Data Science 311 Final Project

The Olympics: A Game of Numbers

Group 3

Mitchell Dohrman, Ben Fry-Holman, Finlay MacLeod, Lucas Ragsdale

---

1. **Project Overview**

We were interested in analyzing data from sports, global health, or economics, and figured that the intersection of these things is the Olympics. As a competition of nations, the Olympics gives us a year and country index to match country-based metrics to, giving us a framework to run machine learning on to predict medals won from those country-based metrics. Our goal was to find which metric are the best predictors, and, using those metrics, see if we can create a good predictor for Olympic success.

Supporting code for the procedures described in this blog can be found in the notebooks included in the .zip file. Specific citations to individual notebooks can be found throughout the text.

---

2. **Datasets**

The core of our project is a set from Kaggle, a large, free repository of datasets found online, (https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results) tracking all Olympic winners from 1896 to 2014. This set is completely indexable by both metrics we are interested in (Country and Year) to give us a medal count for that index. So, it includes fields for Country, Year, and Medals Won. It also includes a lot of superfluous data, including a hierarchical classification of what event the medals were won in (Sport, Discipline, Event), who won the medal (Name and Gender), and the host city (a function of year). Despite all of these features we only chose to use Country, Year, and Medals Won for our analysis. Figure 1 is an example row from this dataset with named columns.

Year	City	Sport	Discipline	Athlete	Country	Gender	Event	Medal
1896	Athens	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100M Freestyle	Gold

Figure 1: Uncurated Dataframe

To compliment the Olympic set, we collected several sets from Gapminder, a global health project trying to improve the understanding of the world through data, found at https://www.gapminder.org/data/. All sets from this source have the same format just with different data and sometimes a different selection of countries. We collected and cleaned seven total Gapminder sets to pair with the Olympic set: change in GDP, children per mother, CO2 emissions per person, Gini coefficient, percent internet users, life expectancy, and population.

- Change in GDP is a float value percentage change in GDP.
- children per mother is the average number of children each woman gives birth to.
- CO2 emissions is measured in metric tons per person.
- Gini coefficient is a measure of wealth inequality within a country, higher numbers representing higher inequality.
- percent internet users is the population who has access to the internet divided by the total population;
- life expectancy is measure in the average lifespan of people within that country.
- population is the count of individuals living in that country.

Figure 2 is an example row from population featuring Afghanistan from 1800 to 1807.

country	1800	1801	1802	1803	1804	1805	1806	1807
Afghanistan	3.28M	3.28M	3.28M	3.28M	3.28M	3.28M	3.28M	3.28M

Figure 2: Gapminder Dataframe

Both sources are completely free to use and download from. The Kaggle set will require a Google account. While they are easily accessible, the sets found on Gapminder tend to have a lot of missing values and an inconsistent selection of countries from set to set. Gapminder also exhibits varying ranges in their years: some go from 1800 to 2100, some only go from 1960 to 2010. Because of these quirks, Gapminder sets should be used with caution; however, with forethought and data processing, we were able to acquire sufficient usable sets.

3. **Data Curation**

With our sets selected, the next step was to combine them into a single data frame where a single row would include a country, a year, Olympic medals won, and all of the country metrics we found from Gapminder for that country in that year; something of the form shown in Figure 3.

Country	Year	Medals_Won	GDP	Children	Emissions	Gini	Internet_Users	Life_Expectancy	Population
AFG	1896	0	1.05	7	0	38.35	0	32.625	4610000

Figure 3: Fully Sorted and Combined Dataframe, Ideal

*Code for the following segment can be found in **Olympic_Cleaning.ipynb***

To get here, we first had to transform the Olympic set so there was a core data frame to melt the other features onto. To get to this state, we dropped City, Sport, Discipline, Athlete, Gender, and Event from the original Olympic set. Then, we aggregated the medals to get a total count for each country in each year. For this step, we assumed that every medal counts as a single medal (no bonus for Gold vs Silver) and then transformed every medal into a single integer and summed all the medals a country won in each year to get a frame which contained only three columns: Country, Year, Medals Won. This step required an intermediate step to make sure we did not overcount team events as a team win should be reported as a single medal for their country, even though each participating team mate is awarded an individual medal. Finally, we combined the winter Olympics with the previous Summer Olympics to reduce the variance between years; winter Olympics have only been held for a short time, and they feature far fewer medals to be won. More on this in the exploratory analysis.

