**Electronic Assignment Cover sheet**

Please fill out and attach as the first page of Assignment.

**Student (s) Number as per your student card:** Juliana Salvadori - No: 10521647

**Course Title:** MSc Data Analytics

**Lecturer Name:** Abhishek Kaushik

**Module/Subject Title:** Machine Learning (B9DA104)

**Assignment Title:** CA2

**No of Words:** 2500

GitHub Regression code: <https://gist.github.com/jusalvadori/2bf43a415f4a7f41eea490df30359a50>

GitHub Classification code:

<https://gist.github.com/jusalvadori/e1379f0c7030d5c3f3bc781057c72b63>

Please let me know if you have problem in access any of the above links. Thank you.

**Q1**

● Define Data sampling and its steps with examples.

Data sampling is a statistical method used to help data analysts to discover patterns in a large set of data without the need to analyse the whole set of data, which would be time-consuming and costly. It means it is possible to find meaningful information about a population by analysing a sample of that population. For example, to know if a cake is good or not, you do not need to eat the whole cake, you just need to taste a slice of the cake. Steps:

1) define population to be analysed - cakes from bakery X

2) define sampling frame, that is, where the sample will be selected from - you want to analyse one cake or the entire cake production for a day.

3) define method of data sampling to be applied. The method can be a probability or non-probability sampling, and there are various methods for each of these types.

4) define sampling size - it will depend on the population size, to analyse one cake, one slice of it would be enough, to analyse the entire cake production for a day, it will probably necessary to take several slices of cake during that day.

5) and then, perform the sample collection and analyse.

References:

Cavello, M. (2020) ‘How to See the Bigger Picture with Data Sampling’, Learning Hub. Available at: <https://learn.g2.com/data-sampling#:~:text=What%20is%20data%20sampling%3F,often%20it%20should%20be%20collected>

(Accessed: 01 Nov 2020)

Rouse, M. (2018) ‘Data Sampling’, Search Business Analytics. Available at:

<https://searchbusinessanalytics.techtarget.com/definition/data-sampling> (Accessed: 01 Nov 2020)

Analytics help, Sampled data and (other), About data sampling (2020) Google. Available at: <https://support.google.com/analytics/answer/2637192?hl=en> (Accessed: 01 Nov 2020)

Hessing, T. (2019) ‘Data Sampling Techniques & Uses’, Six Sigma Study Guide. Available at: <https://sixsigmastudyguide.com/data-sampling/> (Accessed: 01 Nov 2020)

Akhouri, A. (2018) ‘Sampling Process’. Available at: <https://www.amritaakhouri.com/single-post/2018/01/23/Sampling-process> (Accessed: 01 Nov 2020)

● Define Decision Tree, Information gain and Entropy.

Decision tree is one of the supervised learning methods used in decision analysis to solve regression problems and mostly classification problems. It works through a set of rules in the form of conditional statements to show and classify the data according to the defined conditions, in order to help in making decision about the target variable.

Information gain is used as a split criterion for decision trees to decide which is the predictor that should be picked up as root for the decision tree at each step by showing which predictor is the most significant.

Entropy is the measure of uncertainty, which is used to calculate the Information gain, and therefore, impacts on how the decision tree chooses to split the data. Thus, we can say:

* greater probability and less information mean more uncertainty, and therefore, less information gain.
* less probability and more information mean less uncertainty, and therefore, greater information gain.

Accordingly, when we have several predictors, how to know which one is the most significant? The one which gives you the biggest information gain, and consequently, the less uncertainty.

References:

Honari, B. (2020) ‘Decision Tree’, Data Mining. Available at: <https://elearning.dbs.ie/pluginfile.php/1122510/mod_resource/content/1/Decision%20Tree%20Lecture%20Notes.pdf> (Accessed: 01 Nov 2020)

Gupta, Prashant (2017) ‘Decision Trees in Machine Learning’, Towards Data Science. Available at: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052> (Accessed: 01 Nov 2020)

Sayad, S. (2020), ‘Decision Tree – Classification’, An Introduction to Data Science. Available at:

<https://www.saedsayad.com/decision_tree.htm#:~:text=The%20information%20gain%20is%20based,%2C%20the%20most%20homogeneous%20branches).&text=The%20result%20is%20the%20Information%20Gain%2C%20or%20decrease%20in%20entropy> (Accessed: 01 Nov 2020)

Sujan, N. (2018) ‘What is Entropy and why Information gain matter in Decision Trees?’, Medium. Available at: <https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01> (Accessed: 01 Nov 2020)

Loaiza, S. (2020), ‘Entropy and Information Gain’, Towards Data Science. Available at: <https://towardsdatascience.com/entropy-and-information-gain-b738ca8abd2a> (Accessed: 01 Nov 2020)

Alex (2020) ‘Decision Trees, Entropy, and Information Gain’, BOOSTEDML Articles on Statistics and Machine Learning for Healthcare. Available at: <https://boostedml.com/2020/06/decision-trees-entropy-and-information-gain.html>

(Accessed: 01 Nov 2020)

● Define Chinese restaurant algorithm and Agglomerative Hierarchical Clustering with example.

