Student Performance Analysis

FINAL PROJECT REPORT

JaSONma

160 pleasant st.

Malden, MA and 02148

Version 0.1.0

08/23/2020

# VERSION HISTORY

| VERSION | APPROVED BY | REVISION DATE | DESCRIPTION OF CHANGE | AUTHOR |
| --- | --- | --- | --- | --- |
| 0.1 | Jason MA | Aug.10.2020 | Initial | Jason MA |
| 0.2 | Jason MA | Aug.13.2020 | Preprocess data | Jason MA |
| 0.3 | Jason MA | Aug.17.2020 | Exp1 | Jason MA |
| 0.4 | Jason MA | Aug.20.2020 | Exp2 | Jason MA |
| 1.0 | Jason MA | Aug.23.2020 | Finish | Jason MA |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PREPARED BY** | Jason Ma | **TITLE** | Author | **DATE** | Aug.23.2020 |
| **APPROVED BY** | Jason Ma | **TITLE** | Author | **DATE** | Aug.23.2020 |

# ROLES AND RESPONSIBILITIES

| NAME | ROLE | RESPONSIBILITIES |
| --- | --- | --- |
| Jason MA | Author | Exploring data, cleansing data, process data, design network,  train model, test model, documentation. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Project Summary

In this project, I dive into a student grades dataset from a Portuguese high school. I tried to figure out what behaviors or features impact students’ grades most. Would it be the same with many parents’ opinions that exists some “bad” activities cause a decrease in grades. In the beginning, I pick a little data and assume a null-hypothesis and alternative-hypothesis set. By modeling the data and using a single-tailed Binomial Distribution examination, I get the conclusion with data. Then, I replace the dataset with all background information to find out the top 10 important behaviors or features in the dataset. Inspiring by the result of permutation feature importance, I decided to add past grades into the dataset and try to increase the difficulty. From Pass/ Fail binary classification, to 5 levels multi-label classification, actual score prediction in the end.

Detail of Dataset

The dataset used in this project comes from UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets/Student+Performance#> ). There are 33 columns in the original dataset. After cleansing the data, I keep 29 columns for exploring and training.

A close up of text on a black background

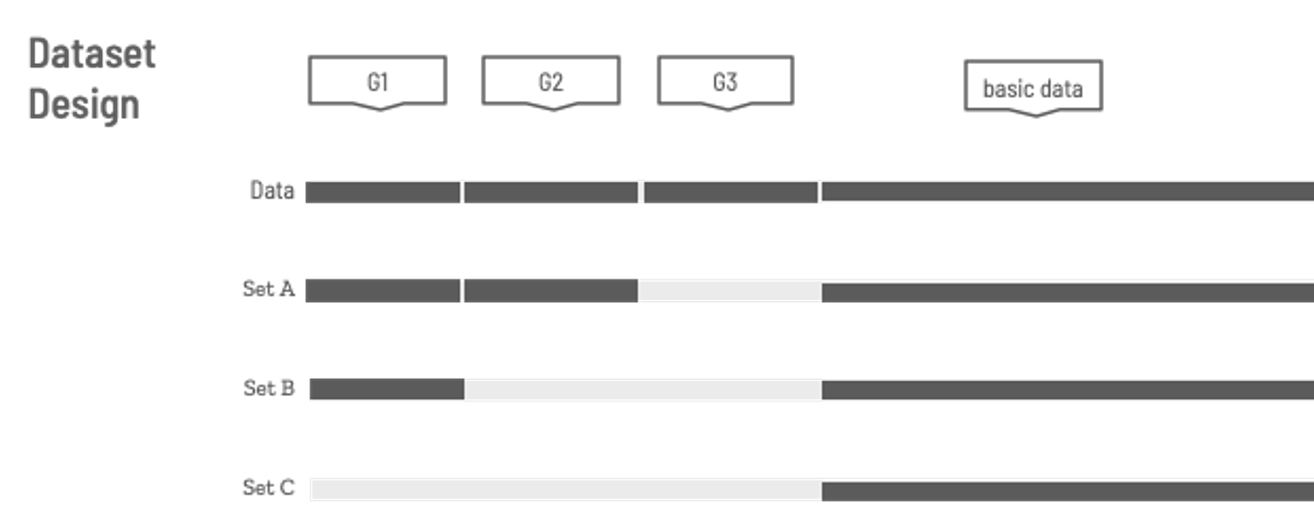
Description automatically generated

Because Portuguese high schools choose Trimester rather than Semester applied in the US, I just simply apply Trimester to a Semester pattern.

* G1 -> the final exam of the first semester
* G2 -> the midterm of the second semester
* G3 -> the final exam of the second semester (the Goal!)

Training and testing datasets

In this project, I separate the entire dataset into three different sets. These three datasets contain different data levels.



* Set A -> basic data + G1 + G2 (Contains most of the original dataset)
* Set B -> basic data + G1 (Contains less information than Set A)
* Set C -> basic data ONLY (Contains the least information)

Similar to the data X, I also choose three different levels of the data y.

A picture containing monitor, screen, television, sitting

Description automatically generated

* A\_output -> Pass/ Fail (The easiest one)
* B\_output -> 5 levels grade (Increasing the difficulty)
* C\_output -> Actual score (The most difficult one. Predicting the score in the final exam 'G3')

Workflow

1. Exploring data

A screenshot of a cell phone

Description automatically generated

Check the type in the dataset

A screenshot of a cell phone

Description automatically generated

1. Cleansing data

A screenshot of a cell phone

Description automatically generated

1. Preprocessing data

Since we cannot use string type in modeling and one-hot encoding, replace them with integer.

A screenshot of a cell phone

Description automatically generated

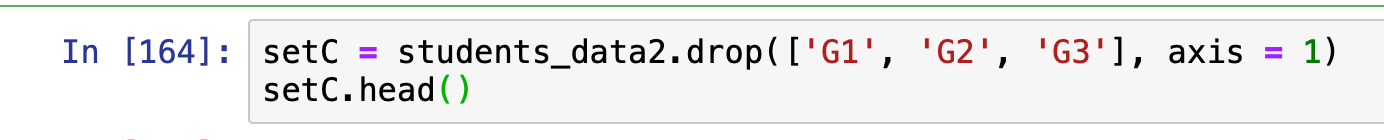
1. Creating different datasets & One-hot encoder

SetA, SetB ,SetC:



A screenshot of a cell phone

Description automatically generated



outputA, outputB, output:

A screenshot of a cell phone

Description automatically generated

A close up of a piece of paper

Description automatically generated

A close up of a piece of paper

Description automatically generated

1. Save to local

A screenshot of a cell phone

Description automatically generated



A screenshot of a cell phone

Description automatically generated

Choose data y and data x from local files, then combined them

A screenshot of a social media post

Description automatically generated

1. Set null hypothesis & alternative hypothesis

A screenshot of a cell phone

Description automatically generated

Using Binomial Distribution Formula to find the critical area

A screenshot of a cell phone

Description automatically generated

1. Upload to S3

A screenshot of a cell phone

Description automatically generated

1. Create Pytorch estimator & training

A screenshot of a cell phone

Description automatically generated

1. Deploy Pytorch estimator

A screenshot of a cell phone

Description automatically generated

1. Test model

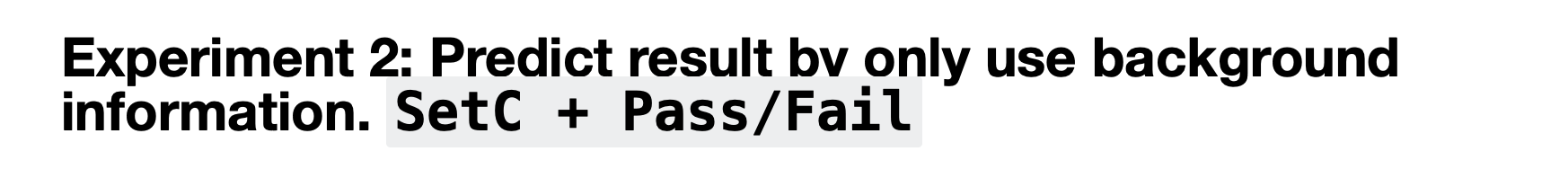
A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated





Choose data y and data x from local files, then combined them

A screenshot of a cell phone

Description automatically generated

1. Upload to S3

A screenshot of a cell phone

Description automatically generated

1. Create Pytorch estimator & training

A screenshot of a cell phone

Description automatically generated

1. Deploy Pytorch estimator

A screenshot of a cell phone

Description automatically generated

1. Test model

A screenshot of a social media post

Description automatically generated

1. Train a Random Forest model to calculate permutation feature importance

A screenshot of a cell phone

Description automatically generated

1. Test scikit-learn model

A screenshot of a cell phone

Description automatically generated

1. Calculate permutation feature importance & plot it

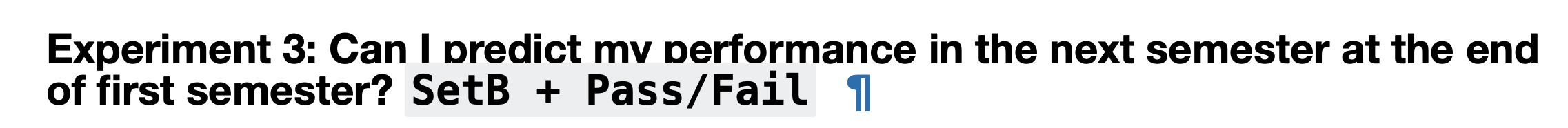
A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. The left four experiments have similarities workflow with these two.









Performance compared to benchmark models

|  |  |  |
| --- | --- | --- |
|  | Benchmark model (NN) | My model (NN) |
| Exp1: “bad” + Pass/Fail |  | 0.570 |
| Exp2: SetC + Pass/Fail | 0.663 ±0.0010 | 0.734, 0.709(RF) |
| Exp3: SetB + Pass/Fail | 0.813 ±0.0050 | 0.873 |
| Exp4: SetB + 5 Level | 0.498 ±0.0012 | 0.595 |
| Exp5: SetA + 5 Level | 0.603 ±0.0016 | 0.721 |
| Exp6: SetA+Actual Score | ±2.05 ±0.0020 | ±1.71 |

Reference

P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.

This is a demo project for me. Practicing Pytorch, scikit-learn, and AWS sagemaker. Thanks to the original authors! Hope you like this project :)