*Code for the following segment can be found in **Gapminder_Cleaning.ipynb***

With the Olympic set in a useable shape, it was time to clean the Gapminder sets and melt them into their own column with the Olympic set. While they had some discrepancies, all sets from this source have the same format just with different data and sometimes a different selection of countries. Because of this, we were able to write an algorithm to clean a single set which can clean any other set from Gapminder. This algorithm cast all the data points from string to float, set all country names to country codes to match the Olympic set, interpolated missing values with SciKitLearn's SimpleImputer(strategy='mean'), and finally listed all countries which were mismatched between the sets. Some of these mismatches could be resolved; for instance, in the Olympic set, Russia is called three different names: RU1 (Russian Empire), USR (Soviet Union), RUS (Russian Federation) while the Gapminder sets only refers to Russia over the same timeframe. To remedy this, we simply renamed all antiquated versions of Russia to RUS after confirming this simple renaming would not lose or double-count any medals. This same process worked for Germany and some multi-country teams, but a handful of countries could not be resolved as they were not included in the Gapminder sets. Because of this, they are unusable for

machine learning and so were dropped from the set. Before we made this drop, we ensured that losing these countries would not destroy too many valid medals, and found that only about 40 out of the thousands of medals in our set.

*Code for the following segment can be found in **Melt.ipynb***

Now, we have an Olympic set in the format we want, and seven Gapminder sets in a form compatible with the .melt() function in Pandas. This function takes a data frame and transforms the index, columns, and data into columns, as shown in Figure 4.

Country	1896	1900
AFG	1.91	4.3
GRE	-2.7	-0.9

Country	Year	Value
AFG	1896	1.91
AFG	1900	4.3
GRE	1896	-2.7
GRE	1900	-0.9

Figure 4: The Melt Function

In this form, each of the Gapminder sets now share an index (technically an index pair) in the form of [Country, Year] that the value can be matched to in the Olympic set. With this done, there is one more simplifying assumption that must be made before the final set is created: Whether a country participates and wins nothing, or if a country does not participate, their Medals Won for that year will be zero. This assumption ensures that all of our data have the right dimensions, and once it is applied, we can combine the Olympic set and all of the Gapminder sets to give our final data frame: a collection of countries and country metrics with a medal count for every year the Olympics have been held (modern Olympics; 1896 to 2014) with 3349 rows featuring 124 countries.

Figure 5 is two sample rows from our dataframe.

Country	Year	MedalsWon	GDP	Children	Emissions	Gini	Internet_Users	Life_Expectancy	Population
AFG	1896	0	1.05	7	0	38.35	0	32.625	4610000
BRA	2012	17	2.02	1.77	2.3425	52.8	46.475	74.45	201250000

Figure 5: Fully Sorted Dataframe, Actual

4. **Exploratory**

Our exploratory analysis had three major components: First, we wanted to better understand the Olympics and how they have changed; Second, we needed to understand the history which corresponds to the major dates we are looking; Third, and finally, we needed to learn about our country metrics we were going to examine.

Code for the following segments can be found in ***Exploratory_Analysis.ipynb***

Medals Over Time

The modern Olympics began in earnest in 1896 based on the ancient Greek tradition of a yearly athletic competition. Since this revival, athlete from around the world have competed in sporting competitions to earn the honor of winning a medal in that sport for their home nation. Since 1896, the games have changed a lot, mostly by the addition of more events. So, to aid our analysis, we began by visualizing how many medals were available in each year. The result was not surprising, but still important to understand: Every year, more events were added to the games, and so more medals were available to be won. Figure 6 demonstrates this trend.

