Machine Learning Analysis of Student Achievements in England

Contents

Introduction

In England, student achievements are a key measure in determining both the funding and quality of education for further education (FE) providers (Ofsted, 2024; ESFA, 2024). This document intends to explore factors influencing student achievements across FE institutions in England by analysing its patterns and trends. To this end, the insights from the associated Jupyter Notebook, 'Machine Learning Analysis of Student Achievements in England,' are utilised.

Data Retrieval

The student achievements dataset was obtained from the GOV.UK (2024) education statistics data catalogue and is available under the Open Government License (The National Archives, 2024). From here the 'fes-geography-population-202324-q2.csv' file was downloaded and imported into a Jupyter Notebook as DataFrame 'fes_df'. Additionally, a 'data-guidance.txt' file was provided, containing useful information about the dataset. The dataset included the following attributes:

1. time_period: The academic year period.
2. time_identifier: The type of time identifier used.
3. geographic_level: The type of geographic breakdown used.
4. country_code: Country identifier.
5. country_name: Country name.
6. region_code: Region identifier.
7. region_name: Region name.
8. new_la_code: New Local Authority identifier.
9. old_la_code: Old Local Authority identifier.
10. la_name: Local Authority name.
11. lad code: Local Authority District identifier.
12. lad_name: Local Authority District name.
13. pcon_code: Parliamentary Constituency identifier.
14. pcon_name: Parliamentary Constituency name.
15. english_devolved_area_code: English Devolved Area identifier.
16. english_devolved_area_name: English Devolved Area name.
17. local_enterprise_partnership_code: Local Enterprise Partnership identifier.
18. local_enterprise_partnership_name: Local Enterprise Partnership name.
19. lsip_code: Local Skills Partnerships identifier.
20. lsip_name: Local Skills Partnerships name.
21. provision_type: The education provision type.
22. level_or_type: The level or type of education.
23. age_summary: The age band of students.
24. starts: The number of enrolments starts for apprenticeships only.
25. participation: The number of students undertaking education.
26. achievements: The number of students who achieved their course.
27. population_estimate: The estimated population of the 'geographic_level'.

28. starts_rate_per_100000_population: starts per 100000 of population.
29. participation_rate_per_100000_population: participation per 100000 of population.
30. achievements_rate_per_100000_population: achievements per 100000 of population.

Data Cleaning

Several data cleaning steps were performed on ‘fes_df’ prior to analysis and machine learning. The shape of the DataFrame was verified against ‘fes-geography-population-202324-q2.csv’ and found to match correctly. Several unnecessary columns were then dropped from ‘fes_df’, as explained by the code comments in figure 1.

Figure 1, dropping unneeded attributes from ‘fes_df’

```
fes_df = fes_df.drop(
    columns=[
        "time_identifier",  # Only value is 'Academic year'
        "country_code",  # Only country is England
        "country_name",  # Only country is England
        "region_code",  # Already have region name
        "new_la_code",  # Already have local authority name
        "old_la_code",  # Already have local authority name
        "lad_code",  # Already have local authority district name
        "pcon_code",  # Already have parliamentary constituency name
        "english_devolved_area_code",  # Already have english devolved area name
        "local_enterprise_partnership_code",  # Already have partnership name
        "lsip_code",  # Already have lsip name
        "starts",  # Only applies to apprenticeships
        "starts_rate_per_100000_population",  # Only applies to apprenticeships
        "participation_rate_per_100000_population",  # Already have participation
        "achievements_rate_per_100000_population",  # Already have achievements
    ]
)
```

The DataFrame information for ‘fes_df’ was then printed, highlighting two issues: several columns contained NaN (blank) values, and the columns 'participation', 'achievements', and 'population_estimate' were of type object when they were expected to be numeric (figure 2).

Figure 2, ‘fes_df’ DataFrame information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584998 entries, 0 to 584997
Data columns (total 15 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   time_period                       584998 non-null  int64
 1   geographic_level                  584998 non-null  object
 2   region_name                       257878 non-null  object
 3   la_name                           83960 non-null   object
 4   lad_name                          168398 non-null  object
 5   pcon_name                         291736 non-null  object
 6   english_devolved_area_name        6624 non-null    object
 7   local_enterprise_partnership_name 20944 non-null   object
 8   lsip_name                         7264 non-null    object
 9   provision_type                    584998 non-null  object
 10  level_or_type                     584998 non-null  object
 11  age_summary                       584998 non-null  object
 12  participation                     584998 non-null  object
 13  achievements                      584998 non-null  object
 14  population_estimate               584998 non-null  object
dtypes: int64(1), object(14)
memory usage: 66.9+ MB
```

The expected numeric columns were defined as type object due to the presence of non-numeric characters, such as ‘x’ indicating unavailable data, as specified in the ‘data-guidance.txt’. To address this, a function was created to replace non-numeric characters with NaN and subsequently fill NaN values with 0, which also converts the columns to numeric. This function was successfully executed on the expected numeric columns (figure 3).

Figure 3, ‘fes_df’ column data types after converting expected numeric columns

```
time_period                       int64
geographic_level                 object
region_name                      object
la_name                          object
lad_name                         object
pcon_name                        object
english_devolved_area_name       object
local_enterprise_partnership_name object
lsip_name                        object
provision_type                   object
level_or_type                    object
age_summary                      object
participation                   float64
achievements                    float64
population_estimate             float64
```

The DataFrame was then checked for duplicate rows, and none were found. Next, the columns containing NaN values were inspected. These NaNs were present only due to

different aggregations of ‘geographic_level’. Filtering by one 'geographic_level' resulted in NaN values for other geographic levels. For example, filtering by 'English devolved area' resulted in NaNs for 'region_name' (figure 4). Therefore, no genuine NaNs existed.

Figure 4, NaN values when filtering ‘geographic_level’ by ‘English devolved area’

```
Geographic level: English devolved area

time_period                          0
geographic_level                     0
region_name                       6624
la_name                           6624
lad_name                          6624
pcon_name                         6624
english_devolved_area_name           0
local_enterprise_partnership_name 6624
lsip_name                         6624
provision_type                       0
level_or_type                        0
age_summary                          0
participation                        0
achievements                         0
population_estimate                  0
```

Lastly, to improve interpretability of the column names, these were renamed accordingly (figure 5).

Figure 5, renaming ‘fes_df’ column names

```python
# Create column renaming dictionary
col_rename_dict = {
    "time_period": "Academic Year",
    "geographic_level": "Geographic Level",
    "region_name": "Region",
    "la_name": "Local Authority",
    "lad_name": "Local Authority District",
    "pcon_name": "Parliamentary Constituency",
    "english_devolved_area_name": "English Devolved Area",
    "local_enterprise_partnership_name": "Enterprise Partnership",
    "lsip_name": "Local Skills Improvement Plan",
    "provision_type": "Provision Type",
    "level_or_type": "Level or Type",
    "age_summary": "Age Group",
    "starts": "Starts",
    "participation": "Participation",
    "achievements": "Achievements",
    "population_estimate": "Population Estimate",
}

