MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCE

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

BACHELOR OF SOFTWARE ENGINEERING

COURSE UNIT: BSE2301 SOFTWARE ENGINEERING MINI PROJECT 2

SUPERVISOR: MR.KAMULEGEYA GRACE

GROUP F (DAY CLASS)

NAME	REGISTRATION NUMBER	EMAIL 1D
SEMPIIRA ISAAC	18/U/854	sempiiraisaac100@gmail.com
NABUNJE DIANA LUBEGA	18/U/23409/EVE	dianalubegan@gmail.com
DALI HILLARY	18/U/21102/PS	dalirichardh@gmail.com
BAKABULINDI MARVIN	18/U/21085/PS	marvinbakabulindi@gmail.com

UCE RESULTS 2011-2016 DATASET ANALYSIS FINAL PROJECT REPORT.

This article is organized as follows; in our sections we will describe the project statement and objectives, we describe a sample solution to the problem including the dataset, code, and the outputs. We present a sample project report for the **UCE RESULTS 2011-2016 dataset with shape (17639, 48).**

Problem Statement.

The challenge is to analyze and evaluate the general performance of schools of Uganda in different districts. This covers trying to evaluate the education system of Uganda as a country at large. In this we shall **find out the consistent-best performing schools and districts, schools trying to improve their performance and those still lagging behind.**

Project instructions.

In this problem, we will forecast the outcome of the analysis. Techniques of data science and predictive analytics are used to predict the probabilities (outcomes). We were **aided by different libraries such pandas, numpy, matplotlib, seaborn, sklearn plus the machine learning algorithms.**

ABOUT THE DATA.

This section describes the columns that we have in our dataset and the data it contains.

YEAR- Represents the years from 2011-2016 and each year has certain data attached to it.

DISTRICT- Represents particular districts to which the schools belong and these have been given names.

SCHOOL- This column includes all school names.

TOTAL CANDIDATES- This column includes the total number of females and males that sat for the exams

TOTAL FOR DIV (1-9)-These are different columns containing the total of students whose scored from division 1-9.

% DIV (1-9) - These are columns stipulating the percentages for the totals of each division from 1-9.

TOTAL X-Represents the number of students whose results did not come back.

% X- Represents the percentage for the total of those whose results did not come back.

FEMALE CANDIDATES- This includes the total number of females who sat for the exams in different schools per year.

FEMALE TOTAL FOR DIV (1-9) - These are columns including the total number of females who scored according to different grades from the first division to the last.

FEMALE % DIV (1-9) - They indicate the percentages of the total of females per the grade and that is from division one to nine.

MALE CANDIDATES- This column includes data about the total number of males who sat for the exams in different schools per year.

MALE % DIV (1-9)-These are columns including the percentages for the total number of males who scored in different grades.

- Going through the csv file (UceResultsBy2011-2016.csv), you notice that the numbers are in decimal especially for the percentages and others in integers for the columns for getting the total number of students. For situations where the total was estimated in decimal, rounding off was taken to get a whole number.
- In other columns for example SCHOOL, DISTRICT the data was entered as text not numbers.
- You will notice that column headers had white space between them and we had to fix this when opening and reading the CSV file by replacing the white spaces with an underscore.

IMPORTS.

For this project, we set up a working environment and we are using libraries say pandas, numpy, matplotlib, seaborn and sklearn that we imported.

READ THE DATA.

We made sure that the CSV file is in our working directory. We then opted for reading the file using the pandas library, treated missing values as null values and even setting the display of a desired view of our dataset.

VIEW THE DATA.

After reading our dataset, we realized it contains:

- No. of data points=17639
- No. of columns=48
- Checking for the data types – we used the statement (df.dtypes ()) to find out and this shows that 43 had (float 64), 3 had (int 64) and 2 had (object) data type.
- Checking for columns with Null or Nan values if any. We used (df.isna().any ()) and this returns a Boolean expression either TRUE or FALSE;