Chinese restaurant process (CRP) is an unsupervised learning method used for clustering of data. It is a random process comparable to a Chinese Restaurant that has an infinite number of tables (clusters) and the customers walk in and chose to seat at an empty or occupied table accordingly with a set of probabilities applied by the algorithm. The algorithm will calculate the probability to select an empty or occupied table based on the total of customers already at each table (partition).

Hierarchical clustering is one of the clustering techniques which groups objects by their similarity aiming to build a hierarchy of clusters. There are two types of hierarchical clustering: agglomerative and divisive, the agglomerative being the most popular of them. In the agglomerative clustering, each object starts as an individual cluster, the next step is to calculate the proximity between them and then each the object (cluster) is joined to the nearest one until there is only one big cluster or K clusters are created. Calculate the proximity or similarity between each object is an important step to help on the merging process, this can be done using several methods.

References:

Singh, M. (2015) ‘Chinese Restaurant Process’, SlideShare. Available at: <https://www.slideshare.net/MohitdeepSingh/chinese-restaurant-process> (Accessed: 01 Nov 2020)

Stoltzfus, J. (2020) ‘How can the Chinese restaurant process and other similar machine learning models apply to enterprise AI?’, Techopedia. Available at:

<https://www.techopedia.com/how-can-the-chinese-restaurant-process-and-other-similar-machine-learning-models-apply-to-enterprise-ai/7/33191> (Accessed: 01 Nov 2020)

Kaustubh, N. (2017) ‘Chinese Restaurant Process’, Medium. Available at: <https://medium.com/somx-labs/chinese-restaurant-process-1bf036b5bf0f> (Accessed: 01 Nov 2020)

‘HIERARCHICAL CLUSTERING IN R: THE ESSENTIALS’ (2020). Data Novia. Available at: <https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/> (Accessed: 01 Nov 2020)

Manning, C., Raghavan, P. and Schütze, H. (2008) Introduction to Information Retrieval. Cambridge University Press. Available at: <https://nlp.stanford.edu/IR-book/html/htmledition/hierarchical-agglomerative-clustering-1.html> (Accessed: 01 Nov 2020)

Patlolla, C. (2018) ‘Understanding the concept of Hierarchical clustering Technique’, Towards Data Science. Available at:

<https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec> (Accessed: 01 Nov 2020)

‘Agglomerative Clustering’ (2020). RapidMiner Studio Core. Available at: <https://docs.rapidminer.com/latest/studio/operators/modeling/segmentation/agglomerative_clustering.html> (Accessed: 01 Nov 2020)

**Q2**

**Introduction**

In this task I went through two different datasets to apply some of the machine learning and data analysis techniques studied on this module.

The first dataset chosen was *Why Many Americans Don't Vote* [1], which has only categorical variables and, therefore, it was suitable to build only the classification model. Thus, I had to select a different dataset *The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women* [11] with the view to build the regression model required.

**Classification**

Classification models are used to predict discrete values, that is, it uses a set of independent variables to the predict a categorical dependent variable [2][3]. Basically, it will classify the data into two or more categories according to the features available to train the model.

There are different approaches for classification and, therefore, different types of algorithms that can be used according to the dataset to be utilised. On this task, I will use K-Nearest Neighbour for classification as it is simple and one of the most used in machine learning for this purpose.

**KNN algorithm**

K-Nearest Neighbour algorithm is one of the predictive supervised learning methods applied in machine learning for classification. It is a simple algorithm and also called as a lazy learner as it actually does not learn from the training dataset and stores a model that will be applied to classify any new data, instead it uses the training set to compare any new data to similar data points and then allocate the new data to the most similar category [6][7][8].

**About the dataset chosen for classification**

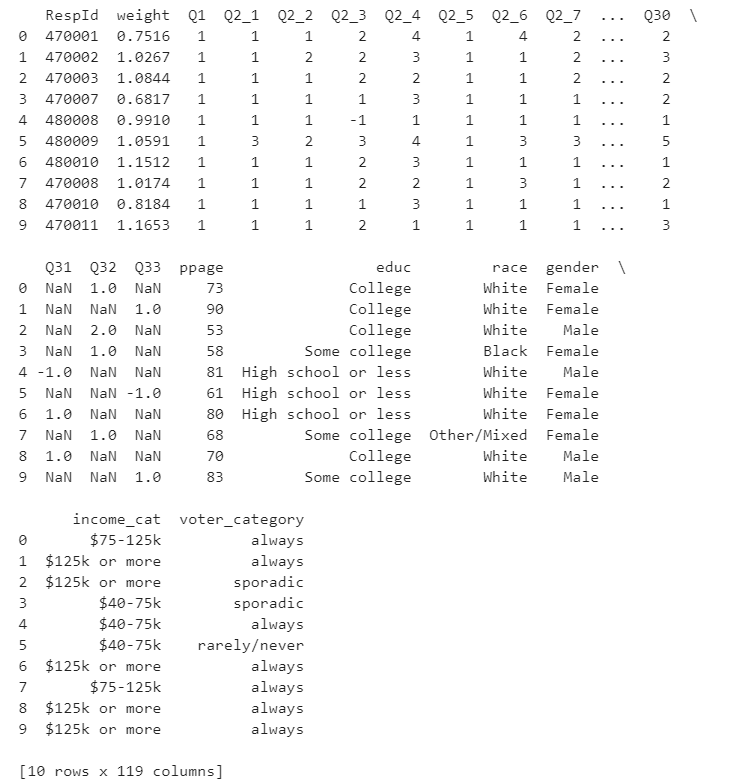
*Why Many Americans Don't Vote* is a survey that aims to map the profile of American voters and non-voters and understand what are the main reasons that lead them to choose not to vote. The survey had 8,327 respondents but only nearly 6,000 of them could be used on this analysis by the authors as they were able to match the answer to the respondents voting history and, therefore, the respondents were categorized in three main groups: people who rarely or never vote, people who sometimes vote and those ones who almost always vote [1]. And that categorization is what the classification algorithm built will try to predict based on a subset of the questions asked.