# Rename columns using col_rename_dict
fes_df.rename(columns=col_rename_dict, inplace=True)
```

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to deepen the understanding of the dataset and assess its viability for machine learning. Since multiple aggregations of 'geographic_level' exist, 'fes_df' was filtered by 'Local authority', as this breakdown offers a suitable balance of data size and granularity, to create 'la_fes_df'.

The unique values of all categorical columns from 'la_fes_df' were displayed, revealing that several columns contained either too few or too many distinct groups to be viable for analysis or machine learning (figure 6).

Figure 6, distinct values of categorical columns from 'la_fes_df'

```
Academic Year: 6
Geographic Level: 1
Region: 10
Local Authority: 156
Local Authority District: 0
Parliamentary Constituency: 0
English Devolved Area: 0
Enterprise Partnership: 0
Local Skills Improvement Plan: 0
Provision Type: 4
Level or Type: 30
Age Group: 3
```

These columns were subsequently dropped from the DataFrame (figure 7).

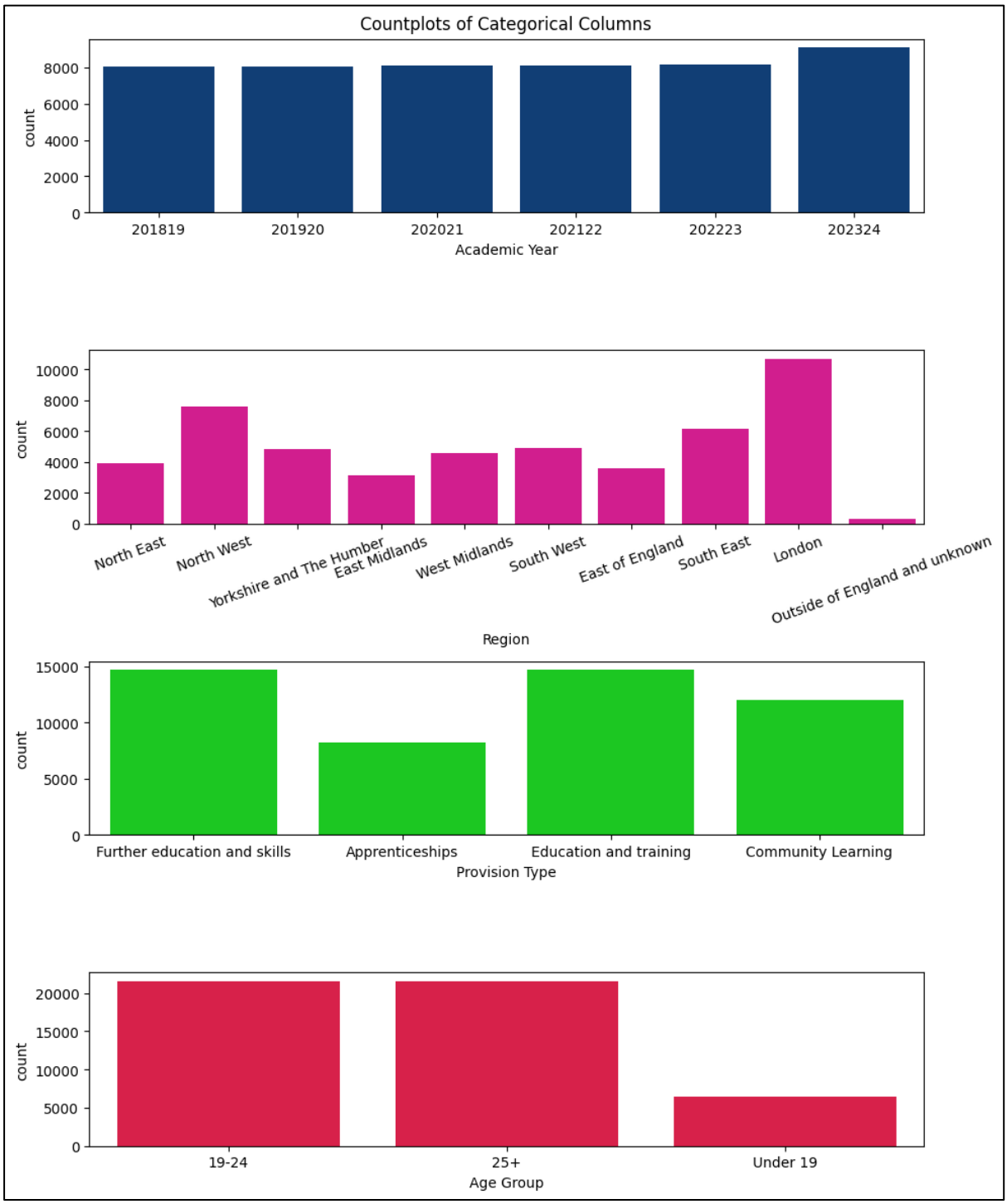
Figure 7, dropping columns with too few or many distinct values

```
la_fes_df = la_fes_df.drop(
    columns=[
        "Geographic Level",
        "Local Authority",
        "Local Authority District",
        "Parliamentary Constituency",
        "English Devolved Area",
        "Enterprise Partnership",
        "Local Skills Improvement Plan",
        "Level or Type",
    ]
)