FIG1.DATA FRAME BEFORE CLEANING

YEAR	False	%_DIV_1	False
DISTRICT	False	TOTAL_DIV_2	True
SCHOOL	True	%_DIV_2	False
TOTAL_CANDIDATES	True	TOTAL_DIV_3	True
TOTAL_DIV_1	True	%_DIV_3	False

```
TOTAL_DIV_4              True
%_DIV_4                 False
TOTAL_DIV_7              True
%_DIV_7                 False
TOTAL_DIV_9              True
%_DIV_9                 False
TOTAL_X                  True
%_X                     False
FEMALE_CANDIDATES       False
FEMALE_TOTAL_DIV1        True
FEMALE_%_DIV1            True
FEMALE_TOTAL_DIV2        True
FEMALE_%_DIV2            True
FEMALE_TOTAL_DIV3        True
FEMALE_%_DIV3            True
FEMALE_TOTAL_DIV4        True
FEMALE_%_DIV4            True
FEMALE_TOTAL_DIV7        True
FEMALE_%_DIV7            True
FEMALE_TOTAL_DIV9        True
FEMALE_%_DIV9            True
FEMALE_TOTAL_X           True
FEMALE_%_X               True
MALE_CANDIDATES         False
MALE_TOTAL_DIV1          True
MALE_%_DIV1              True
MALE_TOTAL_DIV2          True
MALE_%_DIV2              True
MALE_TOTAL_DIV3          True
MALE_%_DIV3              True
MALE_TOTAL_DIV4          True
MALE_%_DIV4              True
MALE_TOTAL_DIV7          True
MALE_%_DIV7              True
MALE_TOTAL_DIV9          True
MALE_%_DIV9              True
MALE_TOTAL_X             True
MALE_%_X                 True
dtype: bool
```

- **No. of rows with null values were 14511.**

PROJECT OBJECTIVES.

Below are the objectives of the project;

- To **monitor the General performance in the country** over year[s] with graphical visualizations where possible.
  - Trend of the number of students who sit exams per the year.
  - A year with maximum total number of candidates sitting UCE.
  - Trend of total UCE candidates in each division per year.
  - Determine years in which a highest number of each division was attained.
  - Number of schools per district.
  - Total candidates per district.
  - Overall district candidates per school.
  - Determining districts which contributed to the worst performance.
  - Rank schools by percentage division 1.
  - Determine consistent schools maintaining positions among top 20 schools.
  - Determine the percentage division 1 per district.
  - Determine the correlation between the percentage division 1 per district and number of schools per district.
  - Describe the dataset statistical measures of mean, median, mode, std and percentiles.
  - Number of districts in the country.
  - Number of Schools in the country.
- To **examine gender based performance in the country** over year[s] with graphical visualizations where possible.
  - Identify a year with highest number of female and male UCE candidates.
  - Total number of females per division over the years.
  - Total number of males per division over the years.
  - Compare females and males division 1 and 2 over the years.
  - Accumulate totals of males and females in division 1 and 2 since 2011 to 2016.
  - Determine top 10 districts where females failed mostly over the years.
  - Determine top 10 districts where males failed mostly over the years.
  - Determine top 10 districts which educated the highest number of females accumulated since 2011 to 2016
  - Grand percentage of female and male UCE candidates educated since 2011 to 2016

- To **demonstrate data preprocessing, data training, and predictive model techniques** of analyzing UceResultsBy2011-2016.csv dataset to deeper insights by **establishing the relation between predicted and actual percentage grades.**

FEATURES (COLUMNS) ANALYZED.

All data set features where analyzed and **different sets of them where used to achieve specific objectives and also used to generate new features**.

PROCESSES AND TECHNIQUES APPLIED.

- DATAFRAME CLEANING PROCESS.

  Data frame before cleaning represented in FIG1.

  Steps and techniques applied;
  - Strip columns using strip() to **remove trailing spaces** on column names.
  - Replace spaces in column names with an underscore.
  - Removing duplicates if any.
  - Inspect data frame null values as shown in FIG1.
  - Remove rows without school names as these cannot be fixed.
  - Fill missing Total Candidates with sum of female and male candidates as these are not null as shown in FIG1.
  - Total Candidates values which are not equal to the sum of female and male candidates are **replaced with sum of female and male candidates** then total candidates column will have no null values.
  - Procedures followed in the **computational logic based on mathematical arithmetic, to deal with NaN values in the dataset without making assumptions, is as described below;**
    - Generated array of grades [1, 2, 3, 4, 7, 9, X].
    - Obtained coefficients of key sections (TOTAL, MALE, and FEMALE sections for each row)

```
# Getting coefficient of key Sections.
for row in df.index:
    total_coefficient = df.at[row,"TOTAL_CANDIDATES"] / 100
    male_coefficient = df.at[row,"MALE_CANDIDATES"] / 100
    female_coefficient = df.at[row,"FEMALE_CANDIDATES"] / 100
```

