Similar MATs

King's Certificate Group J

William Charlton, Jagdeep Dhemrait
Daniel Batchford, Muyang Chang

May. 23, 18

Abstract

We developed a Python-based program which groups MATs together based on similarity with competing MATs. Based on an algorithmic approach, we explain how we specifically planned to produce an optimised, efficient solution to our brief. We explain in detail a logistical solution to this task, as well as detailing individual roles, deadlines and challenges within this project. We also provide details of testing and iterative improvement, with specimen data provided.

1 Personal Introductions

This section introduces individual members of our group, detailing our interests and future aspirations, as well as the benefit of this program to higher education and later life.

1.1 Daniel Batchford

I am a highly motivated mathematician and computer scientist who enjoys collaborative group work - in particular, I enjoy leading a group. My interest in data science allows me to fully exploit the set brief - I will be able to fully apply myself in the statistics aspect of the project. An interest in computer science allows me to contribute to the coding of the program - I can help design the output computationally such that it meets the requirements set in the brief. Furthermore, my application in previous literacy demanding projects allows me to produce academic standard write-ups in a variety of tenses and formats. This will help throughout the publishing of the academic journal, as well as during feedback to the project mentor.

I hope to improve my communication, both in verbal feedback during conversations with the mentor and during presentations, as well as written communication through writing and publishing the journal. I also hope to improve research skills - researching government articles and providing the relevant citations ensures I can enter university with reliable and honest literacy abilities. Moving forward, I hope to study mechanical engineering in a Russell group university, allowing me to pursue a career in industry; I would like to work for an aeronautical company such as British Airways or Rolls Royce.

1.2 Jagdeep Dhemrait

My enthusiasm in mathematics spurs me to undertake challenging and invigorating tasks involving a large responsibility of time and effort. I work well in a team, as a collaborator and an innovator of new ideas and proposals. I have a keen interest in chess and have been playing for many years; this allows me to contribute by analyzing the design of the program, allowing me to propose changes to the program and guide my team through the initial planning phase. Additionally, my public speaking experience and qualifications will allow me to take a leading role during the presentation phase - I will be able to convey information clearly and effectively to a large audience. Furthermore, my experience in essay writing projects, such as Brilliant Club, will allow me to write academically to a high standard befitting of this project.

My goal is to improve my researching and presenting skills by practicing writing and presenting academically in a large group project. This will be a vital learning experience for university applications; this will also help in the foreseeable future in a job. After sixth form, I hope to study Mathematics at either Oxford or Cambridge, leading me to work in accountancy or economics after university.

1.3 William Charlton

I am studying physics, mathematics, and further mathematics to A level, alongside computing to AS level. I hope to become an aeronautical engineer - a profession where the skills required in this project will come in useful: data processing and iterative model design are exceptionally useful in computerized testing of airframes. While I have no prior experience with utilizing skills in the context of school analytics, I have previously run a workshop designed to foster iterative design skills in students from lower years in secondary school.

Having been taught computing for AS level, combined with prior experience coding in Python, I am responsible for the bulk of programming in the group. It is my responsibility to write the program that interprets the government statistics into a single database and write the functions that execute the algorithms we design upon that database. I hope that this project will give me experience at applying my skillset to real life challenges, outside of a classroom setting, and thus strengthen my hand when going on to study the qualifications necessary for my intended career.

1.4 Muyang Zhang

I am currently a Year 12 student at King's College London Maths School. I am studying Maths, Further Maths, Computer Science and Physics. Skills learned in further Maths, such as statistics and logic problems will allow me to help with tasks such as data collection. Moreover, I'm currently studying Python in computer science; I can apply the skills I have developed in computing to tasks required in the King's Certificate brief. My main role in the group is to analyze data from Multi-Academic Trusts; In accordance to my findings from my analysis, I must determine, with the group, the qualities that could group multiple MATs together in terms of their general attributes.

This is the first time I have worked in collaboration with a company, so for starters, I hope to learn how to correctly communicate and work with industry. Since this is a quite sophisticated project, there will be an emphasis on teamwork and collaboration; therefore, I hope to significantly improve my ability to communicate, not just within a team but with industry. Furthermore, upon completion, I will have it as a proof of achievement for both university applications and future jobs.

2 Project Introduction

This section introduces our school, as well as our project mentor. It also briefly introduces the BISS model and how we planned to utilize this model for our project.

2.1 School

King's College London Mathematics School, a small sixth form perched under a tower block in Lambeth, reflects the qualities of a well-rounded, able mathematician. Its values - curiosity, ambition, courage, tenacity and integrity are evident in all 140 of its students. Founded in 2013, it aims to teach mathematics to a wider demographic of children, at an above par standard. Students without the support in their previous school flocked to KCLMS in 2013, finishing with grades well above national average. 4 years later, it has been ranked as the top state school in Britain.