cat_cols = [
    "Academic Year",
    "Region",
    "Provision Type",
    "Age Group",
]
```

Count plots of the remaining categorical columns were then generated to analyse the balance of the distinct groups (figure 8).

Figure 8, count plots of categorical columns from ‘la_fes_df’



'Academic Year’ was well balanced and required no action. For ‘Region’, the ‘Outside of England and unknown’ group was significantly underrepresented; however, it was deemed irrelevant as this business case focuses on learners in England, and thus was filtered out. The remaining regions had varying counts, though these correlate with the population of England (ONS, 2022), and were considered appropriate.

‘Provision Type’ was well balanced, except for ‘Apprenticeships’, which were somewhat underrepresented but not enough to warrant action. Regarding ‘Age Group’, the ‘19-24’ and ‘25+’ groups were balanced, whereas the ‘Under 19’ group had notably fewer counts. This imbalance was noted for consideration during machine learning, as

models may require resampling to address potential performance issues with this age group.

The statistics of the numeric columns were then printed using the ‘describe’ method (figure 9).

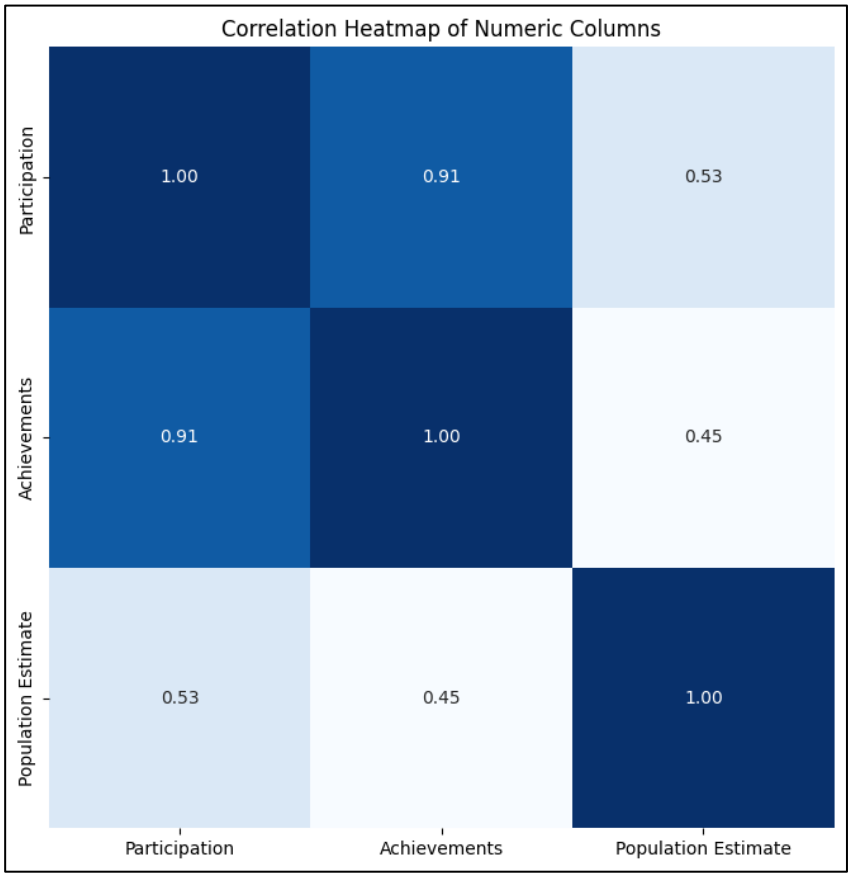
Figure 9, statistics of numeric columns from ‘la_fes_df’

	Academic Year	Participation	Achievements	Population Estimate
count	49269.000000	49269.000000	49269.000000	49269.000000
mean	202077.181493	598.778542	328.623069	98101.825529
std	174.256047	962.616284	693.065452	127378.438347
min	201819.000000	0.000000	0.000000	28.000000
25%	201920.000000	50.000000	20.000000	16006.000000
50%	202122.000000	270.000000	100.000000	44810.000000
75%	202223.000000	730.000000	310.000000	138965.000000
max	202324.000000	17540.000000	13120.000000	812460.000000

Column 'Academic Year' can be disregarded as it is categorical. The average values for 'Participation', 'Achievements', and 'Population Estimate' are 599, 329, and 98,102 respectively. The large standard deviations, and wide ranges between quartiles, indicate significant variance in the data, particularly towards higher values.

A correlation heatmap was produced to determine the relationships between the numeric columns (figure 10)

Figure 10, correlation heatmap of numeric columns



'Participation' shows a very strong positive correlation with 'Achievements', which was expected since higher participation generally leads to more achievements. Interestingly, 'Participation' exhibits only a moderate correlation with 'Population Estimate', suggesting that densely populated areas do not consistently have higher student enrolment. Moreover, 'Population Estimate' also shows a slight positive correlation with 'Achievements', indicating that areas with larger populations tend to have more achievers, though exceptions exist.

Box plot and histograms were then created to more closely scrutinise the distribution of the numeric columns (figures 11 and 12).

Figure 11, boxplots of numeric columns

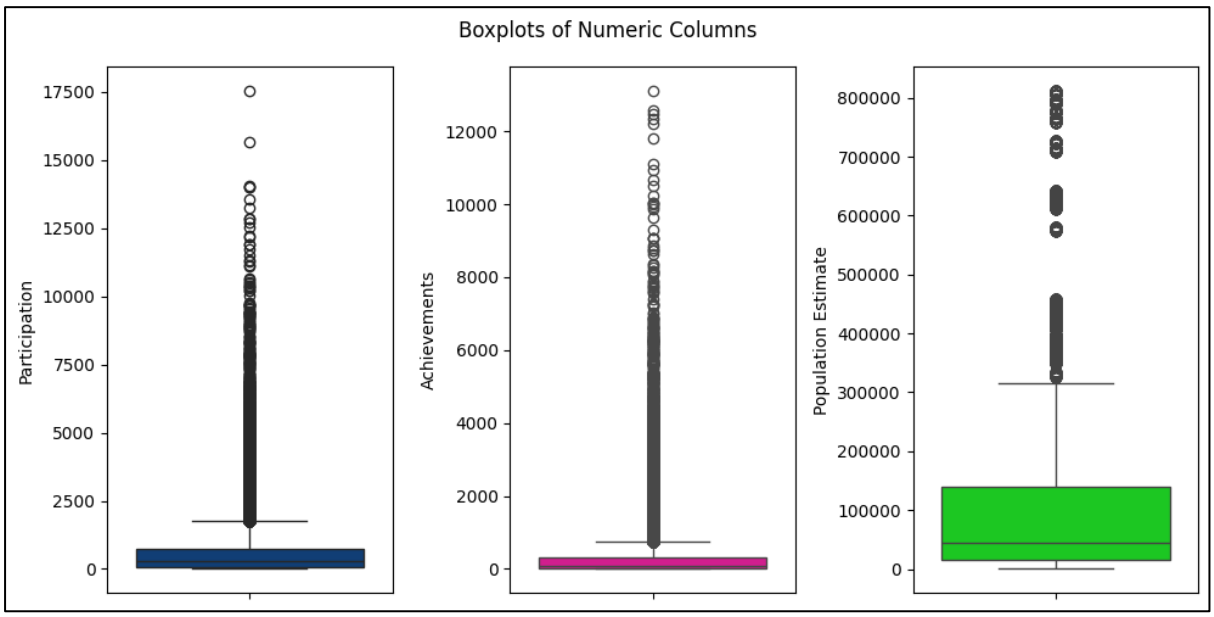
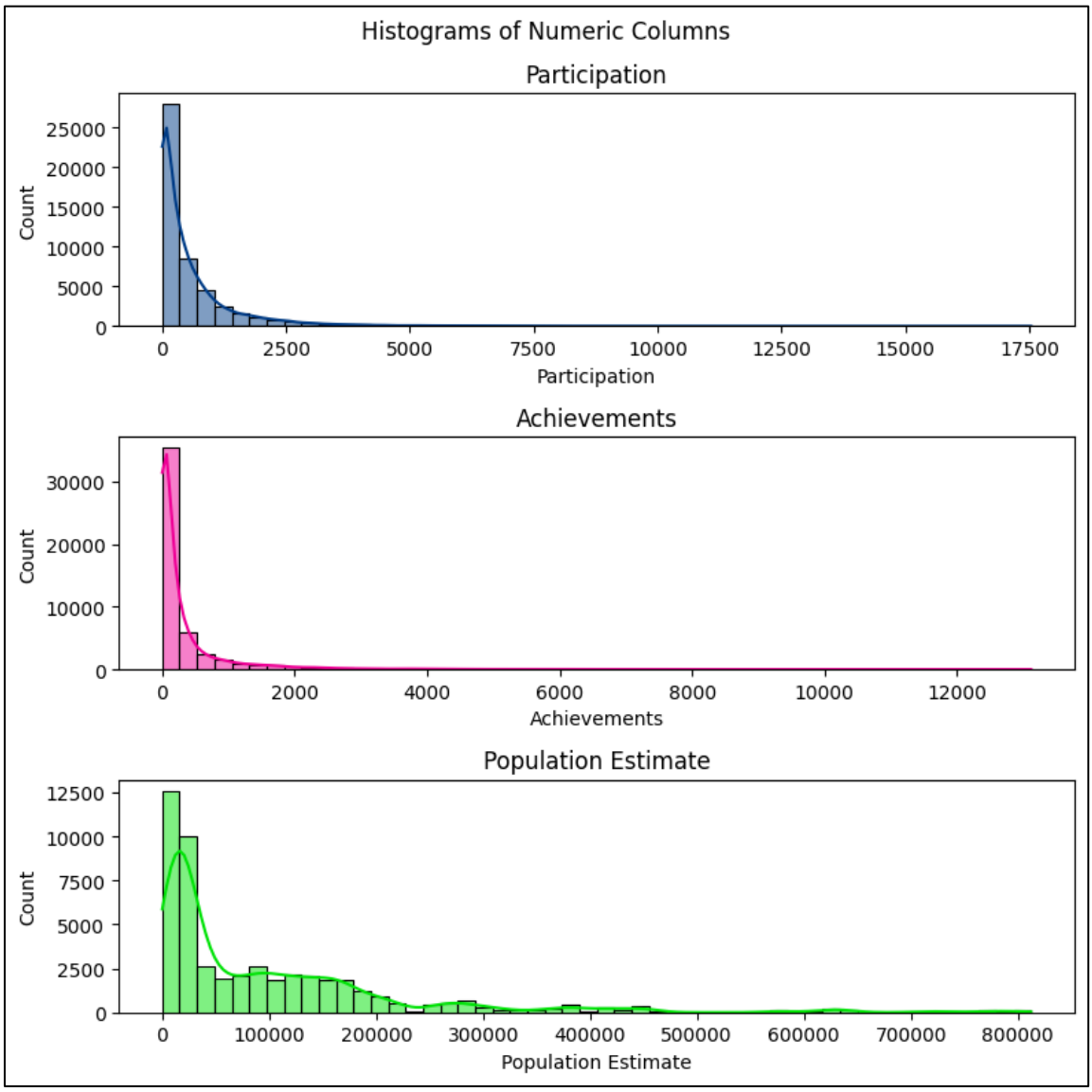


Figure 12, histograms of numeric columns

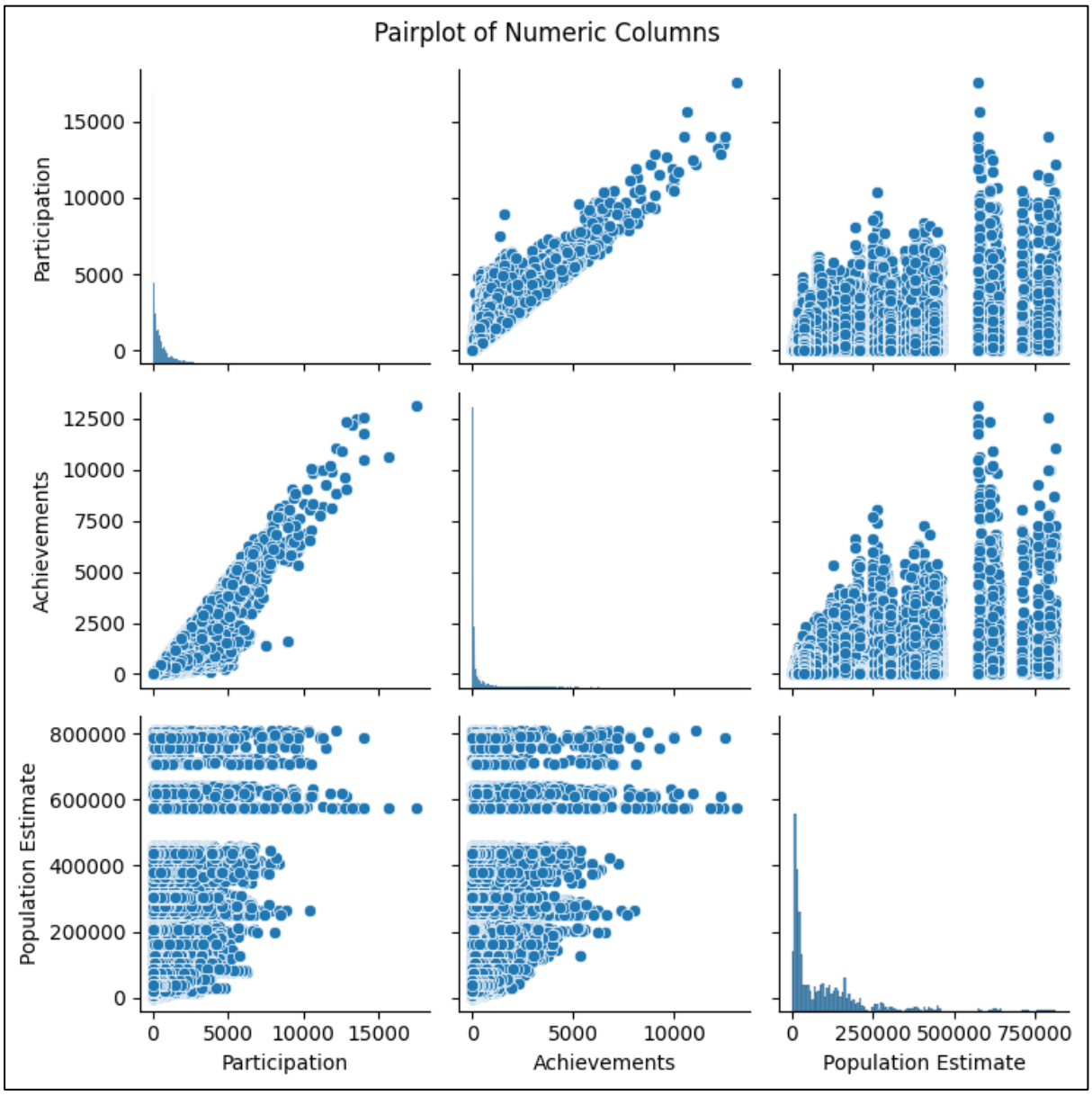


Initially, it appeared that significant outliers were present in the data, as indicated by multiple data points above the upper whiskers in the box plots. However, analysis of the histograms revealed that this was due to the heavily right-skewed distribution of all numeric columns. Right-skewedness is also commonplace in population datasets (McGuinness, Bennett, and Riley, 1997).

Despite the heavy right-skew, the histograms show that the numeric columns follow a similar pattern of distribution. Therefore, no immediate action was taken to transform the data, as its current form may better reflect the trends of these attributes. However, it was noted that if machine learning models performed poorly due to this skewness, transformations such as log-transformation might be required (Feng, et al, 2014).

Pair plots were generated to further inspect the relationships between the numeric columns (figure 13).

Figure 13, pair plots of numeric columns



These plots reinforced the extremely close correlation between 'Participation' with 'Achievements', as well as the weaker trends between 'Participation' with 'Population Estimate', and 'Achievements' with 'Population Estimate'. The positive trends observed between these attributes were considered promising for machine learning modelling.

Machine Learning (ML) Modelling

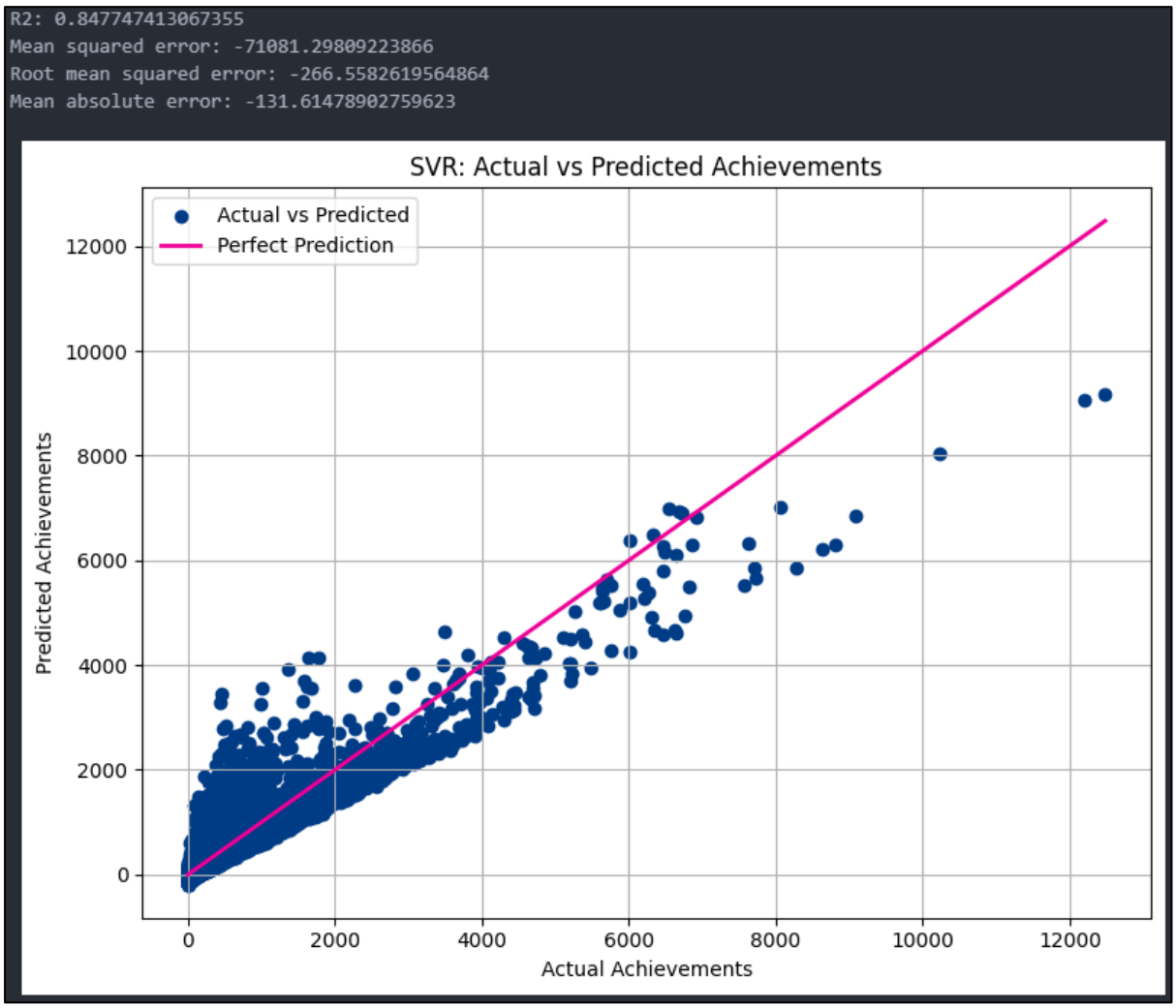
Predicting Student Achievement Numbers

Machine Learning (ML) models were developed to predict the number of achievers based on other attributes in the dataset. Some preprocessing was necessary beforehand. All categorical columns were encoded using one-hot encoding, as most ML models require numerical data (Potdar, Pardawala, and Pai, 2017). The features (all columns except 'Achievements') and the target ('Achievements') were split into training and testing sets accordingly. RobustScaler was then applied to scale the datasets and manage the particularly large values in the right-skewed data.

Support Vector Regression (SVR) was the first model employed. To find the optimal values for the 'C' regularisation strength and epsilon hyperparameters, GridSearchCV was implemented, as it is effective for this task (Shekar and Dagnew, 2019). RepeatedKFold was also utilised to accurately cross-validate the model (Tuson, et al, 2021). From this process, a 'C' of 1 and epsilon value of 0.1 were determined as the ideal values, with an R^2 score of 0.85.

The final SVR model was produced using the aforementioned optimal values for the hyperparameters. Regression metric scores were printed, and a scatter plot comparing predicted versus actual achievements was generated to visualise the results (figure 14).

Figure 14, final Support Vector Regression model results



The model demonstrated decent accuracy with an R^2 score of 0.85. The root mean squared error (RMSE) and mean absolute error (MAE) were -266.56 and -131.61, respectively. Given that English regions typically have populations in the tens of thousands, errors of a few hundred achievers are quite respectable in comparison.

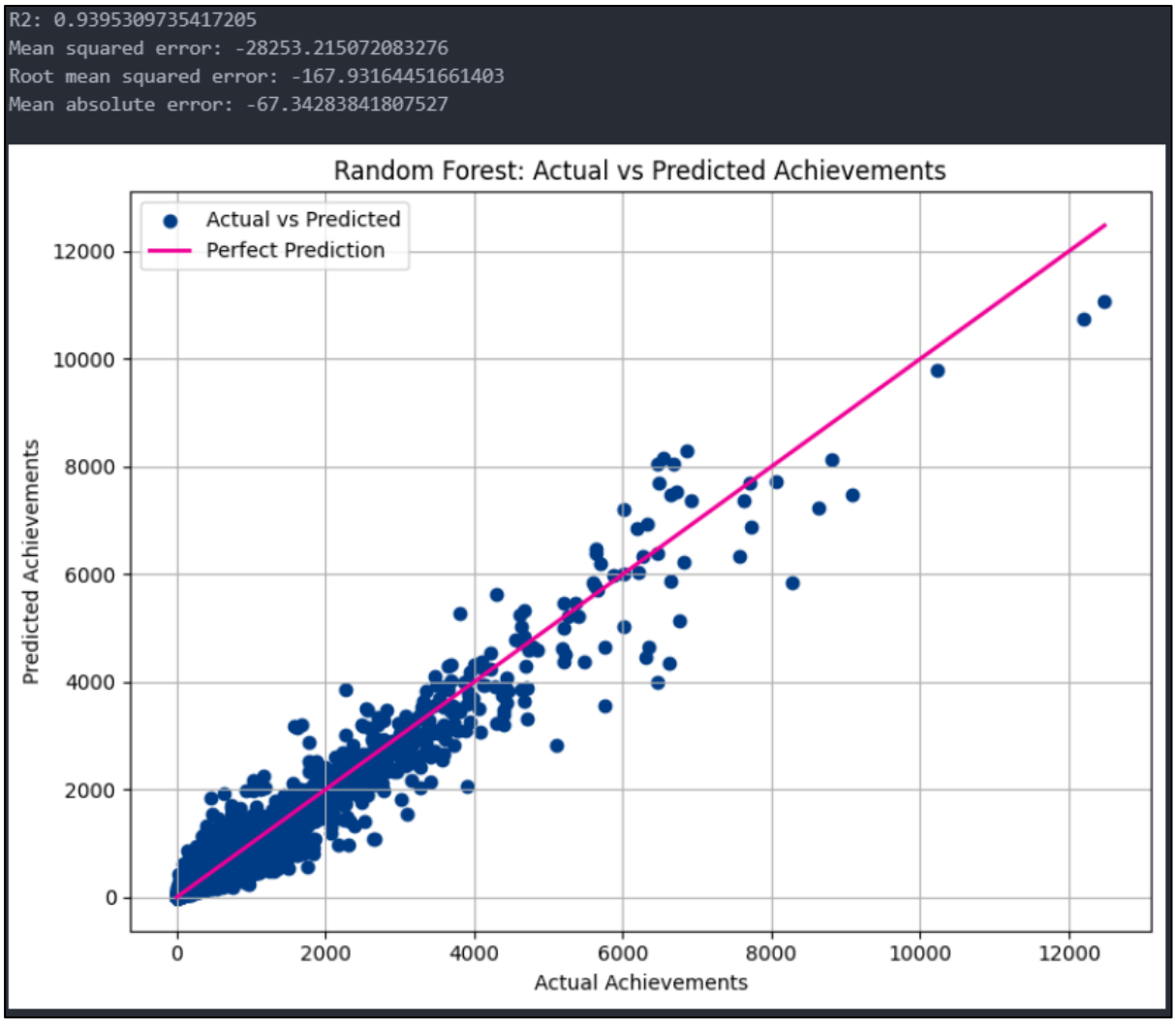
The scatter plot depicting 'Actual vs Predicted' achievements shows that points were generally close to the ideal prediction, indicating effective performance. However, higher values tended to have larger errors. These results suggest that both the numeric

and categorical features of the dataset are closely related with student achievement numbers and contribute effectively to predicting them.

Given the model's high dimensionality, especially after encoding all categorical attributes, an attempt was made to reduce complexity using Principal Component Analysis (PCA). The dimensionality was reduced to one component. However, this significantly decreased performance, resulting in an R^2 score of 0.62. Consequently, this version was discarded.

Next, a Random Forest (RF) model was developed to see if better results could be achieved. The same methodology, employing GridSearchCV and RepeatedKFold, was used to determine the optimal values for the hyperparameters. This time, the hyperparameters considered were 'n_estimators', 'max_depth', 'min_samples_split', and 'min_samples_leaf'. The optimal values found were 100, 10, 5, and 1, respectively. The final RF model was generated using these hyperparameter values. The same regression metrics and scatter plot were produced for the RF model for comparison (figure 15).

Figure 15, final Random Forest model results



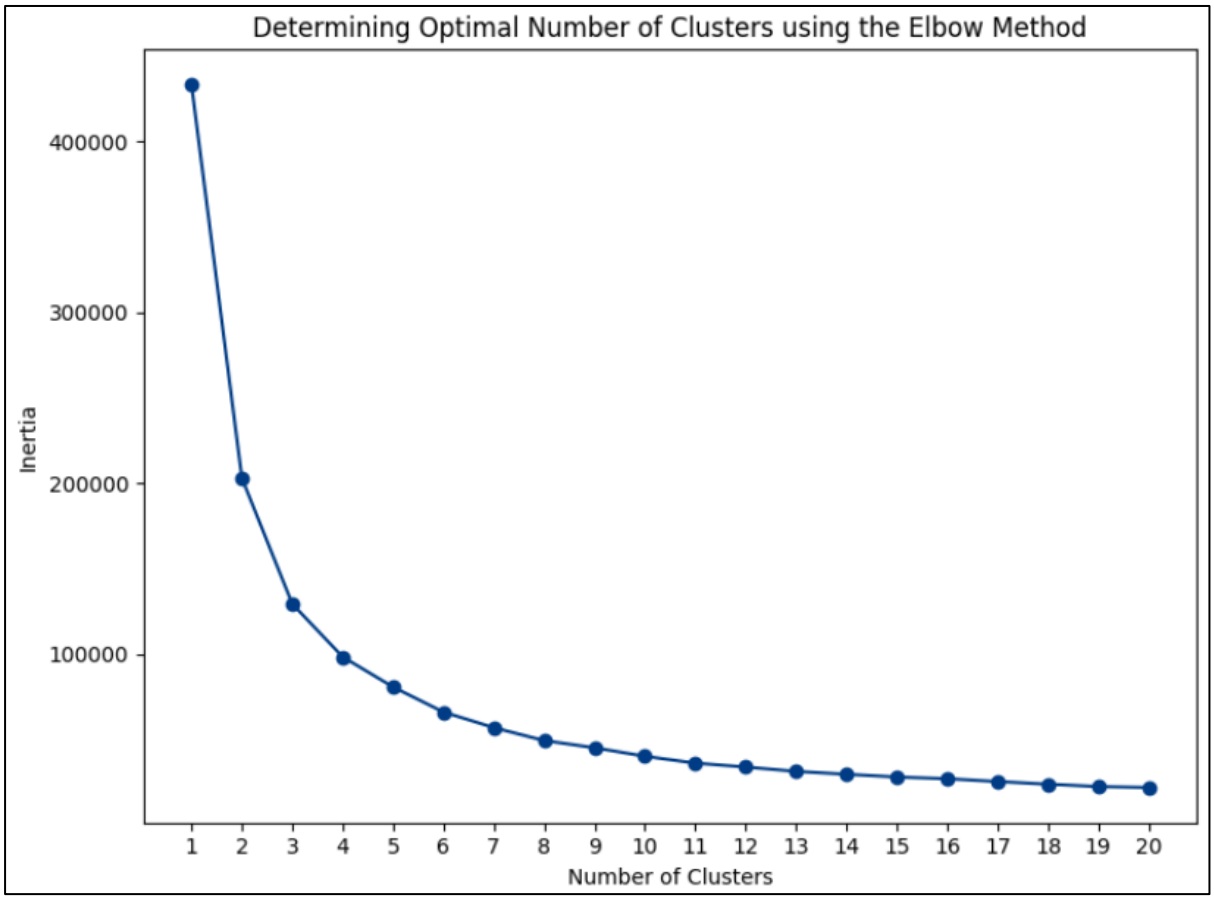
The final RF model produced similar results to the SVR model but with improved overall accuracy. This was demonstrated by its higher R^2 score of 0.94 and lower RMSE and MAE of -167.93 and -67.34, respectively. The scatter plot of 'Actual vs Predicted'

achievements shows data points generally closer to the perfect prediction line. Overall, the RF model outperformed SVR, although it required significantly more execution time due to its higher computational expense.

Identifying Student Achiever Groups

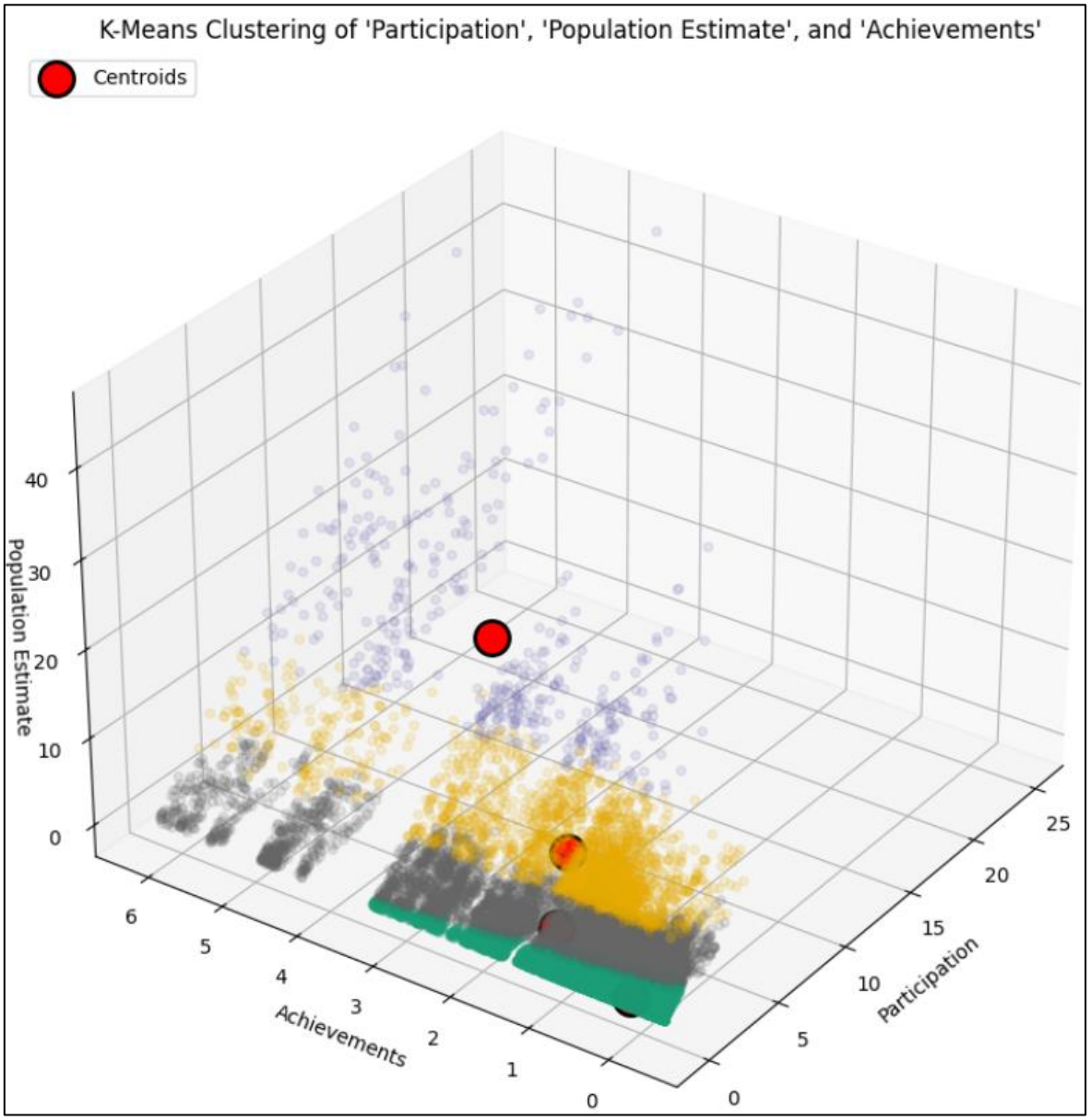
A K-means clustering model was developed to identify distinct student achiever groups, using 'Participation', 'Population Estimate', and 'Achievements' as features. The K-means++ algorithm was employed due to its effectiveness in selecting initial centroids (Bahmani, et al, 2012). The elbow method was used to determine the optimal number of clusters (figure 16).

Figure 16, Determining optimal number of clusters using the elbow method



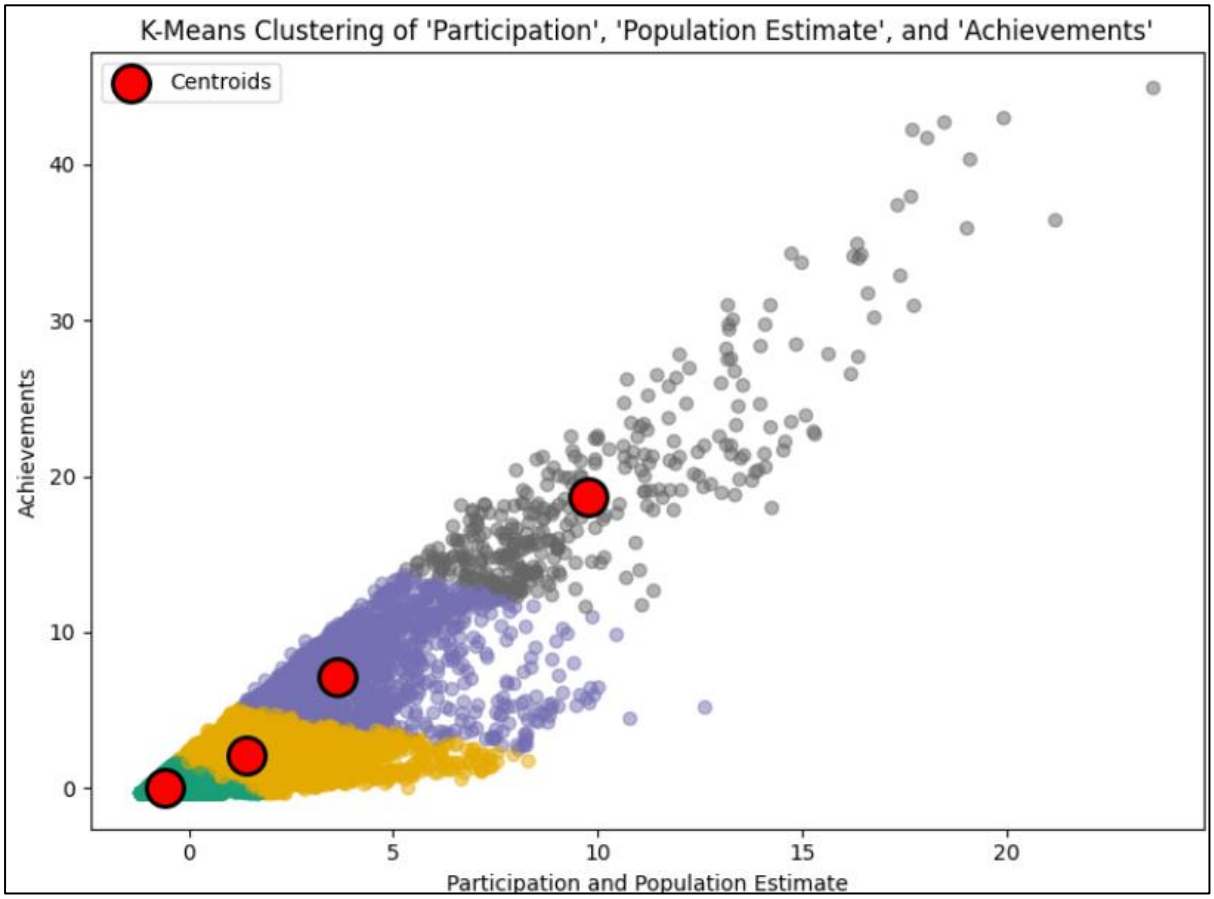
Four was determined to be the ideal number of clusters. A K-means model was then fitted with this in mind, and a 3D plot of the clusters and centroids was generated (figure 17).

Figure 17, K-means clustering of ‘Participation’, ‘Population Estimate’, and ‘Achievements’ 3D plot



