Zhenrong Lao

Professor Robinson

DSBA 6520

July 8, 2021

Project I—Use Case, Data Model and Projections

I. Dataset Use Case

Nowadays, there are so many resources available to students. For example, there are teachers there to help them in and out of class, their peers can give them a hand, tutors are available for parents to hire, and there are so many free online learning materials. However, not every student can do a good job in high school such as obtaining a high score on their exams even if all of them have chances to do it. In this project, the goal is to understand why they should perform great which still is not the case. To better analyze the influence of various factors on the student's performance, student data is collected with 8 variables described as follows:

1) gender: the gender of each student (male/female)

2) race: the race of each student (there are three groups: group A,group B,group C)

3) parental level of education: the highest educational qualification of any parent of each student

4) lunch_type: the type of lunch package selected for each student(standard/reduced)

5) test_prep: if the test preparation course was completed by the student or not

6) math_score: score in math (our target variable)

7) reading_score: score in reading

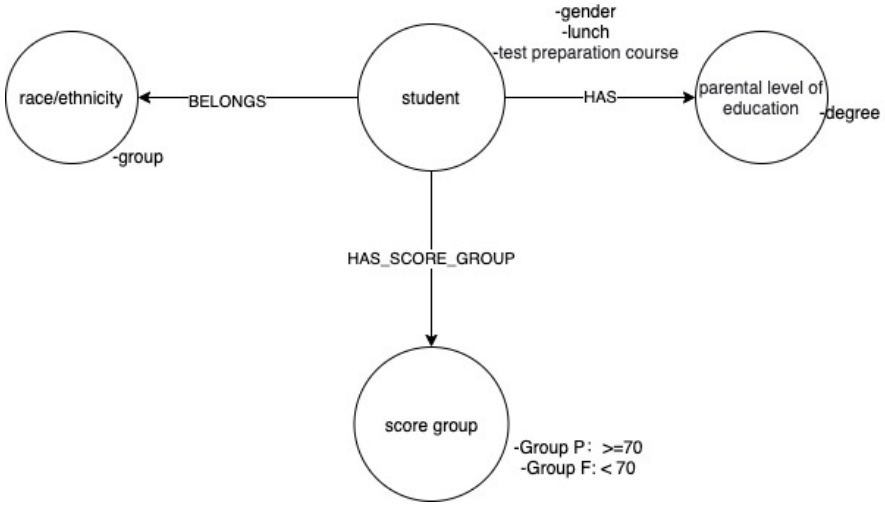
8) writing_score: score in writing

More specifically, several questions through graph analytics on this dataset will be answered:

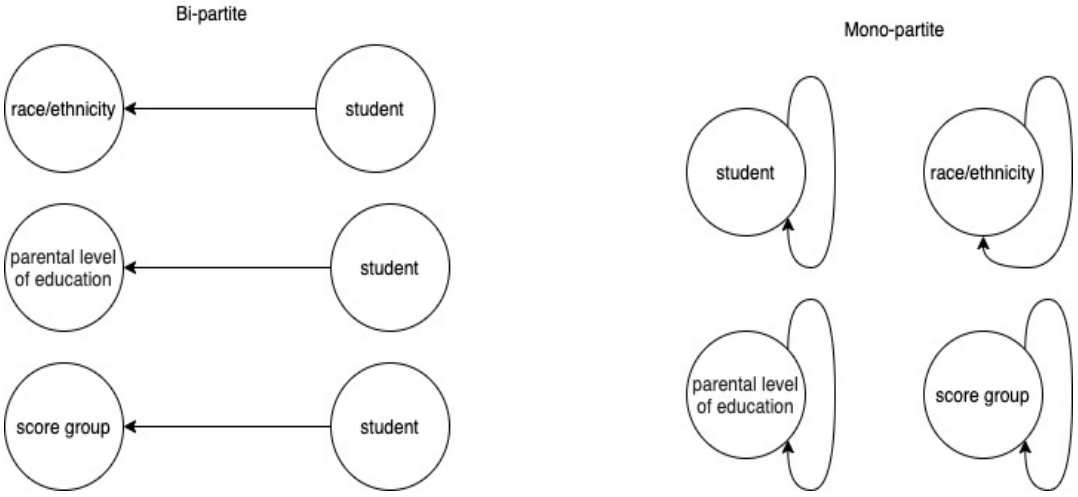
How many students pass the test for each test and how many students fail the test for each test?