**Implementation**

Both classification and regression model were developed using Python and Pandas library to read, load and analyse the dataset. Matplotlib library was utilised in both for visualization and scikit-learn library for machine learning algorithms.

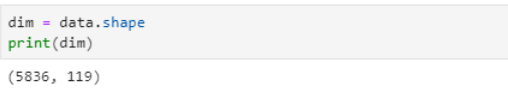
**Classification algorithm**

First step was to read the full data from the csv file into a data frame and then having a quick glance at that to understand how data looks like. See below the results:



It is possible to notice that there are 118 features, where the first columns are the respondent ID and the weight applied to that respondent, those two columns are not relevant for the classification process, hence, they can be removed from the dataset. Also, it is possible to observe that some of the features have lots of missing data (NaN) and will require special attention before applying the classification method. Likewise, I could see that most of the features have categorical values but one of them (ppage) is a continuous value and a few others have categorical information, however, they are expressed on the format of text instead of number.

Next step is to review the dimensions of the data to verify how many of the features I need to use for the classification operation.



The length of the dataset is nearly 6,000 records which means that I do not need to use all the 116 features available on the dataset, otherwise, I will have too many features and model can be confused and it can lead the process to an overfitting error. Thus, for this number of records, the model should ideally use only 6 features.

Afterwards, I carried out some cleaning to dataset:

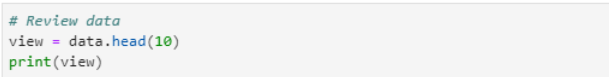
1 – Removing columns *RespID* and *weight*.

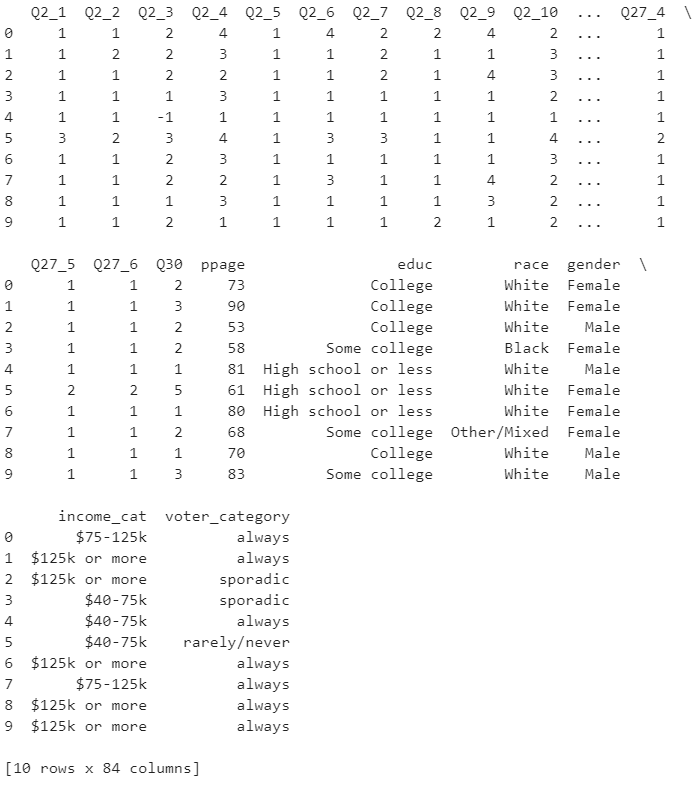
2 – Removing other columns where most of the values are missing (NaN).

3 – Removing column *Q1* as its value is always the same (1).

4 – Removing other columns where most of the values are not missing but they are not valid answer (-1).