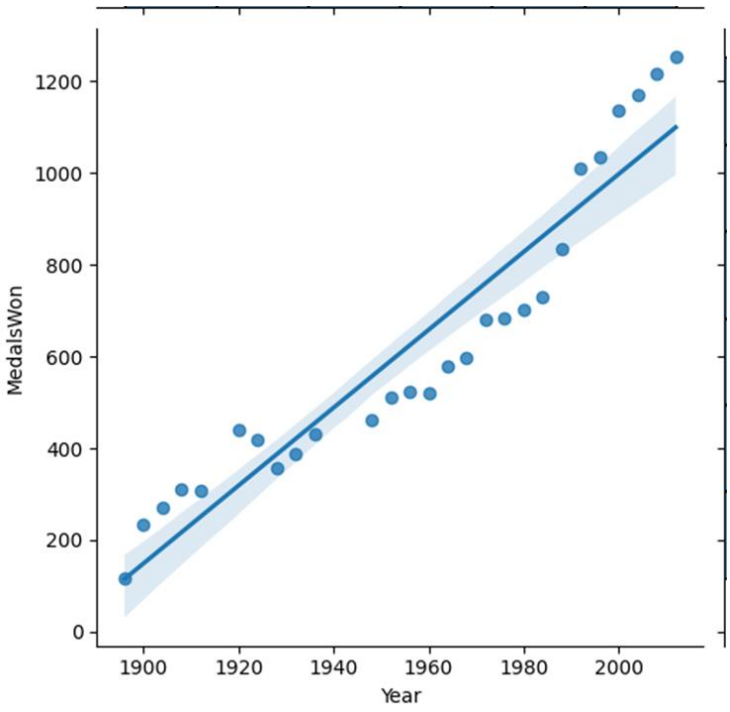

Figure 6: Total Medals Won by all Countries as a Function of Year

Each point is an Olympics games, where the X axis is the year in which they were held, and the Y axis is the total number of medals available to be won, with a best fit line and shaded variance to show the trend. There are a few notable regions in this graph. First, there are three games missing in 1916, 1940, and 1944, which were all cancelled on account of the world wars taking place in Europe. Next, there is a sharp spike in events from combing winter and summer games and 1984. With more medals available, there could be a broader distribution of countries winning medals as time goes one, which we were curious to see in our machine learning component.

Medals by Country

Next, we were interested in exploring who the major players in the Olympics have been since 1896. We had some intuition, but to solidify our understanding, we produced a heatmap for our participating countries (Figure 7.1.a).

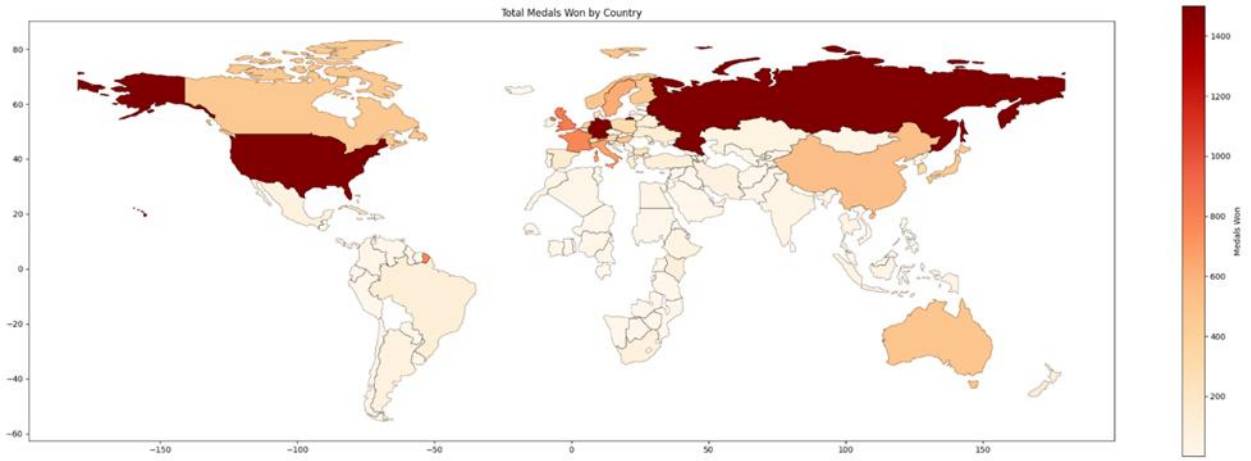

Figure 7.1.a: Geographic Heat Map of Total Medals

This chart plots all of the countries in our dataset (hence the notable omissions) and assigns them a color based on the total number of medals they have ever won at the Olympics. A darker hue of red indicates more medals, where a light orange to white color indicates fewer. At a glance, you could deduce that Russia, Germany, and the USA were the biggest players, with China, Japan, Australia, much of Europe, and Canada being the runners up. This is an accurate analysis for our time frame; however, our data only goes up to 2014, which means many medals from three summer games cycles are missing. If our data was valid to present, China would be a much bigger player than they appear to be in this project.