Is there any similarity for students who pass or fail exams? Is test preparation enough for exam pass? What's the main driver of good test scores? Based on analysis, could we improve the students performance in the tests?

II. Graph Data Model



III. Graph Projections

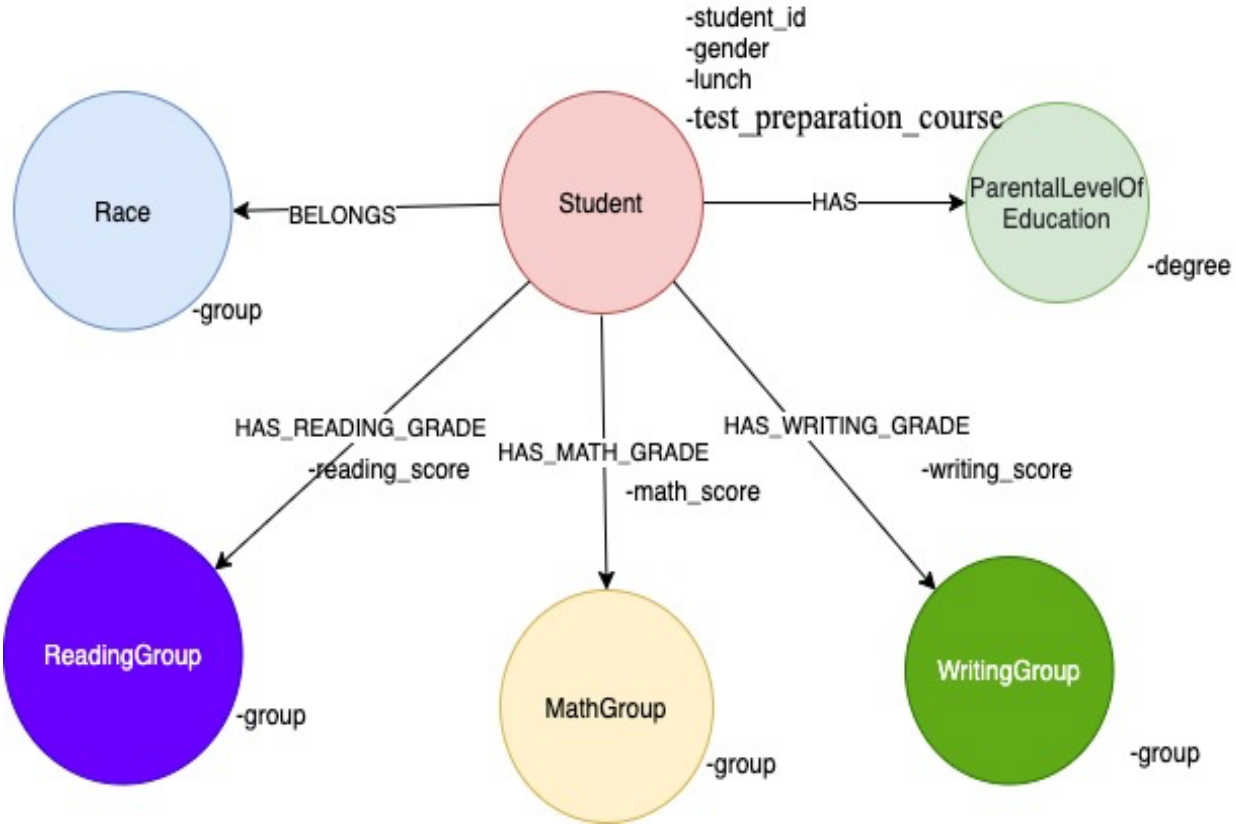


Project II—Graph DB setup and Application of Algorithms

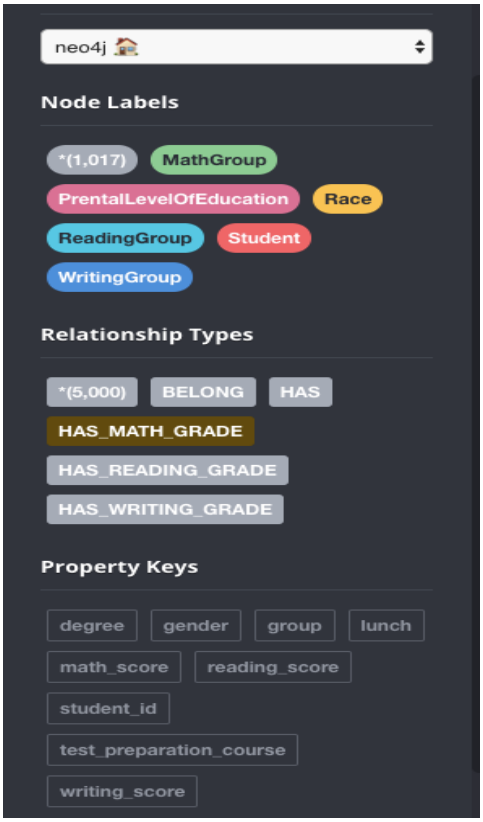
I. Updates to Graph Data Model

Firstly, for better visualization, colors to each node are added.

Secondly, in order to do further analysis, the node" score group" is divided into three nodes, and each course has one node.



II. Neo4J Database Setup



III. Cypher Queries

1. There are total 1000 students in the dataset. Out of them, there are 489 students who passed writing exam, there are 513 students who passed reading exam and there are 409 students who passed math exam.

Cypher Queries	Course	Total Student
MATCH (s: Student)-[: HAS_WRITING_GRADE]-(m:WritingGroup{group: 'True'}) RETURN count(s) as pass UNION MATCH (s: Student)-[:HAS_READING_GRADE]-(m:ReadingGroup{group: 'True'}) RETURN count(s) as pass UNION MATCH (s: Student)-[: HAS_MATH_GRADE]-(m:MathGroup{group: 'True'}) RETURN count(s) as pass	Writing	489
	Reading	513
	Math	409

2. There are 358 students who completed test preparation course. Out of them, almost 44% students passed all three exams. For 642 students who did not complete test preparation course, 174/642 only 27% of them passed all three exams. This indicates that test preparation is quite useful for students to prepare exams. 238/358 who passed writing exam completed test preparation course.

Cypher Queries	Name	Total Student
MATCH (s: Student{test_preparation_course:"completed"}) RETURN count(s)	Completed TP	358
UNION MATCH (s: Student{test_preparation_course:"completed"})-[:HAS_SCORE]-(m:ScoreGroup{math_score_group:'True'}) RETURN count(s)	math	176
UNION MATCH (s: Student{test_preparation_course:"completed"})-[:HAS_SCORE]-(m:ScoreGroup{writing_score_group:'True'}) RETURN count(s)	writing	238
UNION MATCH (s: Student{test_preparation_course:"completed"})-[:HAS_SCORE]-(m:ScoreGroup{reading_score_group:'True'}) RETURN count(s)	reading	234
UNION MATCH (s: Student{test_preparation_course:"completed"})-[:HAS_SCORE]-(m:ScoreGroup) WHERE m.writing_score_group='True' AND m.math_score_group='True' AND m.reading_score_group='True' RETURN count(s)	All pass	150

3. There are 325 female students who passed all exams, while only 164 male students who passed all exams. In general, female has higher passing rate than male.

Cypher Queries	Gender	Total Student
MATCH (s: Student{gender:"male"})-[:HAS_WRITING_GRADE]-(m:WritingGroup{group:'True'}) RETURN count(s) as total_writing_pass UNION MATCH (s: Student{gender:"female"})-[:HAS_WRITING_GRADE]-(m:WritingGroup{group:'True'}) RETURN count(s) as total_writing_pass	Male	164
	Female	325

III. Graph Algorithms

Here, a graph is created based on loaded dataset, with following code

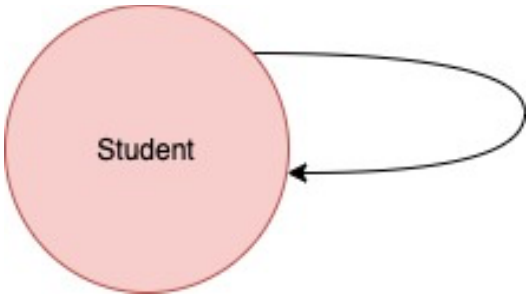
Algorithms	CALL gds.graph.create(
	'Graph_student',
	['Student','ReadingGroup','WritingGroup','MathGroup','Race','PrentalLevelOfEducation','Race'],
	{HAS_MATH_GRADE: {
	type: 'HAS_MATH_GRADE'},
	HAS_READING_GRADE: {
	type: 'HAS_MATH_GRADE'},
	HAS_WRIING_GRADE:{
	type:'HAS_WRITING_GRADE'},
	HAS:{type:'HAS'},
	BELONG:{
	type:'BELONG'} });