    - For each grade [1, 2, 3, 4, 7, 9, X], obtain its total, percentage, maleTotal, malePercentage, femaleTotal, and femalePercentage.
    - Whenever the total is 0.0, fill the remaining related fields for that grade with 0.0.
    - Whenever the percentage grade is 0.0% replace its corresponding NaN grade with 0.
    - Whenever the grade is 0.0 replace its corresponding NaN percentage grade with 0.
    - For non-zero values, we use the calculated coefficients. For-example;
      I. If total has a value and total percentage is missing:
      (total-percentage = total/total-coefficient).
      II. If total percentage has a value and total is missing:
      (total = total-coefficient*total-percentage).

- Applying subtraction from Total percentages, to fill the missing males and females percentages. For-example;

I. If female total percentage missing and male percentage and total percentage known: **(female total percentage = total percentage- male percentage).**

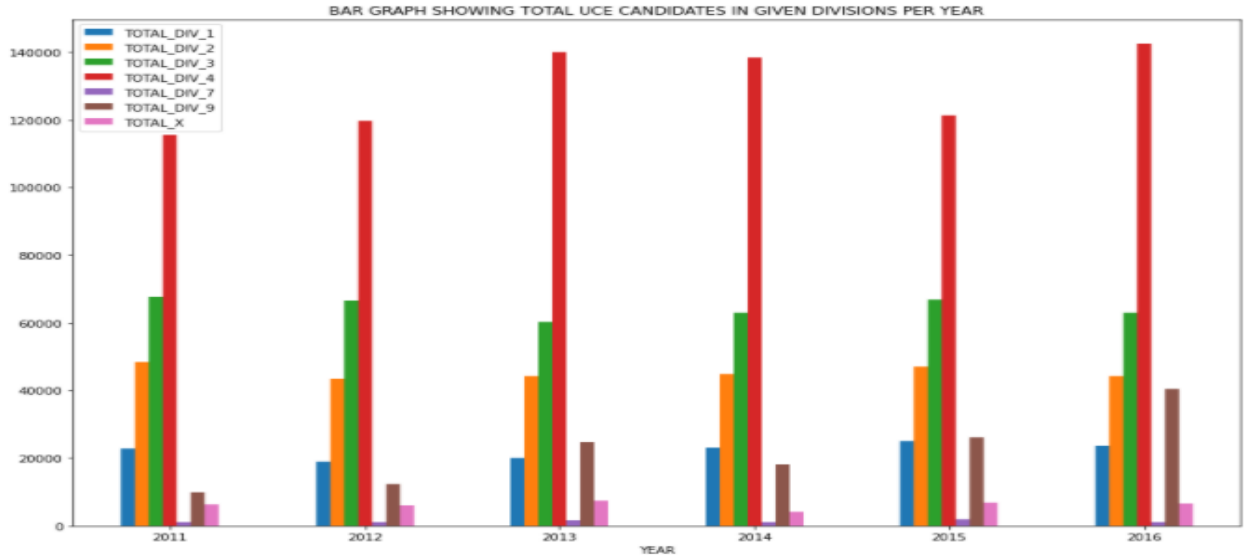
- This **process eliminated all null value**s in our data frame as shown below:

