Model

Our model contains two hidden layers that both use the relu activation function.

```
model.add(keras.layers.Dense(800, input_dim = input_size, activation ='relu'))
model.add(keras.layers.Dense(400,activation='relu'))
```

For our loss function we chose Categorical Cross Entropy and our optimizer was Adamax. Adamax was chosen over Adam because it is able to deal with noise.

Initial Model vs Final Model

Initial Model

Accuracy = 76.92 %

Final Model

Accuracy = 95.38

The five features that most negatively impacted accuracy were removed from the data set.

Our Results

Determining Effects of Features

What Features?

- ✗ The five non-academic features whose removal were most detrimental to our net's accuracy.

Finding the Effects on Student Performance

- ✗ For each unique value of each feature, bar graphs were created to display the distribution of student grades (by percent).
- ✗ For example…when considering the effect of extra paid classes on student performance, two bar graphs were created: One for students with these paid classes and one for students without them.

Our Contributions

James

- ✗ Assisted with proposal
- ✗ Importing data set
- ✗ Data pre-processing
- ✗ Introduction and background sections of essay
- ✗ Assisted with abstract
- ✗ Assisted with presentation
- ✗ Assisted with demo

Austin

- ✗ Results section of paper
- ✗ Model building
- ✗ Assisted with demo
- ✗ Saved model and weights for initial and final models for loading in demo.
- ✗ Categorizing final grades
- ✗ Assisted with presentation
- ✗ Assisted with abstract
- ✗ Comments in code and demo

Jeffrey

- ✗ Model building
- ✗ Methods section of essay
- ✗ Assisted with abstract
- ✗ Assisted with proposal

Gabriela

- ✗ With the help of Dr. Phillips, found data set and came up with overall aims
- ✗ Constructed project milestones
- ✗ Assisted with proposal
- ✗ Cleaned up code & assisted with documentation
- ✗ Analyzed effects of top five features
  - ✗ Discussed these results in our essay
- ✗ Assisted with abstract
- ✗ Created skeleton for presentation
- ✗ Created code walkthrough for demo

Thanks for listening!

Let's head over to the demo: