Predicting Teenage Birthrates in UN Countries

Project Writeup

Shonte Amato-Grill
Jacob Preston

I. Introduction

A large portion of UN activity and funds are regularly spent on outreach programs to nations in dire need of assistance. Many of these are targeted towards high-density population areas with low average GDP per capita. The UN decides upon such locales by gathering data on each of the countries, whenever possible, and analyzing the data to see where to, say, place a new school for young children.

The particular need addressed in this project was teen pregnancy. Teenage births are a significant burden on the mother of the child and it will hinder her life much more than if she had been able to give birth several years later. Each year, roughly 7.3 million girls give birth around the world. Problems persisting from teenage births are lack of education, inability to work and a reliance on men for sustenance. All of these diminish women's rights and the quest toward equality amongst men and women. Providing adequate resources for areas with high teenage birth rates could help to alleviate resulting difficulties in the region, stabilizing literacy rates, and male to female ratios in the labor force, to name two.

1.1 Goals

There is, however, a fundamental flaw in this method of outreach program: by the time a severe GDP/birth rate discrepancy is witnessed, the act of building a school to raise literacy rates (which generally are inversely proportional to the GDP/birth rate discrepancy) is *retroactive*. The goal, as set out in the project proposal, is as follows:

1. **to determine if there were predictive features in determining the teenage birth rates in UN countries** (See *Data Architecture, 2.1*)**, and**
2. **to construct an as-accurate-as-possible model across all UN countries, in an attempt to prove the viability of a method for "preemptive assessment" of potential outreach program locales.**

After taking a closer look at the data, it was decided, however, that a clustering sub problem could provide us with information about the countries as well. Countries were given an ID based on geographic local and then examined to see if clusters of regions would form based on our feature set. This could be used to determine different tiers of countries based on our data.

As the project progressed, it became clear that the predictive models would be insufficient. There was enormous difficulty with the UN data sets as they not only had to be pieced together, but there existed significant data gaps and a general lack of features to be able to use any regression algorithms. They were found to have a wide range of error due to not being able to adequately fracture the data and that it would be a better prediction to simply guess at the future teen pregnancy rates using past averages for the country in question.

However, the sub problem emerged as a more interesting model. While it was not possible to adequately determine whether the features selected were predictive of birth rates, it was shown that those features allowed countries in similar regions of the world to be clustered together with like countries, without using outputs as a feature. These could be grouped in a system of tiers and it was easy to see that certain regions needed more work than others.

1.2 Assumptions

It was necessary to make some basic assumptions regarding the algorithms, data, and models to begin learning.

1. All models would be trained across all countries and years. Values for *Country A* will be predicted using the same model as those for *Country B*.

II. Data

Following is a description of the data, methods used for manipulation, etc.

2.1 Data Architecture

The data consists of several features for each country. Datapoints are keyed using the following:

- Country
- Year

Each datapoint contains the following features:

- Minimum age of marriage without parental consent (Female)
- Minimum age of marriage without parental consent (Male)
- Literacy Rate (Female)
- Literacy Rate (Male)
- Contraceptive Prevalence (Modern Methods)
- Contraceptive Prevalence (Any Method)
- Teen marriage Percentages (Female)
- Teen marriage Percentages (Male)
- Mean Marriage Age (Female)
- Mean Marriage Age (Male)
- Share of Labor Force (Female)
- Gender Ratio

The continuous output (classification) is recorded as column 14 in the dataset (comma delineated) and is

- Teenage Births as a Percentage of Total Female Population ages 15-19

2.2 Data Gaps

2.2.1 General Data Patching

The data was pulled from the UN online archives (located at UN.gov). Due to the design of said archives, however, each feature was located in distinct databases, making the final datasheet a compilation of appended files. While several of the features were located together in the UN archives (Example: *Literacy Rate (Female)* and *Literacy Rate (Male)*), the majority were separated, necessitating a stitching process.

Because of said process, however, and because the data was not uniformly entered, there were introduced into the dataset many missing features. When the stitching process was complete, the data at several locations was entirely inadequate for analysis, looking something like FIGURE 2.2.1.

Some of the best methods for filling gaps were found by examining the data and understanding the relationship of the feature in the current year to prior and future years that had filled out features. For example, minimum age for marriage without consent would not change year to year and thus could be filled out for all years for a particular country.

Country	Year	Mean Marriage Age (Male)
Algeria	1985	x
Algeria	1986	x
Algeria	1987	23.7
Algeria	1988	x
Algeria	1989	x
Algeria	1990	x
Algeria	1991	x
Algeria	1992	25.9

Figure 2.2.1: *Example of data before gap-filling routines showing one feature, between years*

Country	Year	Mean Marriage Age (Male)
Algeria	1985	23.16
Algeria	1986	23.43
Algeria	1987	23.7
Algeria	1988	24.14
Algeria	1989	24.58
Algeria	1990	25.02
Algeria	1991	25.46
Algeria	1992	25.9

Figure 2.2.2: *Example of data post gap-filling routines showing one feature, between years 1985 and 1992. Green cells were pre-existing data from source, yellow indicates a computed value.*

For other features that had gaps but had two years filled out in the same country, it was easy to observe a linear relationship. It was decided that the best method for filling gaps, similar to those shown in FIGURE 2.2.1, was to create a non-learning algorithm that simply analyzed (within each country), entries that were non-empty and used the two to find a general pattern. For instance, it can be inferred by observing FIGURE 2.2.1 that the *Mean Marriage Age (Male)* between years 1987 and 1992 is increasing based on the two entries that are non-empty. By this pattern, many of the missing feature values could be filled with minimal data distortion, leading to a larger (if slightly less accurate) dataset, and thusly, better predictive capability. FIGURE 2.2.2 demonstrates the same datapoints after patching (green cells are preexisting values, yellow are computed).

There existed some limitations still to this method. For example, countries with only 1 entry of the particular feature could not be filled out linearly. This left many points still not filled in and thus had to be filled in via nearest neighbor patching or cut out entirely.

2.2.2 Output Data Patching

Due to the nature of the data compilation, it became apparent that many of the output values of the dataset were likewise missing. Calculating these, however, could not be performed by the method described in **Section 2.2.1** without drastically reducing the accuracy of the dataset, due in large part to the greater quantity of missing values.

There did exist further datasets which included teenage births per 1000 females between 15-19 years of age. In order to use this data, a script was written that performed a relatively simple mathematical function upon the aforementioned dataset and another detailing population statistics for each country and year.

$$\text{Teenage Births} = \frac{\text{15-19 year old births per 1000 women}}{1000} * \text{15-19 Female Total Population} \quad (1)$$

Equation 1 allowed the dataset to be filled out more completely, rendering a set of about 1000 entries. Having complete outputs, it was necessary to backcheck these with the entries given by the original UN datasets. Since the entries matched with very high accuracy, this method was considered to be an effective method for generating outputs.

2.3 Nearest Neighbor Patching

Even with the patching linear patching methods used, there remained many data gaps and therefore a Euclidean distance nearest neighbor algorithm was attempted in order to fill in these gaps. However, steps were taken to ensure relatively accuracy of the prediction. Data points with too many features missing were

thrown out because it would mess up the data integrity. Therefore, any data point missing more than 2 features was thrown out.

Running the nearest neighbor algorithm allowed for 200 more data points to be generated for a larger set, but after examining the data by hand, the nearest neighbor was not a great predictor of gaps and would sometimes generate confusing results. It was decided to test both data sets whenever an algorithm was run and in most cases, the smaller normal data set proved to be more accurate. Only in the case of the random forest algorithm was the nearest neighbor data set a better for prediction accuracy than the normal set. This was probably due to the VC-dimension of the linear models, discussed later in section 3.3.5.

2.4 Data Visualization via Clustering and PCA

To determine the effectiveness of potential data clustering, dimensionality reduction was performed upon the complete dataset, and the results graphed by the first three principal components. FIGURE 2.4.1 demonstrates the unclustered data, visualized by these first three components. Evidently, the data could quite nicely be clustered, at least into three clusters.

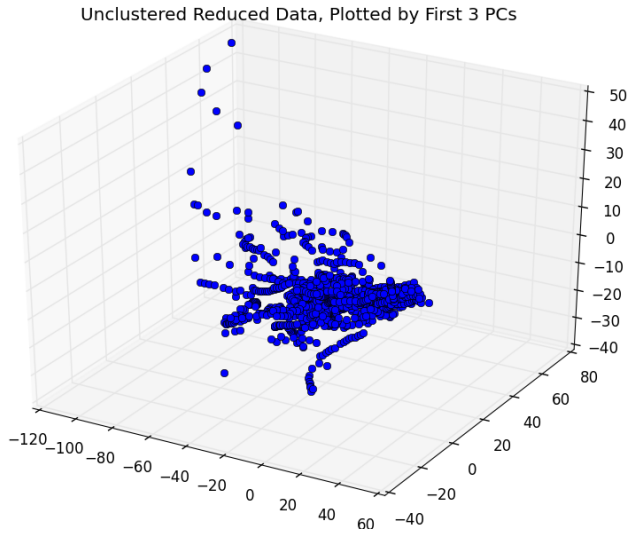


Figure 2.4.1: Unclustered data, plotted by first 3 principle components.

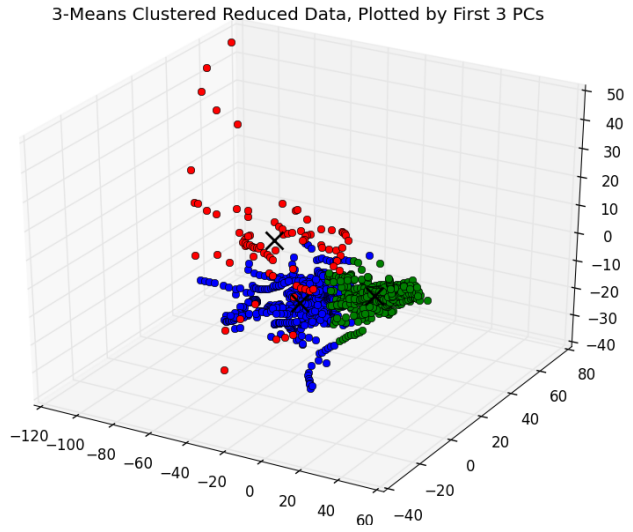
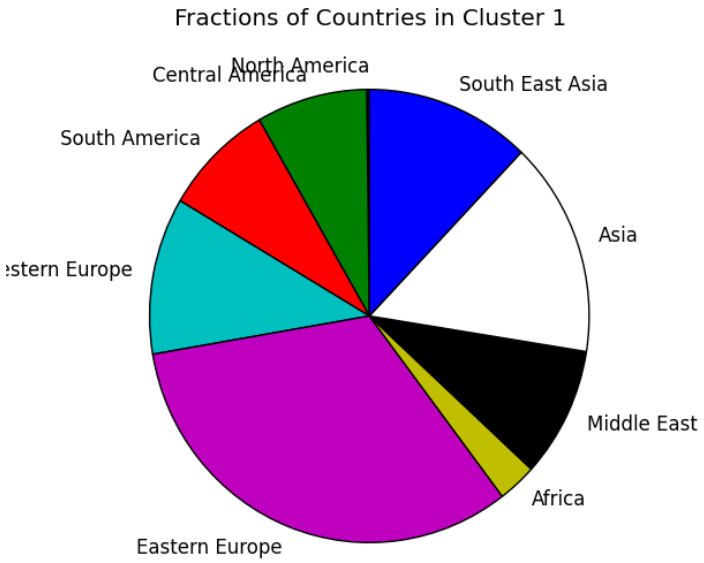
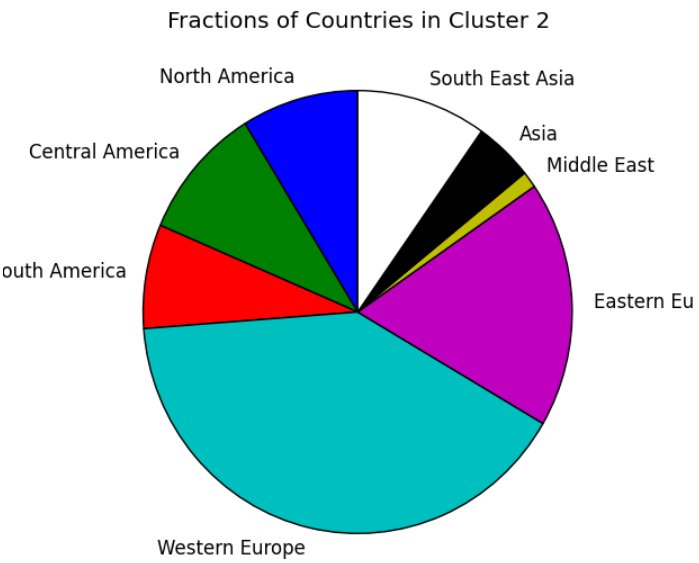


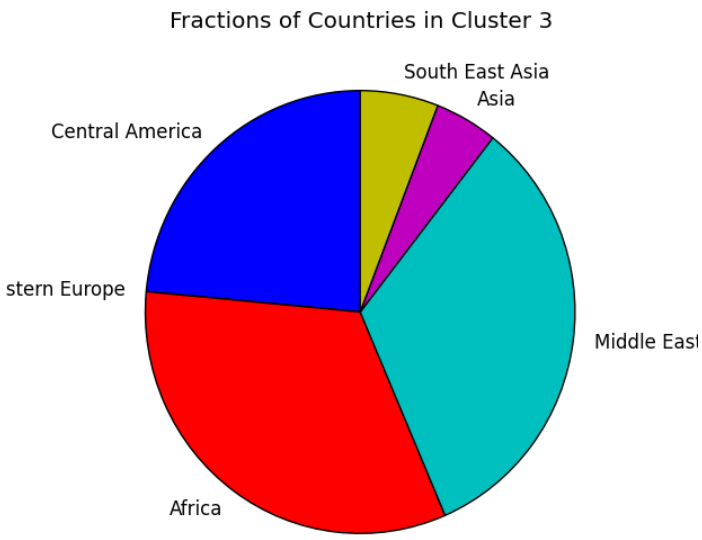
Figure 2.4.2: 3-Clustered data, plotted by first 3 principle components.

Clustering was rather successful, at least by regarding the three dimensional visualization of the PCA reduced clustered data, FIGURE 2.4.2. However, to scry any interesting relationships within clusters, a new method of analysis (other than visualization) had to be developed. As such, manually moving through the dataset, each country was given an *Area Code*, as follows:

0 : 'North America',
1 : 'Central America',
2 : 'South America',
3 : 'Western Europe',
4 : 'Eastern Europe',
5 : 'Africa',
6 : 'Middle East',
7 : 'Asia',
8 : 'South East Asia'







NOTE: These codes were **not** used for clustering, and neither was output. Only features detailed in section 2.1 were used for clustering.

It was now possible to take a more analytical look at the contents of each of the three clusters: by tallying up the number of datapoints from each area that fell into each cluster, it was possible to determine which countries were most similar.

The analysis was much as would be expected: Most datapoints with the same *Area Code* fell into the same clusters. 'North America' and 'Western Europe' fell largely together and so forth.

2.5 Data Constrained

For many of the algorithms used, there was simply not enough data to provide accurate-enough validation error. This led to the possibility that it might have to be conceded that **the selected features** (See *Data Architecture, 2.1*) **were not predictive of teenage birth rates**. To verify this, k-means cross validation is implemented to find average validation error against k-randomly selected validation sets. Success! Well, at least conditionally. A rather large increase in validation error was noted while using cross-validation. (See *Error Calculation, 3.1*)

III. Algorithms

Following is a description of all the algorithms used. When necessary, mathematical analysis is provided for algorithmic methodologies.

3.1 Error Calculation

Before discussion effectiveness of specific algorithms, it is necessary to establish a continuous method of error calculation and analysis. For continuous prediction, some method of confidence margin is usually implemented to allow for slight differences between validation outputs and predicted outputs. Considering each of the outputs is already normalized as a **percentage of total female population ages 15-19**, we can specify a confidence level of a decimal value corresponding to a percent. So, for instance, a .5 error-margin would indicate a 50% confidence margin.

Feeding the predicted outputs and the actual outputs (both as percentages), the error computation method performs the following equation, where .

$$\text{Misses} = \sum_{i=1}^{N} \text{Error}_i \begin{cases} 1 \; if \; \frac{|\text{prediction} - \text{actual}|}{\text{actual}} > \text{error margin} \\ 0 \; if \; \frac{|\text{prediction} - \text{actual}|}{\text{actual}} \leq \text{error margin} \end{cases} \tag{2}$$

$$\text{Validation Error} = \frac{\text{Misses}}{|\text{data}|} \tag{3}$$

Thus, if the program returns ".8 accuracy, .5 error margin," this can be interpreted as "80% of the predictions are accurate within a 50% confidence margin." Many of the included algorithms are accompanied by a graphing function, demonstrating the accuracy of iterations, and for k-fold cross validation, the accuracy over different values of k.

3.2 Training & Testing

For the best analytical use of error for each algorithm, standardized training and validation sets were formed by script. Giving the `makeData.py` script the complete data file as input, along with a validation set size (default 300), two new subsets of the original data file were created, named *train.csv* and *test.csv*. These were used for testing on all algorithms that required a training and validation set.

3.3 Linear Regression Attempts

3.3.1 Linear Regression

The first attempt was to fit a simple linear regression to the model. Using a confidence interval of 50% (.5), the accuracy reached 62% (.62). This indicated that the data likely could not be modeled linearly. To check this conclusion, a Naïve Bayesian Ridge regression was attempted.

3.3.2 Bayesian Regression

(Sanity Check for Linear Regression)

The Bayesian Regression operates much as a standard linear regression, with the exception of the optimization problem to maximize log-likelihood, and a prior assumption regarding the distribution of the data. Because all outputs were normalized as **the percentage of teen births for all women between 15-19 (by country and year)**, it was possible to observe and use a prior probability distribution in Gaussian form (like linear regression). The validation accuracy achieved was identical to that of simple linear regression, and thus sanity was confirmed, at least in this department

3.3.3 Cross Validation on Bayesian Regression

Because the data was so very limited (only a maximum of about 1450 datapoints before splitting into training and validation sets), cross validation seemed like a viable option for checking the linear model. Using the Bayesian Regression implemented in section 3.3.2, k-fold cross validation was added to the program's capabilities.

FIGURE 3.3.2 is a graphical display of *all average cross-validation accuracies* across multiple values of *k*. As *k* increased, the *accuracy* increased, indicating that smaller validation sets did better overall.

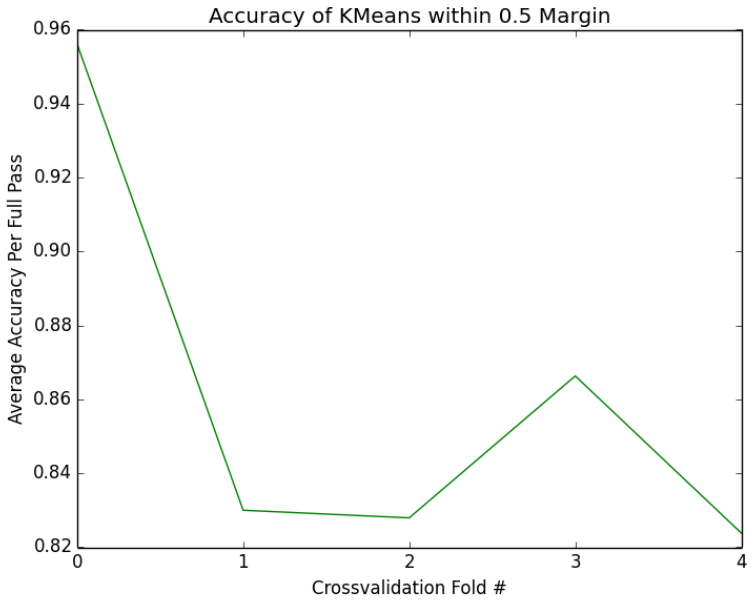


Figure 3.3.1: Graphical display of validation accuracy on crossvalidation sets. In this case, k=5, and average validation error = .86

3.3.4 Stochastic Subgradient Descent

The original thought was that stochastic gradient descent would be a good model for predicting outputs, however, the same problems became apparent on gradient descent as all regression algorithms. Given the relatively small feature set, it was unable to adequately fit the data and proved to be an extremely poor predictor (around 32% accuracy with a 50% confidence interval) of future forecasts.

3.3.5 Linear Woes

It seemed, then, that **none of the linear methods worked terribly well for determining whether or not the features were predictive of the outcome.** Upon further thought, however, this fact, although unfortunate, is not surprising in the least.

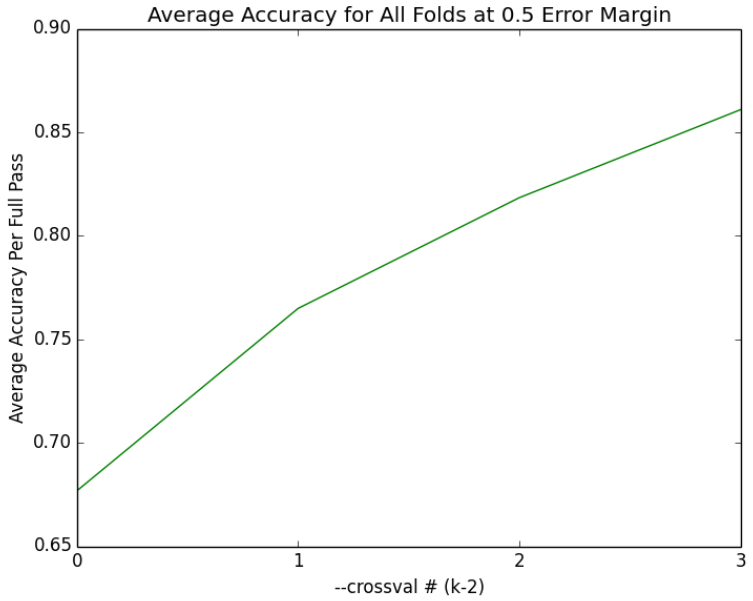


Figure 3.3.2: Graphical display of Cross Validation Accuracy as the number of k-folds in the cross validation increased from 2 to 5.

As witnessed in the cross validation of the Bayesian Regression (See *Cross Validation on Bayesian Regression, 3.3.3*), the accuracy increased as the datasets decreased in size. This indicates rather nicely that the linear regression simply couldn't fit a large data space since the VC-dimension of the linear models is simply *the number of features + 1*. Thus, a non-linear model would be necessary to determine whether or not **features were predictive of desired outcome.**

The results of this were clear when considering a simple **base case**: if the averages of the teen birth rate were taken from previous years and used to project forward, it would predict all outputs within a 50% confidence margin. Thus no linear projection method produces adequate results.

3.4 Random Forests

Due to the VC-dimensions woes of linear regression models previously detailed, the next attempt was a swing in the opposite direction: a model with varying VC-dimension, based upon depth.

> A Bit About Random Forests (RF)
> The RF algorithm is an ensemble method of regression or classification that utilizes outputs from randomized decision trees to generate a "group output." In many cases, this group decision is more accurate than the output of any one of the trees individually. Because they are randomized slightly, each tree will split on features at different thresholds, and in different orders, rendering different outputs. These outputs are then combined and returned as the group output of the RF regressor.

To prevent a RF regressor from overfitting, however, it is necessary to specify a maximum depth. FIGURE 3.4.1 is a plot of RF regressor's returned accuracy as the maximum depth (x-axis) increased from 1 to 50. The best accuracy achieved was .92 at a max depth of 41 on the nearest-neighbor modified dataset. It is simple to see that below a max-depth of 10, the RF regression drastically underfit the data. At a maximum depth of 6 or lower, RF was generally no better than the linear regression attempts. As the depth increased, however, so too did the VC-dimension of the model, and the accuracy increased to match.

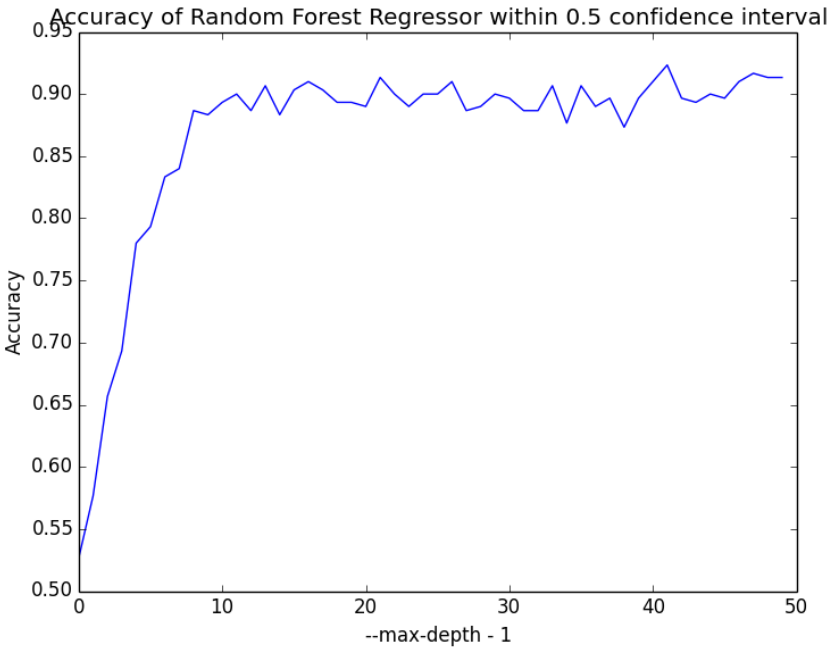


Figure 3.4.1: Plot of Random Forest accuracy over different limitations on --max-depth of each tree.

IV. Experimental Evaluation

Analysis of accuracies across several different datasets.

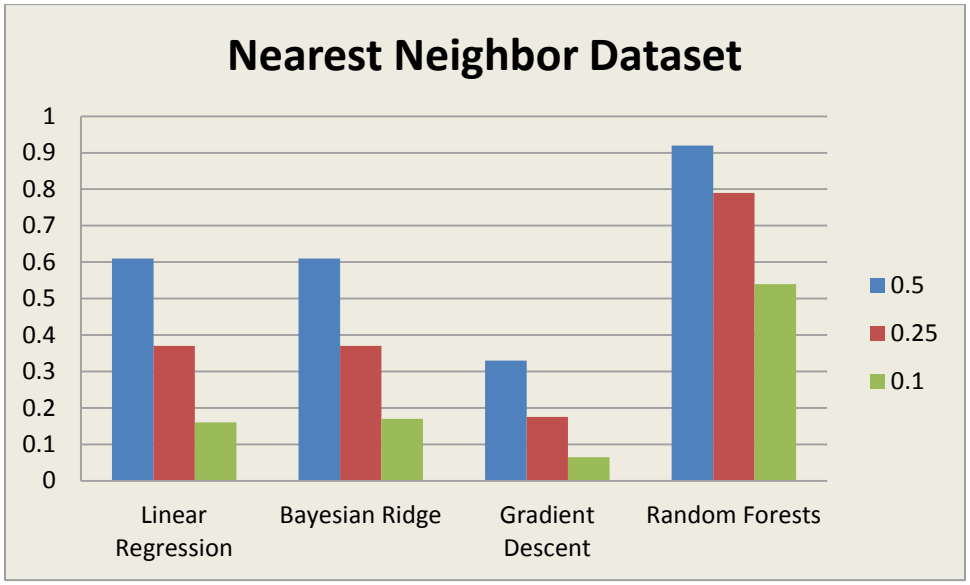
4.1 The Datasets and Their Results

nearestNeighbor_train.csv *normal_train.csv* *countries_train.csv*
nearestNeighbor_test.csv *normal_test.csv* *countries_test.csv*

4.1.1 Nearest Neighbor Dataset

As discussed in section 2.3, the first of the three datasets utilized a KNN algorithm to fill in gaps in the data. Analysis of the algorithms attempted upon the nearest neighbor dataset follows.

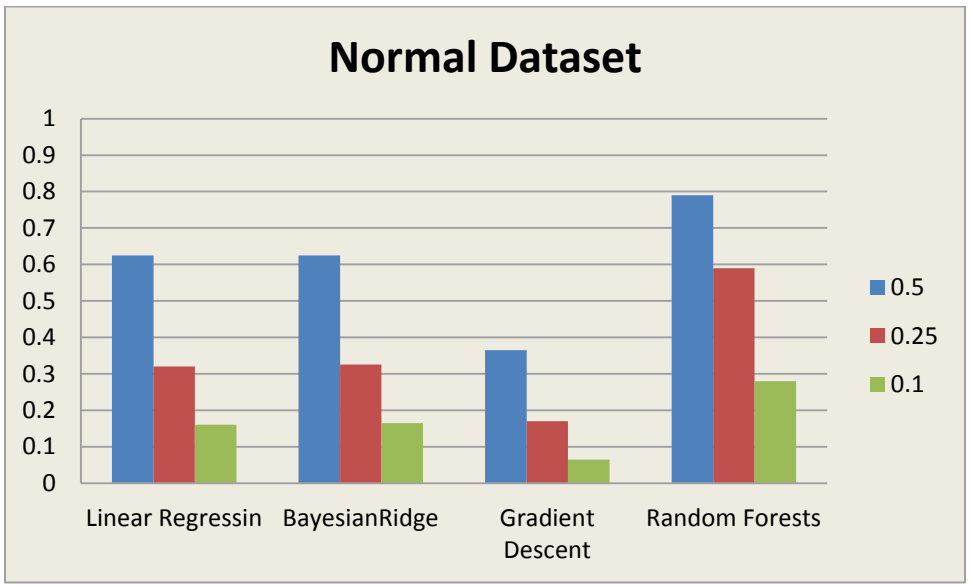
This dataset provides more training data for fitting the model, but has been manipulated before model fitting. This dataset may have decreased the accuracy of the linear models. Another reason for the lower linear accuracy is likely the VC-dimension (See *Linear Woes, 3.3.5*): as the size of the dataset increases, it becomes increasingly difficult to effectively fit a linear model, given it's small VC-dimension (namely, *features + 1*).



The larger amount of data, however, increased the prediction of the Random Forest method, indicating that the prediction power of the training set was not significantly reduced by the KNN method.