FIG2. DATAFRAME AFTER REMOVING NULL VALUES

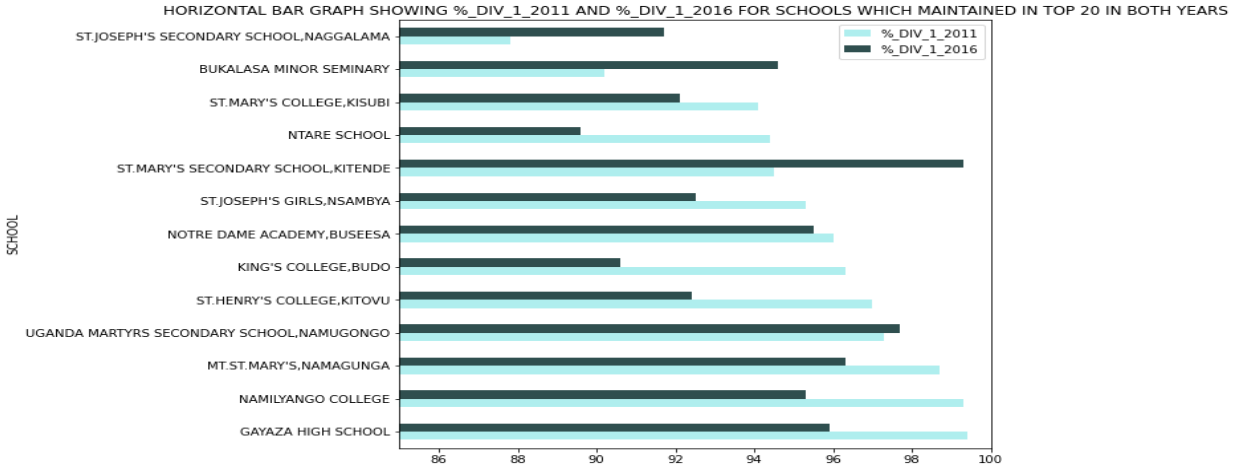
```
YEAR                    False
DISTRICT                False
SCHOOL                  False
TOTAL_CANDIDATES        False
TOTAL_DIV_1             False
%_DIV_1                 False
TOTAL_DIV_2             False
%_DIV_2                 False
TOTAL_DIV_3             False
%_DIV_3                 False
TOTAL_DIV_4             False
%_DIV_4                 False
TOTAL_DIV_7             False
%_DIV_7                 False
TOTAL_DIV_9             False
%_DIV_9                 False
TOTAL_X                 False
%_X                     False
FEMALE_CANDIDATES       False
FEMALE_TOTAL_DIV1       False
FEMALE_%_DIV1           False
FEMALE_TOTAL_DIV2       False
FEMALE_%_DIV2           False
FEMALE_TOTAL_DIV3       False
FEMALE_%_DIV3           False
FEMALE_TOTAL_DIV4       False
FEMALE_%_DIV4           False
FEMALE_TOTAL_DIV7       False
FEMALE_%_DIV7           False
FEMALE_TOTAL_DIV9       False
FEMALE_%_DIV9           False
FEMALE_TOTAL_X          False
FEMALE_%_X              False
MALE_CANDIDATES         False
MALE_TOTAL_DIV1         False
MALE_%_DIV1             False
MALE_TOTAL_DIV2         False
MALE_%_DIV2             False
MALE_TOTAL_DIV3         False
MALE_%_DIV3             False
MALE_TOTAL_DIV4         False
MALE_%_DIV4             False
MALE_TOTAL_DIV7         False
MALE_%_DIV7             False
MALE_TOTAL_DIV9         False
MALE_%_DIV9             False
MALE_TOTAL_X            False
MALE_%_X                False
dtype: bool
```

- Removed repeating schools using **unique(), split(), and strip() techniques.**
- Removed repeating districts due to trailing spaces and different names for the same district like Masaka and Masaka Main, using unique() and split() techniques.

- DATA FRAME ANALYSIS.
  - Generated total candidates per year.
  - Generated a year with **maximum total number of candidates sitting UCE and that was 2016, hence sampled out for further analysis.**
  - Generated trend of total UCE candidates in each division per year and **year 2011 registered the best performance hence also used in the proceeding analysis.**

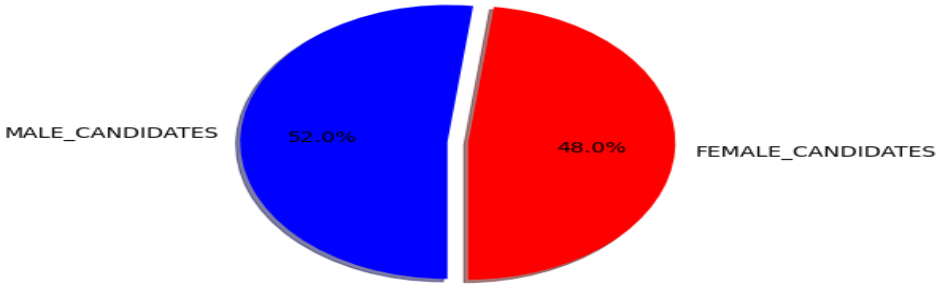


- Generated years in which a highest number of each division was attained.
- Generated number of schools per district in year 2016.
- Generated total candidates per district in year 2016.
- Generated overall district candidates per school in year 2016.
- Generated districts which contributed to the worst performance in year 2016.
- Ranked schools by percentage division 1 in both 2011 and 2016.
- Generated **consistent schools maintaining positions among top 20 schools in both 2011 and 2016.**