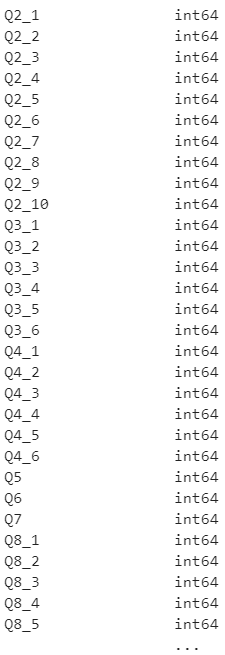
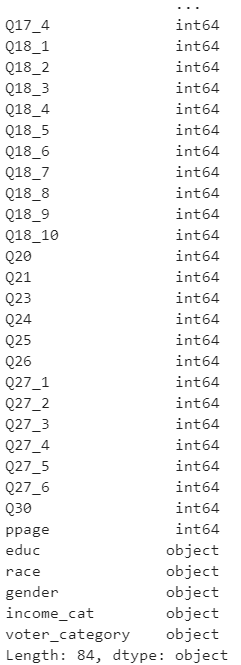
And then, having a new quick glance to the data, it is possible to noticed that data looks better, and the number of features was reduced to 83.



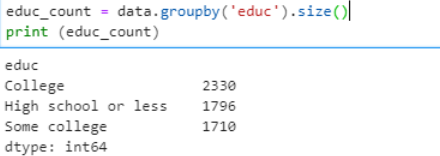


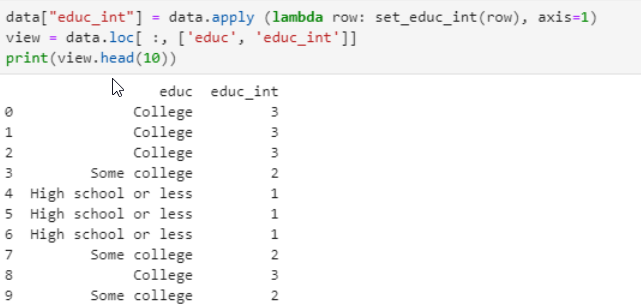
In order to be able to apply some type of method to visualise and select the most 6 relevant features that I would use later on the classification process, I needed to verify the data type of the features and apply any necessary conversion. Hence, using the *dtypes* function I could see which were the features that would need to be converted from ‘object’ (string) to numerical.



By using the *groupby* function I could identify the different answers to each of those features. Below I show one example of that conversion process for Education where a new column *educ\_int* was added to the dataset. The same process was applied to race, gender, income\_cat features and for the response variable voter\_category.

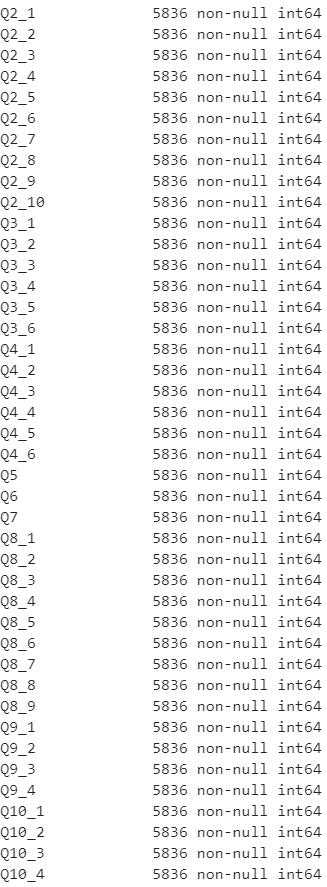
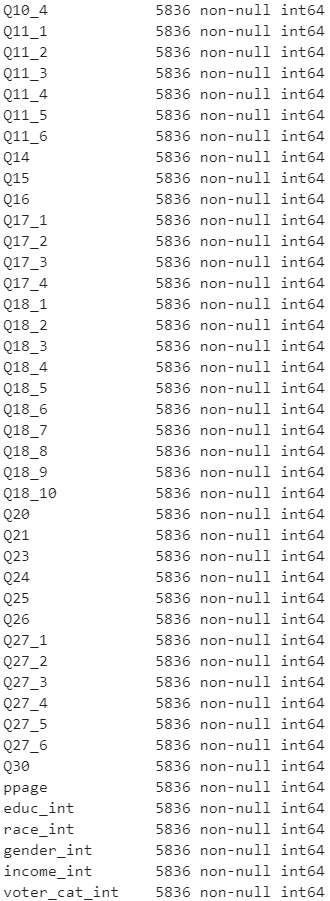




At this stage, I had a look at the description of data to identify if there is still any issue that needs attention. Through the *info* function I could see number of non-null values and the datatype of each variable.

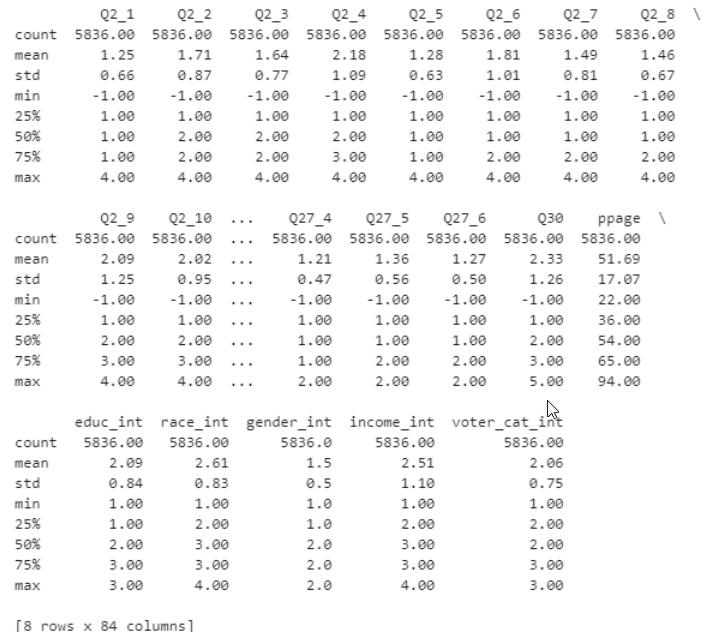


It shows that there is no NaN values as all the features are listed as 5836 non-null rows.

I also had a look at the summary of statistics for the dataset using the *describe* function.





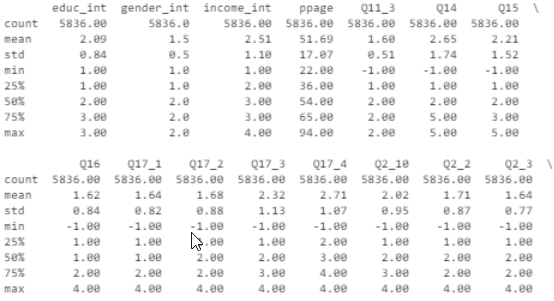
This summary shows that even there are no NaN values some of the features have min value as -1, which is not a valid value and it will probably require either remove the line or replace that value. Before applying any of these methods, I had a look at the features skewness, through a density plot, trying to identify any other columns that could be removed for not follow a normal (Gaussian) distribution.

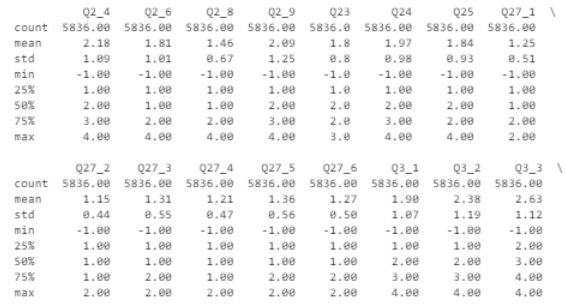
Calendar

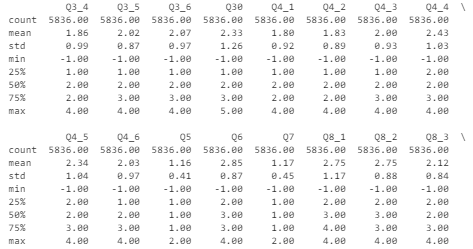
Description automatically generated

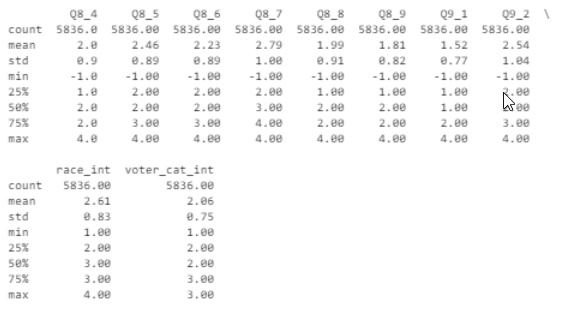
Based on data skewness that can be observed on the density plot, it is possible to reduce the number of features from 83 to 56, and then, I reviewed the summary of statistics for the dataset to verify if there is still any feature with invalid value -1.



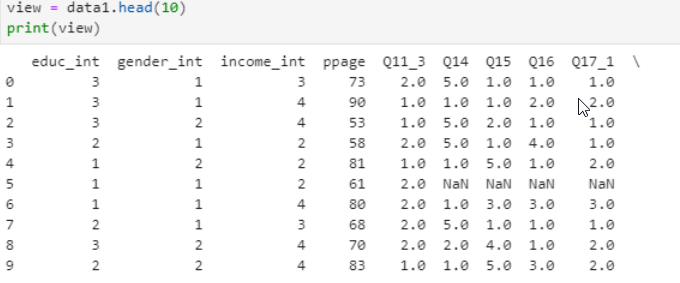








The summary shows that there are still features where min value is -1, so, next step was to verify that those lines that contain -1 values could be removed from the dataset. For that, I identified which columns have min value as -1 using the description result set and replaced those values for NaN and, therefore, I would be able to use any NaN function to deal with those values.



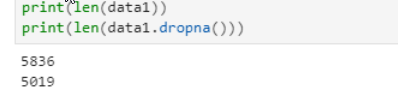
I also used the *missingno* library to visualize the missing values and confirm that the overwrite process had worked as expected.



A picture containing schematic

Description automatically generated

After added those NaN values I could assess the impact on the dataset size if the lines with NaN values were removed.



The actual dataset size is 5836 and removing lines with NaN values would reduce the size to 5019 records. The difference was not significant and as I would still have enough data, I removed those lines instead of inputting the missing values.

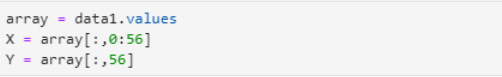
Note that at this stage I still had 57 features and I needed to select only 5 to build the classification model. Plotting the correlation matrix between those remaining features did not help to select the most significant ones. Thus, I had to use an alternative method to select the best features.

Graphical user interface

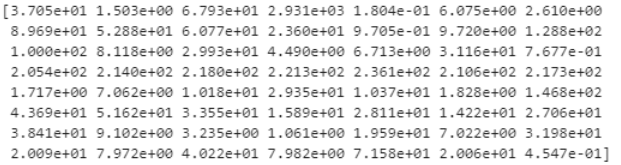
Description automatically generated

I decided to use *SelectKBest* function for that. Note that there was no need to apply data normalization, as all the remaining features are categorical.







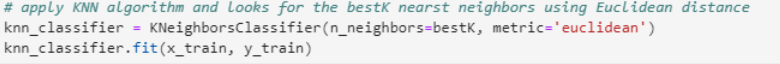


Base on the *SelectKBest* result set I selected the 5 features with the highest scores to build the model using the K Nearest Neighbor algorithm with Euclidean distance. Data was split in 25% for testing and 75% for training and the final accuracy found for the model was 59.60%, which is not a bad result but there is room for improvement.











See below confusion matrix showing actual vs predict data.

Chart, treemap chart

Description automatically generated

**Regression**

Regression models are also used to predict values, however, the output for this type of algorithm is numerical (continuous. The idea on this algorithm is to estimate the dependent variable by building the relationship between the independent variables on the format of a function y = f(X) [10][3][23]. There are also different approaches for regression and, therefore, different types of algorithms that can be used according to the dataset to be utilised. On this task, I will use Linear regression as it is simple to understand, easy to apply and one of the most used methods for prediction in machine learning.

**Linear regression**

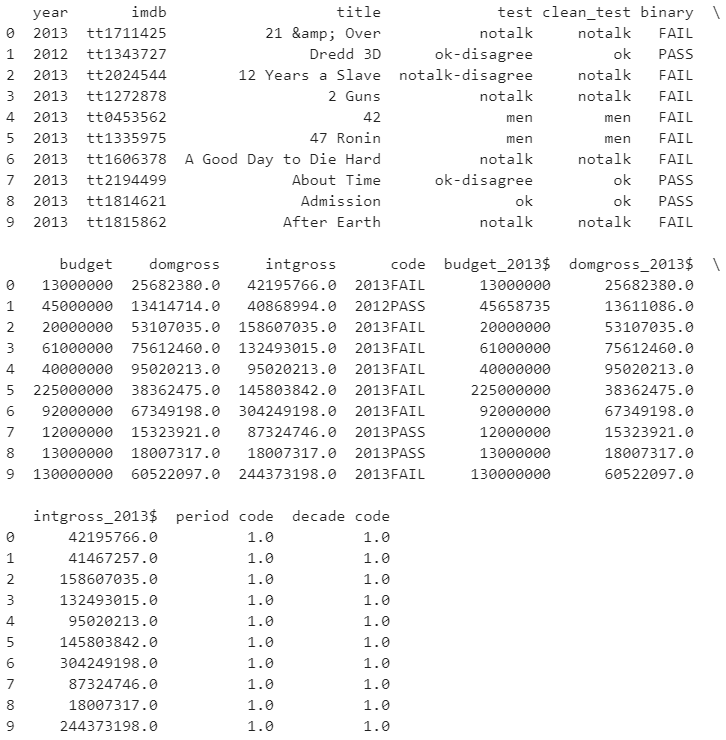
Linear regression algorithm is one of the predictive supervised learning methods applied in machine learning for regression. It is a simple and commonly used algorithm that builds the linear relationship between the dependent variable and one or more independent variables aiming to find the best fit line, that is, minimize the error between predicted and actual values [3][10][17].

**About the dataset chosen for regression**

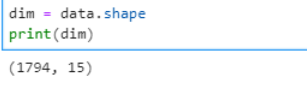
*The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women* uses two dataset to observe the association between the importance of women in a film and that film’s budget and gross profits. The BechdelTest.com which contains the list of movies that were analysed and verified if they pass the Bechdel test and The-Numbers.com that contains the financial information on these movies such as box office and budget data [11]. Thus, the regression algorithm built will verify if there is a linear relationship between the movie’s budget (response variable) and the indicator that the movie passed/failed the Bechdel test and its gross profit (international and domestic).

**Regression algorithm**

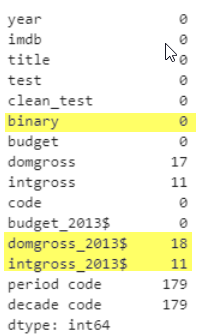
After reading the full data from the csv file into a data frame, I had a quick glance at that to understand how data looked like. It did not show missing values, the continuous features seem to be in the same scale and there are a few categorical features as well, some expressed on the format of text and others as numerals. I am going to use budget\_2013$ as response variable and three features: binary, domgross\_2013$ and intgross\_2013$, so, I kept the focus on analysing the data for those 4 columns only.