Aiming to provide an outstanding mathematics education leads the school to solely teach Mathematics, Further Mathematics, Physics and a choice of either Computer Science or Economics. Furthermore, the King's Certificate offers students a chance to work on a larger project as a group of 4, potentially in industry. This helps improve employability, speaking and teamwork skills, alongside with knowledge of how to collaborate in a literacy rich way with academics or industry professionals.

2.2 Project Mentor

Stephen Higgins, Senior Project Manager, Arbor

Stephen Higgins is a project manager at Arbor Education Partners. Leading teams on development for secondary school products, he manages the development and implementation of software to schools and MATs. Stephen Higgins has previously worked on the Aspire Award, in software development. Aspire aims to increase co-operation in extra-curricular activities in secondary schools. Higgins has also worked as a humanities teacher for over 5 years.

2.3 BISS Model

Schools are judged unfairly through lazy data analysis, consequently leading to multi academy trusts appearing lower in government tables than their respective true performance. Data holds a hidden value, dictating subtly the outcomes of major events. It is therefore unjust to rank schools hastily without consideration of hidden measures overseen by statistics. A true evaluation into a Mat's performance would allow schools to appear in government rankings in a more justified and concrete way.

Current models of MATs do not take into consideration the multitude of MAT setups under one umbrella term. MATs can range from a single school to over fifty. Operations in large academies are often more streamlined, through the synchronization of resources and the ability for the MAT to transfer students between schools for a more optimal standard of education. MAT models are dictated by many key factors, often overlooked by poor data analysis. Location, demographic, travel times and salaries, to name a few, play a delicate role in grouping MATs. A small change to variables leads to a large change in the grouping of an MAT – It is therefore ignorant to group MATs hastily without consideration into the subtle factor's behind them.

Arbor, a school specialized statistics company, do not currently have a system to group MATs. Their school grouping system has been used with clients of Arbor, during reports to the respective schools. It is easy to see how this grouping system can be extended to MATs – a larger client base will allow Arbor to expand their business, through a higher number of clients. The variety of statistics companies such as Arbor lead to a competitive market - therefore, it is important we produce a solution up to industry standard. Our solution could also have implications with MATs themselves; accurate statistics publishing to MATs will allow them to improve their education standard, benefiting the entire community. This is through simple comparison bar charts, which show a MAT and its performance against similar MATs.

Our solution aims to group MATs over several factors, publishing the reports into a file. Initially, we will calculate, through testing, real world observation and mathematical techniques, the factors that should dictate how MATs are grouped. We will have to decide on different weightings for variables, based on their importance. Importing the government published data into a computer program will allow us to view how MATs should be grouped. Coded in Python, the program will be designed, implemented and potentially enrolled into the company. A sufficient standard will see Arbor using our software solution every morning, before collating the data into reports for clients of Arbor. Attentive behavior will be needed in our team to ensure our program fits the requirements of the company; for example, it will have to be optimised, lightweight and executable on multiple operating systems. We will also have to ensure it produces a realistic grouping, as our solution could potentially have real world implications.

Overall, this project will allow us to better our understanding of industry practice, statistical methods and MATs, while simultaneously improving literacy, verbal and management skills. Furthermore, it will have a direct impact to society provided the solution is used, both for MATs and Arbor.

3 Literacy Review

3.1 Education A. Arbor Education / Home [Available from: https://arbor-education.com.]

This webpage is published by Arbor, our company mentor, it is intended to advertise the company's role as a company that provides analytical research to schools. As a source published by the subject of the information we will not use this source for anything but ascertaining the company's position in the market.

3.2 Hanushek EA, Woessmann L. How Much Do Educational Outcomes Matter in OECD Countries? NBER working paper series no w16515; Cambridge, Mass: National Bureau of Economic Research; 2010. p. 67.

This source evaluates the usefulness of education in OECD countries, using reliable data to draw conclusions about the growth of economic variables compared to the standard of education. Eric A Hanushek is an established economist who specialises in economics in education. His publishing, of over 60 pages, contains well collated data collected in an unbiased fashion, with later conclusions being explored in great depth. The authority and experience of this author leads me to believe that the author has educated and unbiased work, which can be trusted. However, limitations of data collection and statistical analysis can lead to potential deviations from real world happenings. Eric concludes, through great explanation, that increasing education funding and refinement gives a higher net gain to the country in question. This source is intended for academics in economics, but it can be interpreted for our project. It is useful for justification of why research into MATs is worth doing – It is clear from this article that helping schools through statistics research companies such as Arbor has a direct, beneficial impact on the economics and GDP of said country. While it does not form the backbone of our research, it is useful not just for justification for the project, but for an insight into how vast amounts of data is collected. It also displays how data should be presented in a professional manner.