- Generated the percentage division 1 per district in year 2016.
- Generated the **correlation between the percentage division 1 per district and number of schools per district basing on year 2016 and its regression curve plotted.**
- Described the dataset statistical measures of **mean, median, mode, std and percentiles.**
- Generated number of districts in the country as per year 2016.
- Generated number of Schools in the country as per year 2016.
- Identified a year with highest number of female and male UCE candidates.
- Generated total number of females per division over the years.
- Generated total number of males per division over the years.
- Compared females and males division 1 and 2 over the years.
- Accumulated totals of males and females in division 1 and 2 since 2011 to 2016.
- Generated top 10 districts where females failed mostly over the years.
- Generated top 10 districts where males failed mostly over the years.
- Generated top 10 districts which educated the highest number of females accumulated since 2011 to 2016
- Generated **grand percentage of female and male UCE candidates educated since 2011 to 2016**

PIE CHART SHOWING CUMMULATED TOTAL NUMBER OF FEMALES AND MALES SINCE 2011 TO 2016



- DATA PREPROCESSING AND MODELING

  In any machine learning environment, there are basic **steps of approaching any data science** problem as stated below;

  1. Gathering data
  2. Preparing that data
  3. Choosing a model
  4. Training
  5. Evaluation
  6. Hyper parameter tuning
  7. Prediction

  In this section, am going to be going through steps 3 to 7 above.

  Before choosing any model to use and passing in data for it to use, it is always important to do some **feature selection.** Not necessarily every column (feature) is going to have an impact on the output variables in our problem.
  If we add these **irrelevant features in the model, it will just make the model worse** (Garbage In Garbage Out). This gives rise to the need of doing feature selection.
  Proposed Features are: ***Year, District, School, Total_Candidates, Female_Candidates, Male_Candidates.***
  Feature selection can be done in multiple ways but there are broadly 3 categories of it:
  1. Filter Method
  2. Wrapper Method
  3. Embedded Method

  In this problem we used the **Embedded Method.**

  Embedded methods are iterative in a sense that, it takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration.

  Regularization methods are the most commonly used embedded methods which **penalize a feature given a coefficient threshold.**

  Set Targets are: ***percentages of female and male divisions from 1 to X.***

NB: *Before passing the train features to the model(), we had to first handle the categorical data i.e. Districts, Schools since this was represented as* ***strings thus need to change them to integer.***

	YEAR	DISTRICT	SCHOOL	TOTAL CANDIDATES	FEMALE CANDIDATES	MALE CANDIDATES
0	2011	WAKISO	GAYAZA HIGH SCHOOL	176	176	0
1	2011	MUKONO	NAMILYANGO COLLEGE	151	0	151
2	2011	MUKONO	MT.ST.MARY'S,NAMAGUNGA	153	153	0
3	2011	WAKISO	UGANDA MARTYRS SECONDARY SCHOOL,NAMUGONGO	222	113	109
4	2011	BUSHENYI	KITABI SEMINARY	73	0	73
...	...	...	...	...	...	...
17634	2016	ZOMBO	PAIDHA SECONDARY SCHOOL	148	44	104
17635	2016	ZOMBO	PAKADHA SEED SECONDARY SCHOOL	68	28	40
17636	2016	ZOMBO	CHARITY COLLEGE,PAIDHA	128	24	104
17637	2016	ZOMBO	NEGRINI MEMORIAL SECONDARY SCHOOL	45	14	31
17638	2016	ZOMBO	JANGOKORO SEED SECONDARY SCHOOL	50	28	22