Interpreting both the clusters and centroids from the 3D plot proved challenging. Consequently, PCA was used to reduce the dimensions to two by combining 'Participation' and 'Population Estimate'. The K-means model was then reproduced and visualised using a 2D plot instead (figure 18).

Figure 18, K-means clustering of ‘Participation’, ‘Population Estimate’, and ‘Achievements’ 2D plot



The 2D plot proved more interpretable while retaining similar cluster groupings to the initial K-means model, and thus, this model was retained as the final version. The following achiever group clusters were identified from the 2D plot:

- Green cluster: very low ‘Participation and Population Estimate’ with very low ‘Achievements’.
- Yellow cluster: low ‘Participation and Population Estimate’ with low ‘Achievements’.
- Purple cluster: moderate ‘Participation and Population Estimate’ with moderate ‘Achievements’.
- Grey cluster: high ‘Participation and Population Estimate’ with high ‘Achievements’.

The final K-means model successfully highlighted distinct student achiever groups. This can be useful for further applications, such as cross-referencing the regions to identify where the different achiever clusters are located across England.

Concluding Remarks

Overall, the analysis determined that numerous factors, such as population and location, can influence the number of student achievers. Although, these factors do exhibit patterns and trends, and can be visualised, analysed, and utilised for

subsequent machine learning. For instance, the level of participation strongly correlates with the number of achievers.

In terms of predicting achievement numbers, the RF model performed the most accurately. However, if quicker, indicative numbers are sufficient, the SVR model is suitable. Additionally, distinct achiever clusters were identified through K-means clustering. This could be useful for further applications, such as analysing the clusters across different educational provisions to compare performance.

References

Bahmani, B., et al (2012) 'Scalable k-means++', *arXiv preprint arXiv,*1203.6402 [online]. Available at: https://arxiv.org/abs/1203.6402 (Accessed 12th July 2024)

Education & Skills Funding Agency (ESFA) (2024) *Funding guidance for young people 2024 to 2025 rates and formula* [Online]. Available at: https://www.gov.uk/government/publications/funding-rates-and-formula/funding-guidance-for-young-people-2024-to-2025-rates-and-formula (Accessed 10th July 2024)