****

Next step was to review the dimensions of the data to verify that there was a balance between number of features and number of records.



The length of the dataset is nearly 2,000 records which is fine to apply linear regression for that number of features. I also needed to verify how many NaN values there were for each of the features to be utilised. Note that there were a few NaN values only for domgross\_2013$ and intgross\_2013$ columns.

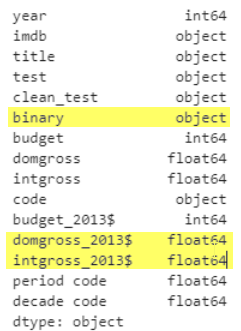


Then, the first step was to assess the impact on the dataset size if the lines with NaN values were removed.

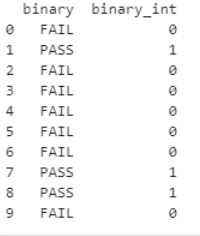


The actual dataset size is 1794 and removing lines with NaN values would reduce the size to 1600 records. The difference was not significant for the number of features and as I would still have enough data, I removed those lines instead of inputting the missing values.

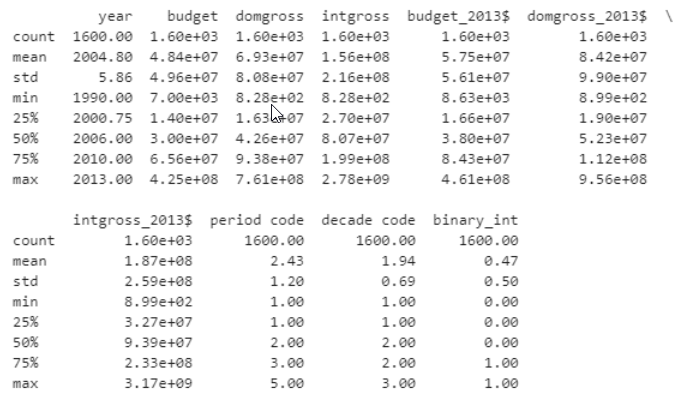
Afterwards, I verified the data type of the features to apply any necessary conversion. Hence, using the *dtypes* function I could see that only the binary feature would need to be converted from ‘object’ (string) to numerical.



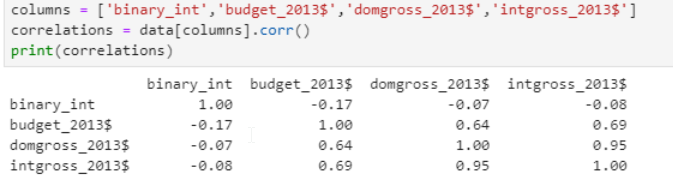
By using the *groupby* function I could identify the different answers for that feature and add a new categorical column binary\_int to the dataset.



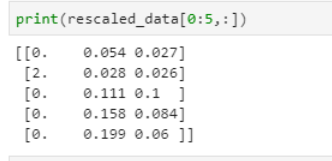
I also had a look at the summary of statistics for the dataset using the *describe* function, which showed that the target variable (budget\_2013$) and features (domgross\_2013$, intgross\_2013$, binary\_int) have no negative or NaN values.



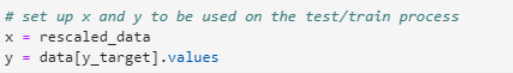
And on the correlation matrix I could see that domgross\_2013$ and intgross\_2013$ had a high correlation but I decided to keep both features to build the model.



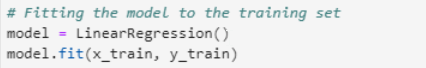
The last step before building the linear regression model was rescaling the data as the feature binary\_int was on a different scale.



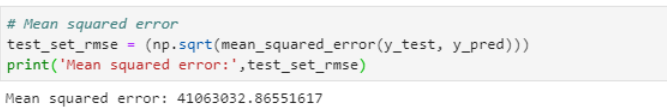
Data was then split in 20% for testing and 80% for training and the final RMSE found for the model was really high (41063032.86) which indicates the model is not good and leads to underfitting error. The model needs to be rebuild using a different set of features or perhaps using just one feature, another type of regression algorithm may also be better suited to this data set.