17638 rows × 6 columns

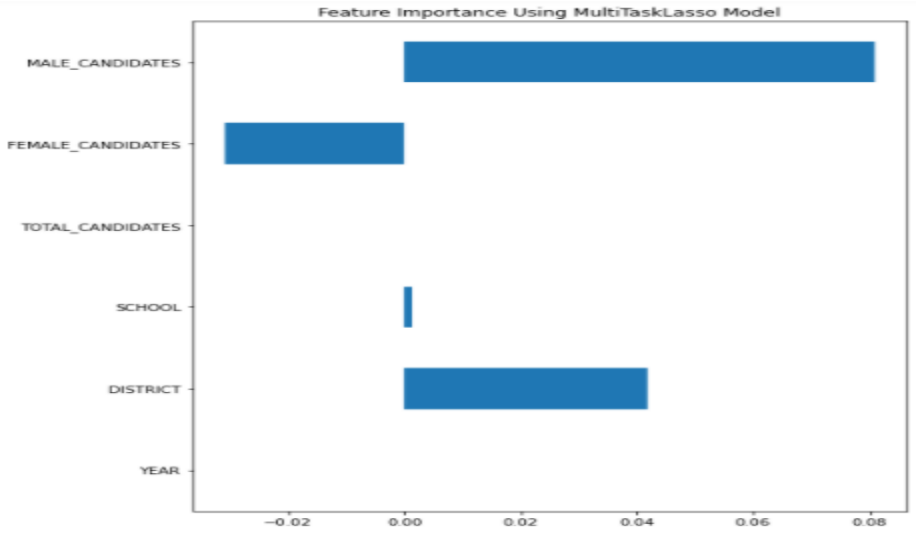
A picture showing data before applying LinearEncoder on the District and Schools Column

	YEAR	DISTRICT	SCHOOL	TOTAL CANDIDATES	FEMALE CANDIDATES	MALE CANDIDATES
0	2011	111	772	176	176	0
1	2011	85	2147	151	0	151
2	2011	85	1966	153	153	0
3	2011	111	3347	222	113	109
4	2011	21	1458	73	0	73
...	...	...	...	...	...	...
17634	2016	113	2403	148	44	104
17635	2016	113	2406	68	28	40
17636	2016	113	570	128	24	104
17637	2016	113	2205	45	14	31
17638	2016	113	998	50	28	22

17638 rows × 6 columns

A picture of District and School in integer format after applying LinearEncoder

To achieve this, ***LinearEncoder*** *class was used. This works by calculating the hash of each string and it will assign the same integer to any string(s) that produce the same hash thus distinct integers for the different strings.*



A picture of the scores given to the proposed features

As observed above, ***Year and Total Candidates are not suitable Features for Modeling, thus eliminated since they get a score of 0.***

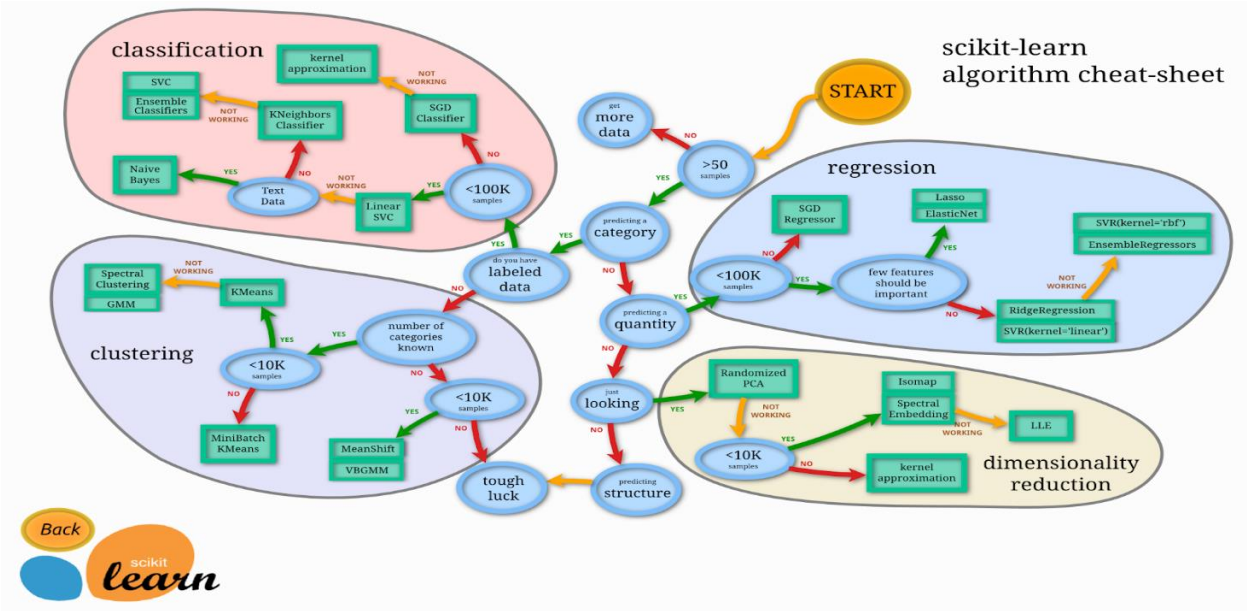
Model Results:

```
Best alpha using built-in MultiTaskLassoCV: 11.975472
Best score using built-in MultiTaskLassoCV: 0.040145
```

Choosing a model to use

Different models solve different problems, and some are not fit to solve certain problems for example the one we are solving.