Feng, C., et al (2014) 'Log-transformation and its implications for data analysis', *Shanghai archives of psychiatry*, *26*(2) [online]. Available at: https://doi.org/10.3969%2Fj.issn.1002-0829.2014.02.009 (Accessed 10th July 2024)

GOV.UK (2024) *Data Catalogue* [Online]. Available at: https://explore-education-statistics.service.gov.uk/data-catalogue (Accessed 10th July 2024)

McGuinness, D., Bennett, S., Riley, E. (1997) 'Statistical analysis of highly skewed immune response data', *Journal of immunological methods*, 201(1) [online]. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0022175996002165 (Accessed 10th July 2024)

The National Archives (2024) *Open Government License* [Online]. Available at: https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/ (Accessed 10th July 2024)

Office for National Statistics (ONS) (2022) *Population estimates for the UK, England, Wales, Scotland, and Northern Ireland: mid-2022* [Online]. Available at: https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/bulletins/annualmidyearpopulationestimates/mid2022 (Accessed 10th July 2024)

Ofsted (2024) *School inspection handbook* [Online]. Available at: https://www.gov.uk/government/publications/school-inspection-handbook-eif/school-inspection-handbook-for-september-2023 (Accessed 10th July 2024)

Potdar, K., Pardawala, T.S., Pai, C.D. (2017) 'A comparative study of categorical variable encoding techniques for neural network classifiers', *International Journal of Computer Applications*, *175*(4) [online]. Available at: http://dx.doi.org/10.5120/ijca2017915495 (Accessed 12th July 2024)

Shekar, B. H., Dagnew, G. (2019) 'Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data' *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 1–8 [online]. Available at: https://doi.org/10.1109/ICACCP.2019.8882943 (Accessed 12th July 2024)

Tuson, M. et al (2021) 'Predicting future geographic hotspots of potentially preventable hospitalisations using all subset model selection and repeated k-fold cross-validation',

International Journal of Environmental Research and Public Health, 18(19) [online]. Available at: https://www.mdpi.com/1660-4601/18/19/10253 (Accessed 12th July 2024)