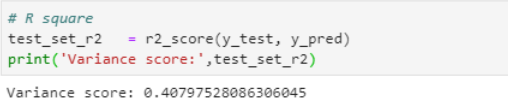












References

[1] Thomson-DeVeaux, A., Mithani, J., Bronner, L. and Lannes, L. (2020) ‘Why Many Americans Don't Vote’, FiveThirtyEight. Available at: <https://projects.fivethirtyeight.com/non-voters-poll-2020-election/> (Accessed: 03 Nov 2020)

[2] ‘Linear Regression vs Logistic Regression’ (2020), Java T Point. Available at: <https://www.javatpoint.com/linear-regression-vs-logistic-regression-in-machine-learning> (Accessed: 15 Nov 2020)

[3] ‘Regression vs. Classification in Machine Learning’ (2020). Java T Point. Available at: <https://www.javatpoint.com/regression-vs-classification-in-machine-learning> (Accessed: 15 Nov 2020)

[4] Brownlee, J. (2020) ‘Develop k-Nearest Neighbors in Python From Scratch’, Machine Learning Mastery. Available at: <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/> (Accessed: 07 Nov 2020)

[5] Maklin, C. (2019) ‘K Nearest Neighbor Algorithm In Python’, Towards Data Science. Available at: <https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55> (Accessed: 07 Nov 2020)

[6] Zoltan, C. (2018) ‘KNN in Python’. Towards Data Science. Available at: <https://towardsdatascience.com/knn-in-python-835643e2fb53> (Accessed: 07 Nov 2020)

[7] Srikanth, P. (2014) ‘Naive Bayes or K-NN for classification?’, Medium. Available at: <https://medium.com/data-science-analytics/naive-bayes-or-k-nn-for-classification-60a4d92e7bab> (Accessed: 10 Nov 2020)

[8] ‘K-Nearest Neighbor(KNN) Algorithm for Machine Learning’ (2020), Java T Point. Available at: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning> (Accessed: 15 Nov 2020)

[9] Kaushik, A. (2020) ‘Introduction to ML’, Dublin Business School. Available at: <https://elearning.dbs.ie/mod/resource/view.php?id=921696>

[10] Hoare, T. (2020) ‘Data Mining: Supervised Learning Classification – Ensemble Models - Regression - Deep Learning’, Dublin Business School. Available at: <https://elearning.dbs.ie/pluginfile.php/1157117/mod_resource/content/0/Revision%20Lecture%201%20-%20Supervised%20Learning%20Classification%20-%20Ensemble%20Models%20-%20Regression%20-%20Deep%20Learning%20%20.pdf>

[11] Hickey, W. (2014) ‘The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women’, FiveThirtyEight. Available at: <https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/> (Accessed: 15 Nov 2020)

[12] Sublett, D. (2019) ‘Data Science Modeling: How to Use Linear Regression with Python’, Medium. Available at: <https://medium.com/better-programming/data-science-modeling-how-to-use-linear-regression-with-python-fdf6ca5481be> (Accessed: 07 Nov 2020)

[13] Pierre, S. (2019) ‘Predicting Missing Values with Python’, Towards Data Science. Available at: <https://towardsdatascience.com/predicting-missing-values-with-python-41e581511117> (Accessed: 07 Nov 2020)

[14] Moreno, A. (2019) ‘Simple and multiple linear regression with Python’, Towards Data Science. Available at: <https://towardsdatascience.com/simple-and-multiple-linear-regression-with-python-c9ab422ec29c?gi=929a4ebc8b98> (Accessed: 15 Nov 2020)

[15] ‘Multivariate Linear Regression in Python with scikit-learn Library’ (2020), Finance Train. Available at: <https://financetrain.com/multivariate-linear-regression-in-python-with-scikit-learn-library/> (Accessed: 15 Nov 2020)

[16] ‘Multiple Linear Regression’ (2020), Java T Point. Available at:

<https://www.javatpoint.com/multiple-linear-regression-in-machine-learning> (Accessed: 15 Nov 2020)

[17] ‘Simple Linear Regression in Machine Learning’ (2020), Java T Point. Available at: <https://www.javatpoint.com/simple-linear-regression-in-machine-learning> (Accessed: 15 Nov 2020)

[18] Loukas, J. (2020) ‘Is your model overfitting? Or maybe underfitting? An example using a neural network’, Towards Data Science. Available at: <https://towardsdatascience.com/is-your-model-overfitting-or-maybe-underfitting-an-example-using-a-neural-network-in-python-4faf155398d2> (Accessed: 18 Nov 2020)

[19] Subrahmanya, S. (2018), ‘Missing Data Imputation using Regression’, Kaggle. Available at: <https://www.kaggle.com/shashankasubrahmanya/missing-data-imputation-using-regression> (Accessed: 14 Nov 2020)

[20] Pierre, S. (2020), ‘Predicting Missing Values with Python’, Towards Data Science. Available at: <https://towardsdatascience.com/predicting-missing-values-with-python-41e581511117> (Accessed: 14 Nov 2020)

[21] Bilogur, A. (2018) ‘Missingno: a missing data visualization suite’. Journal of Open Source Software, 3(22), 547, <https://doi.org/10.21105/joss.00547> (Accessed: 14 Nov 2020)

[22] Brownlee, J. (2020) ‘How to Use StandardScaler and MinMaxScaler Transforms in Python’, Machine Learning Mastery. Available at: <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/> (Accessed: 14 Nov 2020)

[23] Nelson, D. (2020) ‘Multiple Linear Regression with Python’, Stack Abuse. Available at: <https://stackabuse.com/multiple-linear-regression-with-python/> (Accessed: 15 Nov 2020)