3.3 Analysis IaFD. Multi-academy trust performance measures: 2015 to 2016. In: Education Df, editor.: UK Government; 2017. p. 36.

This data source, published by the UK government, evaluates from collected data, the differentiating factors of multi-academy trusts. It is completely unbiased and gives a literal description of how MATs differ from each other. These claims are backed up with a large database of raw data. This Is the only official source for MATs – it is therefore not possible to fairly compare this source to another resource. This database will contribute highly to our King's certificate project, as it is the only complete source of data from a first party source. We will be able to use this data to evaluate and draw our own conclusions, potentially in a different fashion to the government.

3.4 Multi-academy trusts: Good practice guidance and expectations for growth 2016.

In this report, the government sets out what they know about the characteristics of successful academy trusts and the barriers that they will need to overcome in order to secure their ongoing success. The government's Department of Education has collected data from 67 established MATs through a survey conducted in 2015, which was followed by discussions with schools and trusts. The main aim of this research was to provide a mainframe of how successful MATs act, allowing others to follow in example, enhancing the standard of education in the UK. The research focuses on what regional school's commissioners will look for when deciding whether to approve academy arrangements. This article is useful to our research topic as the government demonstrates how successful MATs behave, which can be used in designing the variables used in the algorithm of our project.

3.5 Great Britain. Parliament. House of Commons. Education C. Multi-academy trusts: Seventh Report of Session 2016-17-2017.

This source is written by the Education Select Committee of the British Parliament. As a cross-party committee its findings are unlikely to be biased, politically, at least. The report evaluates the roll-out of Academies and Multi Academy Trusts. It draws the conclusion that it has been by and large successful, but the government pursued it too overzealously. This conclusion in itself is not relevant to our research, however, along the way the report references and includes useful data and information about MATs that is useful. The intended audience is British MPs, with parliamentary reports existing to inform voting, but this report along with all the others is used by the press when reporting on the Government.

4 Methodology

This section gathers ideas about our research throughout the project timeline. It also explains team roles within our group, as well as internally/externally set deadlines and specific plans for data collection and analysis.

4.1 Initial Research and Ideas

When programming started, the basis of the program was clear: it had to be flexible, with a high throughput. A company like Arbor needs to be able to adapt and improve, and this cannot be done using a program with a hard-coded algorithm or data. Nor can it wait hours for the results to be submitted, meaning that the program had to process data quickly and efficiently. As a result, the requirements set were that the program should be able to integrate new datasets, and hot-swap algorithms, as well as perform tests quickly. We also initially realised that the program needs to be able to run on multiple operating systems effectively, without a significant chance of crashing, as this could result in a loss of data and time for Arbor.

4.2 Team Roles and Responsibilities

William is the main programmer in our group; he oversees development of the program in Python. William also helps contribute to the written aspects of the project, as he has a vast knowledge of specifics of the program which is useful in writing tasks.

Daniel is the team leader and organises meetings and sets deadlines for the group. He communicates with members of the group helping them meet deadlines and complete tasks. Daniel also contributes to the written aspects of the program as he has oversight on timeframes and current group affairs.

Jagdeep is the program tester and algorithm designer. He takes the end-user program from William and tests it to make sure that the program works and meets the needs or Arbor. He then provides feedback to William, who can iterate and improve the program.

Jagdeep also contributes to the written aspects of the project as he has created and revised a model for the algorithm used in the model.

Muyang is the main researcher in the group. He has researched in depth the background of Arbor and has a detailed knowledge of MATs from his research. Muyang helps William and Jagdeep with designing the program and algorithm, as he knows in detail the important features of MATs.

4.3 Project Timeframes

Our group organised weekly meetings, on varying days of the week, based on the times available for members. In these meetings, we set individual tasks for each member – we also discussed research ideas and software improvements. These meetings allowed us to keep up to date weekly on current affairs and tasks needed to meet greater deadlines (detailed below).

Our group decided a Gantt chart was suitable for the organisation of deadlines in the project. Our chart outlines brief, flexible, long-term deadlines for software development stages, alongside journal submission deadlines. Moreover, the chart includes individual deadlines for each group member, tailored to their strengths and abilities. For example, we placed a deadline towards the end of January for the initial software prototype to be compiled. While these deadlines are not strict, it is important we meet these deadlines to keep up rapid progression within the project. The deadlines also act as an incentive to publish work, allowing us to progress efficiently through stages of the project brief.

