Analyzing the Effects of Non-Academic Features on Student Performance





## Why You Should Care

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



#### What is Our Data?

Extra-curricular activities?

School, sex, age, address

First, second, and final grade period

Parents live together?

Family Size

Mother's education

Father's education

Paid extra for classes?

Weekly study time Preschool attended?

Free time?

Romantic?

Family educational support?

Reason to choose this

school

Travel time to school

Want to go to college?

Drink alcohol? Even during the week?

Father's job

Guardian

Internet access?

Home dramas?

Mother's education

# of past class Failures # Absences

Go out w/ friends? Healthy?

Extra educational

support?

## The Original Study

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.

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 Guimaries, PORTUGAL
Email: pcortez@dsi.uminho.pt, alicegsilva@gmail.com

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:
  - x 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

x To ensure proper weight updates, we converted all O'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

- x 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

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



#### 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.



## Determining Effects of Features

#### What Features?

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

#### Finding the Effects on Student Performance

X For each unique value of each feature, bar graphs were created to display the distribution of student grades (by percent).



X 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.



## James

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

## Austin

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

# Jeffrey

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

## Gabriela

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

# Thanks for listening!

#### Let's head over to the demo:

https://github.com/CSCI4850/S19-team3-project/tree/master/Demo

#### Check the ReadMe for instructions:

https://github.com/CSCI4850/S19-team3-project/blob/master/README.md