nodeProjection	{"WritingGroup": {"properties": {},"label": "WritingGroup"},"PrentalLevelOfEducation": {"properties": {},"label": "PrentalLevelOfEducation"},"MathGroup": {"properties": {},"label": "MathGroup"},"Race": {"properties": {},"label": "Race"},"ReadingGroup": {"properties": {},"label": "ReadingGroup"},"Student": {"properties": { },"label": "Student"}}

Relationshio Projection	{ "BELONG": {"orientation": "NATURAL","aggregation": "DEFAULT","type": "BELONG","properties": {}},"HAS_WRIING_GRADE": {"orientation": "NATURAL","aggregation": "DEFAULT","type":"HAS_WRITING_GRADE","properties": {}},"HAS": {"orientation": "NATURAL","aggregation": "DEFAULT","type": "HAS","properties": {}},"HAS_READING_GRADE": {"orientation": "NATURAL","aggregation": "DEFAULT","type": "HAS_MATH_GRADE","properties": {}},"HAS_MATH_GRADE": {"orientation": "NATURAL","aggregation": "DEFAULT", "type": "HAS_MATH_GRADE","properties":}}}
graphName	" Graph_student"
nodeCount	1017
Relationship Count	5000
createMillis	264

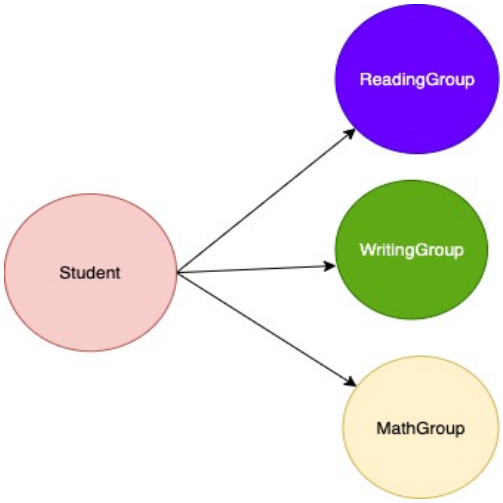
1. As a community detection algorithm, WCC is used to evaluate how groups of nodes are cluster. It can be considered as a preprocessing step for directed graphs, since it helps quickly identify disconnected groups. From componentID, we can easily know which group each node belong to. In general, node about student belong to the same group.

Algorithms	name	ComponentId
CALL gds.wcc.stream('Graph_student') YIELD nodeId, componentId RETURN gds.util.asNode(nodeId).student_id AS name, componentId ORDER BY componentId, name	1	0
	2	0
	3	0
	4	0



2. The PageRank algorithm measures the importance of each node within the graph, based on the number incoming relationships and the importance of the corresponding source nodes. The node of fail in math has highest score, which implies most of students fail in math exam. Therefore, math exam is hardest one among all exams.

Algorithms	Title	Score
CALL gds.pageRank.stream('Graph_student') YIELD nodeId, score RETURN gds.util.asNode(nodeId).group AS name, score ORDER BY score DESC, name LIMIT 4	Fail in Math	30.29105596542358
	Pass in Reading	21.009038877487182
	Fail in writing	13.180489194393157
	Pass in Writing	12.61948972940445



3. The Node Similarity algorithm compares a set of nodes, depending on which node they are connected to. Nodes with same neighbors are consider as similar. Top five pairs with highest similarity score are picked up. Those students usually have same results in all exams, are in the same race group and have the same parental education level.

Algorithms	Student 1	Student 2	Similarity
CALL gds.nodeSimilarity.stream('Graph_student') YIELD node1,node2, similarity RETURN gds.util.asNode(node1).student_id AS Student1, gds.util.asNode(node2).student_id AS Student2,similarity ORDER BY similarity DESCENDING, Student1,Student2;	1	117	1.0
	1	200	1.0
	10	104	1.0
	10	82	1.0
	10	249	1.0