Specifically, we aimed to complete the first version of software by January 24th. We met this deadline, which allowed us to display the draft software to our project mentor during the second meeting in February. In the coming months, we aim to produce finalised software, through multiple iterative improvements to the pre-existing software. We placed a deadline on porting the software to the Windows operating system on the 15th of April. The issue of incompatibility on Windows halted the project progress significantly – currently, the lack of working Windows software places a bottleneck on the development and tweaking stage of our project's timeline. The deadline allowed for an in-depth analysis on errors in Windows software -porting has proven to be harder than initially anticipated.

We placed a second deadline on the 15th of May for completion of the software – we expect to have finalised, optimised and streamlined the software by this date. We will also be ready for submission of the software to Arbor on this date, given that previous deadlines are met to a reasonable standard. We placed a final deadline on 23th May to finalise the journal. This deadline should result in us being able to submit a high standard journal article, allowing us to move onto the preparation of the upcoming presentation to Arbor.

4.4 Data Collection / Design / Analysis Plans

We planned to produce one algorithm for the default analysis and categorising of MATs, which carries out the whole process imperatively. However, if the user does have specific standards for MAT categorising, we planned to include an alternate algorithm allowing the user to customise variables, tweaking algorithmic models to produce desired results. We planned to use a weighting system to group MATS, however we require further research regarding implementation of this system. Within the main algorithm we have produced an Import program, as well as a Testing program. The import program takes in the range of MATs that need to be analysed, collecting the necessary data from a database. This process takes a substantial period of time, but only needs to be run once for each update cycle. The testing program calls in the database created by the Import program, carrying out the analysis and categorising of the data; this is relatively quick and generates a group of MATs that the program has concluded as similar. We planned to collect data by producing an automated algorithm that results in excel or json files, storing them in a public filesystem shared between group members.

We will first briefly go over the rationale behind the algorithm individually and evaluate their contributions to similar MATs. This should give us a rough idea about the capability of our program. After evaluating the previous results, we then carried out a test on the fluency of our prototype for the actual program model. This was specifically aimed at checking whether the program actually ran the way we expected the prototype to function, or even to test out whether the program interprets all the functions or not. Actual MAT data is not essential for this test, but it's better to use real data, and it should not be complicated. Once we completed the previous steps, we collected actual MAT data from specifically chosen MATs that we used to evaluate how robust our program is, the results were evaluated based on our hypotheses and expected outcomes.

The testing program iterated through all the possible permutations of the variables a-e in the above algorithm, using the values 0-3 inclusive. This is 1024 permutations - to test each of these permutations we split a MAT into two and picked the permutations that rank the second section of the MAT in the top 5 most similar to the first section of the MAT. From these we looked at the algorithms that give the MAT the highest percentage similarity with special consideration for those that rank the second section of the MAT the highest. From this process, we discerned the most fitting weights for these variables. We only used the weight function with numerical variables as we considered qualitative analysis to be the most appropriate when used for a MAT by MAT basis.

5 Findings / Results / Improvements

In this section, we address the initial findings our group made for a prototype that we produced for our algorithm. We then analyse these findings and explore how they may be beneficial to us for improving our program in the future.

We used a separate algorithm that uses trial and error to improve our existing algorithm through permutations of testing. We split a large MAT in our database into two groups using a random number generator that generated variables of 1 or 2. For example, we used the random number generator to split the MAT, United Learning Trust, into two groups of 20 schools and 17 schools. Following this, we tested our algorithm to try and match these two groups together in our database. Our testing algorithm procedurally assigned variables to the weightings for the factors in our main algorithm and calculated the similarity of the groups. Also, this algorithm created a list of the most similar MATs and has calculated the position of the other group in this list. We repeated this test for different procedurally generated weighting systems and we have split up three different MATs to test our algorithm. We have compiled a table[(2)] which shows the weightings that the testing algorithm has assigned for the MAT United Learning Trust, the position of the second group in terms of similarity and the percentage similarity that the two groups exhibited.

Initially, we found that the defining factor of a MAT is the distribution of statutory high and low school cut off ages. Similar MATs seemed to be grouped confidently by the algorithm once permutations were run - therefore, we can confidently conclude that this is the defining factor for MAT groupings. Next, we found average age of students in schools in a MAT, followed by the number of schools in the MAT of high importance. Geographical dispersion of schools around an area in a MAT was also key in our grouping, as well as the median house price in the county of each school. This implies that MATs in areas of similar affluence are likely to be similar with each other.

To improve our current algorithm and findings, we need to run the algorithm for more MATs, as we only currently have findings for three MATS. We are currently limited by publishing of data, as MAT data currently needs formatting manually. Automation is required before the final software is submitted – this can be done relatively easily in future iterations of the algorithm.

6 Final Results and Discussion

The outcome of our project, which is to complete the task of comparing similar MATs, is a GUI based Python programming model composed of algorithms, that takes MAT datasets as the input, outputting both graphical and textual representation of the comparison the algorithms made to the similar MATs. The outputs are quotable, considering each stage of the algorithm will be able to be collected. We acknowledged that MAT's CEOs are "time-poor" and expect time-efficient, professional, evidence-backed conclusions. When the algorithms were designed, they aimed to achieve the above requirements for its outputs; namely the outputs are strictly limited to 4, representing each of the field of variables for the comparison. We are able to manipulate these variables as an array listing to improve its effectiveness of interpretation.

Specific Algorithm Referenced:

a	`MAT Trust Size`
b	`Geographic RMSD Weighted Dissimilarity`
c	`Houseprice AVG Weighted Dissimilarity`
d	`Statuatory Low Age AVG Weighted Dissimilarity`
e	`Statuatory High Age AVG Weighted Dissimilarity`

7 Conclusions

Our program worked as expected – we can therefore conclude that the project was a success. Our program compiles successfully, in a reasonable amount of time. We believe that Arbor will be pleased with the quality of the program, which now includes a GUI and multi-threading.

From this project, we have learnt a significant amount regarding programming – this includes how to use GUI's and link multiple Python scripts together into a sufficient program. We also learned how to construct a solution to a problem as a team – this involved organizing work between group members, as well as meeting internal deadlines within our group. We have learned how to answer a broad question to a sufficient standard; moreover, this question is tackled alongside industry – we have therefore learned how to address and meet professionally with industry. Communication skills have been improved, as well as organizational skills within agenda meeting forms.

In broader life, this program can be ported to other companies. While designed for Arbor, this program can be redesigned for other clients if required. The detailed statistics knowledge learned from this project will allow us to pursue similar projects in the future, potentially covering fields such as machine learning through neural networks. It is not a significant stretch to see how our group can go on to more challenging software development projects with the knowledge learned from this program. Furthermore, we will be confident when communicating with industry on future projects, as we have the experience from this project to address an industry professional under time constraints.

8 Personal Appendix

8.1 William Charlton

I learned much about statistical computing using Python. Specifically, I gained skills using a variety of database interfaces, asynchronous computation, and multiprocessing. I gained these skills working as the group's programmer, as I coded the computer program, and devised tests for algorithms in consultation with other members of the group. These skills are important to me, as I wish to pursue a future in engineering - a field similar to statistics, in that Python is commonly used. Python's limitations lead to it not scaling well with multi-core systems due to the global interpreter lock (GIL)[5]. Multiprocessing bypasses the GIL, and, in a world with 28 core server CPUs, this skill is incredibly significant.

8.2 Daniel Batchford

This project has significantly impacted my outlook on industry standards. The degree of organization and communication behind meetings has allowed me to fully grasp the level of professionalism behind UK industry. My role in this project as a coder has allowed me to learn Python syntax – going into Year 12, I had no knowledge of Python. I can confidently admit that this project has allowed me to code at an acceptable level, which will allow me to push for top grades in AS Computer Science. I can also apply this knowledge into industry fields such as software development in the future. My Python knowledge can be disclosed on a CV, which will help me with applications for university going forward.

I have learnt how to meet deadlines effectively, as well as how to work well within a team. Leading our group has allowed me to challenge ideas and suggest improvements, teaching me valuable people management skills.

8.3 Muyang Zhang

My main role within the whole project is research, and at times I assisted the programmers Daniel and William with minor programming and programming related tasks. Through most of our tasks, the group worked independently on our distributed tasks according to the plan designed beforehand, mainly only sharing information that needs to be shared; the team communicated extremely efficiently for the short durations when the group needed to gather and present data - this improved my communication skills which I admittedly lacked at the start of the project. Communication skills are very important, both in and out of work. I have also improved my teamwork skill, which is undisputedly essential for my career in university and future work. Also, the field of MATs and its related data is a previously unexplored topic for me, so as well as being a challenging research task for me, I have also developed a detailed knowledge base on the topic. This includes government policies on MATs. My research ability is polished by the challenging nature of this research task - this is going to help me with my ability to contribute in a group project later in life. Since my work also included assisting with minor programming tasks for William and Daniel, I have learned and explored more about Python. The programming involved in this project was challenging at times and I have significantly explored the range of capabilities of Python. I value what I explored in Python through William's narrations and explanations. If I end up working with computer systems, the knowledge I've gained will be useful in future career aspirations.

8.4 Jagdeep Dhemrait

Through the King's Certificate project, I learnt how to write and efficiently read academic articles. As the main article writer, I learnt vital skills in referencing and crafting well-structured essays.

Furthermore, I improved my teamwork skills by working constructively and effectively with others to produce a solution to a problem at an exemplary standard. These skills are essential for me to develop in the future as they will enable me to perform exceptionally when I work in a team in the future. Also, the progression of my writing skills will lead to me produce clear and persuasive articles and reports, allowing me to excel in the world of economics which is predominantly theory and essay based.