4.1.2 Normal Dataset

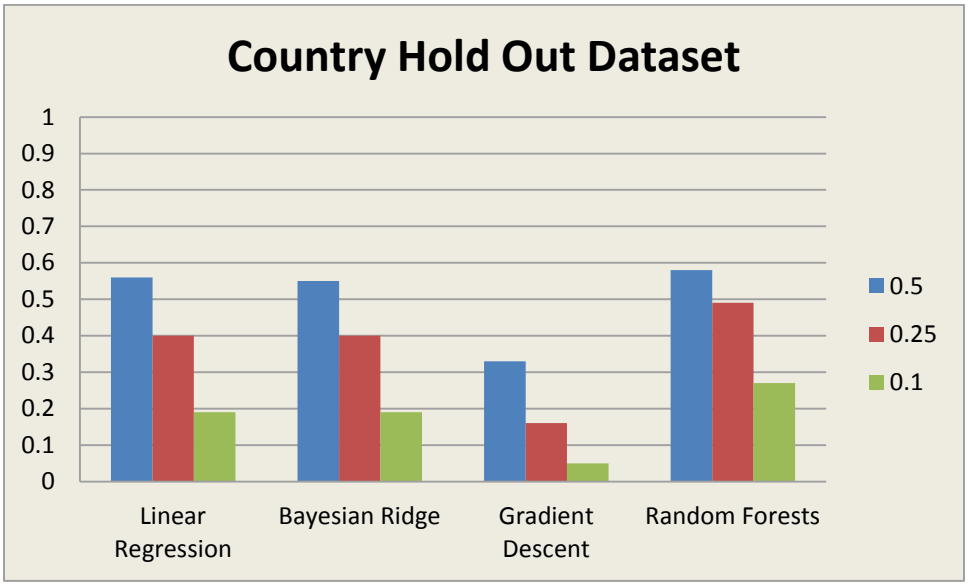
It was also discussed that performing the KNN data manipulation might decrease prediction accuracy. As such, the dataset without KNN manipulation was also attempted. The results follow.



The linear models receive a better prediction accuracy using this smaller, un-KNN-modified dataset, likely due to the fewer number of datapoints present (See *Linear Woes, 3.3.5*). The Random Forest, however, obtained a lower accuracy, due to the decrease in training data.

4.1.3 Hold-Out Countries

A good way to test the key assumption was to train on certain countries, and use others as test data. As such, building off of the Normal Dataset (4.1.2) the Country Hold-Out dataset was tested on all algorithms, the results follow.



Interestingly, the Held-Out countries set performed rather badly across all algorithms, indicating that **Assumption 1** did *not* hold true. It seems training a model to all UN countries is an inadequate method for prediction in each. This is discussed further in the Conclusion (Section 5).

4.2 Discussion

The results of the linear models above are fairly definitive in pointing out the shortcomings of the chosen methodologies. The Random Forest regressor, however, is somewhat misleading: .92 accuracy seems all well and good, but in actuality, given the confidence interval, is not particularly useful for *prediction*. Thus, the goals of the project were only partially met.

Prediction, although initially included in the project goals (See *Introduction, I*), was never a realistic expectation, considering the amount of data, and the necessary assumptions. The best prediction accuracy of all of the algorithms displayed above (See *The Datasets and Their Results, 4.1*) falls inside a 50% confidence margin, indicating that if prediction was within 50% of the actual value, it was classified as a correct prediction.

Although this makes for relatively miserable prediction potential, 92% accuracy at 50% confidence interval *does* indicate that trends can indeed be modeled, in other words **there *does*, in fact, exist a correlation between the chosen features and teenage birth rates**. To accurately predict these, however, **Assumption 1** must be removed, and a model must be trained to each country. Given the amount of available data, however, this is a frank impossibility (See *Data Constrained, 2.5*).

V. Conclusion

As it turns out, the most successful approach to this project was actually clustering. The clustering described (See *Data Visualization via Clustering and PCA, 2.4*) allowed, not for predictions, but indicated that countries with similar features were indeed grouped together. As such, it seems highly plausible that said features are indeed predictive of birthrates.

So, although the prediction goal was *not* met, it is possible to predict general trends in birthrates in UN countries, demonstrating that **there *does* exist a correlation between the specified features and birthrates.**

In future continuation of the project, there are several points of feasible alteration. Firstly, the assumption concerning training a single model across all UN countries must be removed to accurately predict within a reasonable confidence margin (perhaps, better than 20%). This would require quite a few more years of data collection in order to provide sufficient data for each country. Data was only provided for the past 50 years and worse very few countries benefited from continuous statistics filled out.

Secondly, use of only non-linear methods is undoubtedly advisable for prediction, given the previous discussion of VC dimension.

Thirdly, given greater data collection efforts by the UN itself, it could be possible to construct a higher-dimension feature vector for each country. This would allow for more powerful prediction.

VI. Bibliography

SCHAEFER MUÑOZ, SARA. "U.N. Cites Teen Pregnancy's Harm to Developing Nations." *Wall Street Journal*. N.p., 30 Oct. 2013. Web. 15 Dec. 2013.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

United Nations Statistics Division. data.UN.gov