To **be effective and optimal, we made use of the sklearn roadmap below:**



From the fact that we were aiming to get the number **(quantity)** of grades given 17638 records **(<100K)** and supposed to use all the remaining features, then as per roadmap, **an ensemble regressor was best fit** and in our case (**GradientBoosting**) **was used to get maximum performance.**

Note: *GradientBoosting regressor does* ***not support Multioutput targets.***

To ***solve this, the MultiOutputRegressor was used and the ensemble regressor used as an estimator****. This runs multiple instances of the Gradient Boosted regressor to produce multiple targets thus solving the single target problem.*

```
from sklearn.multioutput import MultiOutputRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

#split dataset into test and training data
xtrain, xtest, ytrain, ytest=train_test_split(X, y, test_size=0.10)

#training the model
gbr = GradientBoostingRegressor()
 #create multiple parallel instances of the estimator(Gradient Boosting Regressor)
model = MultiOutputRegressor(estimator=gbr).fit(xtrain,ytrain)

#determine Score of model in approximating targets
score = model.score(xtrain, ytrain)
print("Training score: ", score)
```

```
Training score:  0.27040847392603107
```

As seen above, ***on training the model, a score of 0.27040847392603107 was obtained.***

- The model was tested with some data and the **Mean Squared Errors (MSE) obtained.**
- **Relation between predicted and actual percentage grades is established using the scatter plots with lines of best-fit.**
- The **model worked well in some targets and yet in others, not good at all**.

CONCLUSIONS FROM THE DATASET

- Best general performance observed in 2011.
- High failure rate observed in 2016.
- Highest number of UCE candidates where in 2016 (**321,254 candidates**).
- Highest total number of division one where attained in 2015 amounting to **25,081**.
- Wakiso has the highest number of schools totaling to **352 schools**.
- Wakiso, Kampala, Mukono, Mbale, and Jinja in that order, had the highest number of candidates amounting to **33220, 20739, 12277, 8161, and 8019** respectively.
- Wakiso, Mbale, Iganga, Kasese, and Kampala, in that order, leads the top districts that contribute to the worst performance in the country.
- Majority of the students passed in division four.
- Gayaza High School, Namilyango College and MT.ST.Mary's, Namagunga are the **top three best schools** which maintained their performance in the top 20 schools for both 2011 and 2016.
- Mukono, Wakiso, Kampala, Mbarara, and Bushenyi, in that order, are the **top best districts in the country, ranked by percentage of division one per district.**
- As Number of schools per district increase, percentage division one per district is more likely to also increase.
- Number of Districts in the country by year 2016 amounts to **112 districts**.
- Total number of schools in the country by year 2016 amounts to **3255 schools**.
- Year 2015 had almost **equal number of female and male** UCE candidates with females and males totaling to **142831 and 149432** respectively.
- However referencing with year 2015, still the male gender registered best performance over the females.
- Year 2016 had the maximum number of both female and male candidates totaling to **157705 and 163549** respectively.
- Wakiso, Mbale, Kampala, in that order, where the **top three districts where females failed most.**
- Wakiso, Kampala, Mbale, in that order where the **top three districts where males failed most.**
- Wakiso, Kampala, Mukono, in that order **educated the highest number of females accumulated since 2011 to 2016.**
- Male gender registered the overall best performance in all years.
- Grand percentage total number of UCE certified females and males accumulated since 2011 to 2016 are as follows;
  - **MALE: 52.0%**
  - **FEMALE: 48.0%**

  Males still in the lead!
- Providing the input features for the designed model, the following can be **accurately predicted with minimal error margin:**

  `FEMALE_%_DIV7 MSE: 3.4432 and MALE_%_DIV7 MSE: 2.3692`