9 Data Appendix

This appendix contains data from our initial program outputs, as well as our Github repository and the plaintext Python code used in the program.

[1] Github Code Repository

This repository hosts the program online for access by our group and Arbor:

https://github.com/willwill2will54/MATSimCheck

[2] Table 1

Mat used – United Learning Trust

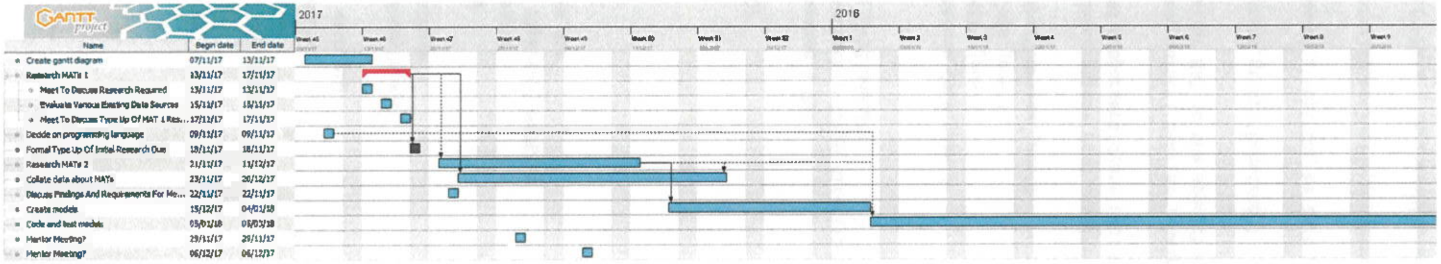
Algorithm Variables:

a	MAT Trust Size
b	Geographic RMSD Weighted Dissimilarity
c	Houseprice AVG Weighted Dissimilarity
d	Statuatory Low Age AVG Weighted Dissimilarity
e	Statuatory High Age AVG Weighted Dissimilarity

Output:

a	b	c	d	e	Position	Score
2	0	2	2	2	5	93%
2	2	1	2	3	5	93%
2	2	1	3	3	5	93%
3	1	3	3	3	4	92%
1	0	1	1	2	5	92%
1	2	1	3	2	5	92%
2	2	2	3	3	5	92%
2	3	1	2	3	5	92%
2	3	1	3	3	5	92%
3	1	2	3	3	5	92%
3	2	2	3	3	5	92%
2	0	2	3	2	4	91%
2	0	3	3	2	5	91%
2	1	2	2	2	5	91%
2	1	2	3	2	5	91%
2	2	1	2	2	5	91%
2	2	1	3	2	5	91%
2	2	2	3	2	5	91%
2	3	0	3	2	5	91%
2	3	1	3	2	5	91%
1	1	1	3	1	5	90%
1	1	1	2	1	5	89%
1	2	0	2	1	5	89%
1	2	0	1	1	5	88%

[3] Gantt Chart used for the project:



Name	Begin date	End date
Create gantt diagram	07/11/17	13/11/17
Research MATs 1	13/11/17	17/11/17
Meet To Discuss Research Required	13/11/17	13/11/17
Evaluate Various Existing Data Sources	15/11/17	15/11/17
Meet To Discuss Type Up Of MAT 1 Res...	17/11/17	17/11/17
Decide on programming language	09/11/17	09/11/17
Formal Type Up Of Initial Research Due	18/11/17	18/11/17
Research MATs 2	21/11/17	11/12/17
Collate data about MATs	23/11/17	20/12/17
Discuss Findings And Requirements For Me...	22/11/17	22/11/17
Create models	15/12/17	04/01/18
Code and test models	05/01/18	05/03/18
Mentor Meeting?	29/11/17	29/11/17
Mentor Meeting?	06/12/17	06/12/17

[4] Code Plaintext

This is the plaintext code used in our program.

Main.py:

```
# Python modules, built-in
import sys
import random
import csv
import os
import itertools
from shlex import split as shlexsplit
from itertools import product
from collections import Counter
# Python modules, install with pip
from tinydb import TinyDB
from gooey import Gooey, GooeyParser
# My stuff, included in package
import defaults as defs
import lib.messages as msg
from lib.testing import tester
from lib.importing import importer, MATList
from lib.misc import ensure_dir

_1 = sys.argv[0]
_2 = os.path.dirname(_1)
if _2 is not '':
    os.chdir(_2)

sys.setrecursionlimit(1500)

# Gooey automatically generates a GUI
@Gooey(program_name='MATSimCheck', image_dir='lib/img', monospace_display=True, p
rogress_regex=r"^This is (\d+)% done$")
def main():
    # Open the databases
    noncore = TinyDB('./dbs/non_Core.json')
    MATs = TinyDB('./dbs/MATS.json')
    core = TinyDB('./dbs/Core.json')
    counties = TinyDB('./dbs/Counties.json')

    parser = GooeyParser(description='Find similar MATs.')
    subparsers = parser.add_subparsers(dest='action')

    similar = subparsers.add_parser('Similar', help='''\
        Finds the most similar MAT to the specified MATs''')

    req = similar.add_argument_group('Required')

    can = similar.add_argument_group('Can be left alone')

    whichmats = req.add_mutually_exclusive_group(required=True)

    whichmats.add_argument('-MATs', choices=MATList(), nargs='*', widget='Listbox
',
                           help='Specifies MATs (hold cmd/ctrl to select multiple
)')

    whichmats.add_argument('-All', action='store_true',
                           help='Analyses all MATs')

    can.add_argument('--algorithm', '-a', default='defaults', nargs='*', help='''
\
        Configure the selection algorithm (optional)''',
                     gooey_options={'validator': {'test': "(user_input == 'defaul
ts') or (all(b in ('avg', 'rmsd', 'med', 'rng', 'mode', 'size') and (c in ['wgt',
'is', 'isnot']\
 or c.endswith('gets')) for b, c in zip(user_input.split()[1::4], user_input.spli
t()[2::4])) and len(user_input))",
                                                   'message': 'That is not a valid
algorithm. See documentation.'}})

    subparsers.add_parser('Display', help='''\
        Displays datatables currently stored''')

    subparsers.add_parser('Purge', help='''\
            Purges the compiled databases.''')

    test = subparsers.add_parser('Test', help='''\
        Runs Testing Utililty''')

    can.add_argument('--multi', '-m', type=int, default=10, metavar='x', help='''
\
        Displays x most similar MATs''')
    test.add_argument('--plural', '-p', type=int, default=20, metavar='x', help='
''\
Splits this many of the MATs into 2.''')

    args = parser.parse_args()

    if args.action == 'Purge':
        msg.PURGE()
        for x in [core, noncore, counties, MATs]:
            x.purge()
        MATs.purge_tables()
        msg.DONE()

    if args.action == 'Display':
        for x in MATs.tables():
            if not x == '_default':
                print(x.replace('|', ' '), flush=True)

    def doit(algorithm, tested, num, testing=False):
        if algorithm in (None, ['defaults', ], 'defaults'):
            algorithm = defs.algorithm
        if tested is not None:
            importkeys = []
            for a, b in zip(algorithm[:-3:4], algorithm[1:-2:4]):
                importkeys += [a, b]
            table = importer(importkeys, testing=testing)[0]
            results = []
            for x in tested:
                result = tester(x, table, algorithm=algorithm, number=num, testin
g=testing)
                print('{} is most similar to {}'.format(x, ' then '.join(result[0
])), flush=True)
                results.append((x, result[1]))
            if testing:
                return results
            else:
                with open('result.csv', 'w', encoding='utf-16') as resultsfile:
                    print('Writing results to file...', flush=True)
                    chain = itertools.chain.from_iterable
                    fields = sorted(list(chain(['Average ' + x, 'Subject ' + x] f
or x in defs.ProgressScoreHeaders)))
                    writer = csv.DictWriter(resultsfile, fieldnames=['MAT', ] + f
ields)
                    writer.writeheader()
                    writer.writerows([{**{'MAT': x[0]}, **x[1][0], **x[1][1]} for
x in results])
                    msg.DONE()

    def testprep(matnum):
        for encoding in ['utf-8', 'utf-16', 'Windows-1252']:
            try:
                openfile = open(defs.CoreDirectory + '/Core.csv', encoding=encodi
ng)
                openfile.read()
                openfile.close()
                with open(defs.CoreDirectory + '/Core.csv', newline='', encoding=
encoding) as donor:
                    raw = csv.DictReader(donor, delimiter=',')
                    dicts = [dict(row) for row in raw]
                    break
            except UnicodeError:
                pass
        MATs = list(dicts)
        MATnames = tuple(x[defs.MatNameKey] for x in dicts)
        ValidMATS = [x for x, y in Counter(MATnames).items() if y >= 3]
        ChangeMATS = random.sample(ValidMATS, matnum)
        retChangeMATS = {}

        def MATtransform(x):
            tobe = x[defs.MatNameKey] in ChangeMATS
            mustbe = x[defs.MatNameKey] not in retChangeMATS.keys() or not retCha
ngeMATS[x[defs.MatNameKey]][0]
            mustbe2 = x[defs.MatNameKey] not in retChangeMATS.keys() or retChange
MATS[x[defs.MatNameKey]][1] < 2
            if tobe and ((bool(random.getrandbits(1)) and not mustbe2) or mustbe)
:
                if x[defs.MatNameKey] in retChangeMATS:
                    retChangeMATS[x[defs.MatNameKey]][0] = True
                else:
                    retChangeMATS[x[defs.MatNameKey]] = [False, 0]
                x[defs.MatNameKey] += '1'
            elif tobe:
                retChangeMATS[x[defs.MatNameKey]][1] += 1
                x[defs.MatNameKey] += '2'
            return x
        random.shuffle(MATs)
        NewDicts = [MATtransform(x) for x in MATs]
        all_keys = set().union(*(d.keys() for d in NewDicts))
        ensure_dir('./special/test/core.csv')
        with open('./special/test/core.csv', mode='w', encoding='utf-16') as subj
ect:
            writer = csv.DictWriter(subject, fieldnames=list(all_keys))
            writer.writeheader()
            writer.writerows(NewDicts)
        return list(x + '1' for x in retChangeMATS.keys())

    def experimentalfunc():
        testeds = testprep(args.plural)
        go = 0
        ensure_dir('special/testresults/result.csv')
        with open('special/testresults/result.csv', 'w', encoding='utf-16') as re
sultsfile:
            writer = csv.DictWriter(resultsfile, fieldnames=['a', 'b', 'c', 'd',
'e', 'position', 'score', 'MAT'])
            writer.writeheader()
            dictstobewritten = []
            algorithm = defs.TestAlgorithmMaker([list(x)[0] for x in defs.TestAlg
orithmVariables])
            importkeys = []
            for a, b in zip(algorithm[:-3:4], algorithm[1:-2:4]):
                importkeys += [a, b]
            table = importer(importkeys, testing=True)
            if table[1]:
                MATs.purge_table(table[0])
            for var in product(*defs.TestAlgorithmVariables):
                go += 1
                algorithm = defs.TestAlgorithmMaker(var)
                results = doit(algorithm, testeds, 1, testing=True)
                msg.TRY(go)
                print(algorithm, flush=True)
                for resulthing in results:
                    done = False
                    for pos, thing in enumerate(resulthing[1]):
                        if thing[0].startswith(resulthing[0][:-1]) and thing[0].e
ndswith('2'):
                            dicttobewritten = {'a': var[0], 'b': var[1], 'c': var
[2], 'd': var[3], 'e': var[4],
                                               'position': pos, 'score': thing[1]
, 'MAT': thing[0]}
                            done = True
                            break
                    if done:
                        dictstobewritten.append(dicttobewritten)
            writer.writerows(dictstobewritten)

        print('\a')

    def normfunc():
        if args.All:
                args.MATs = MATList()
        elif type(args.algorithm) == str:
            args.algorithm = shlexsplit(args.algorithm)
        doit(args.algorithm, args.MATs, args.multi)

    if args.action != 'Purge':
        if args.action == 'Test':
            experimentalfunc()
        elif args.action == 'Similar':
            normfunc()


main()
```

Defaults.py:

```
algorithm = [
    'trust', 'size', 'wgt', '1',
    'geo', 'rmsd', 'wgt', '1',
    'houseprice', 'avg', 'wgt', '1',
    'StatutoryLowAge', 'avg', 'wgt', '1',
    'StatutoryLowAge', 'rmsd', 'wgt', '1',
    'StatutoryHighAge', 'avg', 'wgt', '1',
    'StatutoryHighAge', 'rmsd', 'wgt', '1']
CoreDirectory = './Core'
NonCoreDirectory = './non_Core'
MatNameKey = 'Trusts (name)'
# Heading of the postcode field in the data
PostCodeKey = 'Postcode'
# URL of the postcodes.io API
ApiURL = 'http://api.postcodes.io/postcodes/'


TestAlgorithmVariables = (range(4), ) * 5


def TestAlgorithmMaker(variables):
    algorithm = [
        'trust', 'size', 'wgt', str(variables[0]),
        'geo', 'rmsd', 'wgt', str(variables[1]),
        'houseprice', 'avg', 'wgt', str(variables[2]),
        'StatutoryLowAge', 'avg', 'wgt', str(variables[3]),
        'StatutoryLowAge', 'rmsd', 'wgt', str(variables[4]),
        'StatutoryHighAge', 'avg', 'wgt', str(variables[3]),
        'StatutoryHighAge', 'rmsd', 'wgt', str(variables[4])]
    return algorithm


ProgressScoreHeaders = ['School level reading progress score', 'School level writ
ing progress score',
                        'School level maths progress score', 'School level progre
ss 8 score']
```

[5] Chetan Giridhar. Understanding Python GIL. Available from: https://callhub.io/understanding-python-gil/. [Accessed 23-05-18]