Medals by Country and Time

Accompanying this graphic, we produced a heatmap which sacrificed the geographic data for time data, to show our biggest winners from 1896 to 2014 (Figure 7.1.b).

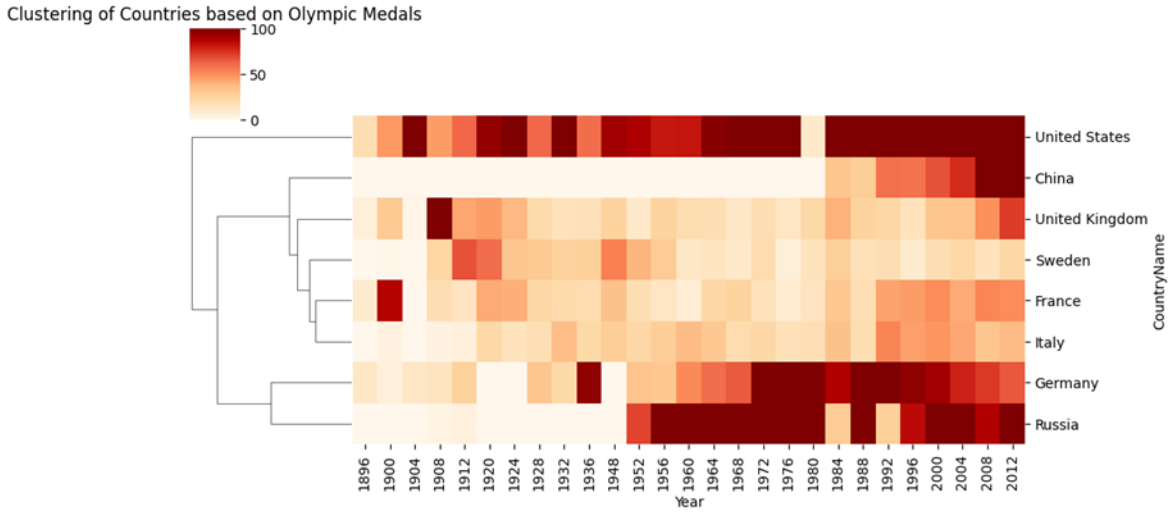

Figure 7.1.b: Time-Based Heat Map of Total Metals

Once again, a darker color of red indicates more medals won for that year. This shows the top eight winners in each year that the Olympics have been held (hence, no breaks for worlds wars). What that leaves us with is an interesting look at how the games have changes: Early on, the top performer in a year seems to jump around. For instance, in 1900 France was dominant; in 1904, the USA was dominant; in 1908, the UK was dominant. A simple google search revealed to us that these were the host countries in those years. The 1904 Olympics are an excellent example of this: the games were hosted in St. Louis, Missouri, and took place over five months (more recent games usually take at most three weeks). The inconvenience of travel and the long time-frame meant that 80% of the participants were American athletes, who took 82.5% of all of the medals. This story played out many times over in different places in the early history of the modern Olympics, and continues to be a trend, although the host advantage has become less significant as travel has become easier. Looking forward in time, that same time-based heat map shows another interesting set of events from 1980 to 1992 where top performing countries suddenly win significantly less. These are Olympic boycotts, which were a favorite tool of superpowers to express displeasure during the Cold War era. Similarly, China is an example of a country who wanted to participate but was barred from the games for political reasons; no one could agree on which China should be called China as a result of a civil war: was it the Republic of China in Taiwan, or was it the Peoples Republic of China on the mainland? In 1984, both Chinas finally

Metrics by Country

With a solid base in Olympic drama over the hundred-year history we are examining, it was time to understand the country metrics we curated from Gapminder. Using a similar geographic heatmap to before, we could visualize these metrics for the entire world at once, starting with population:

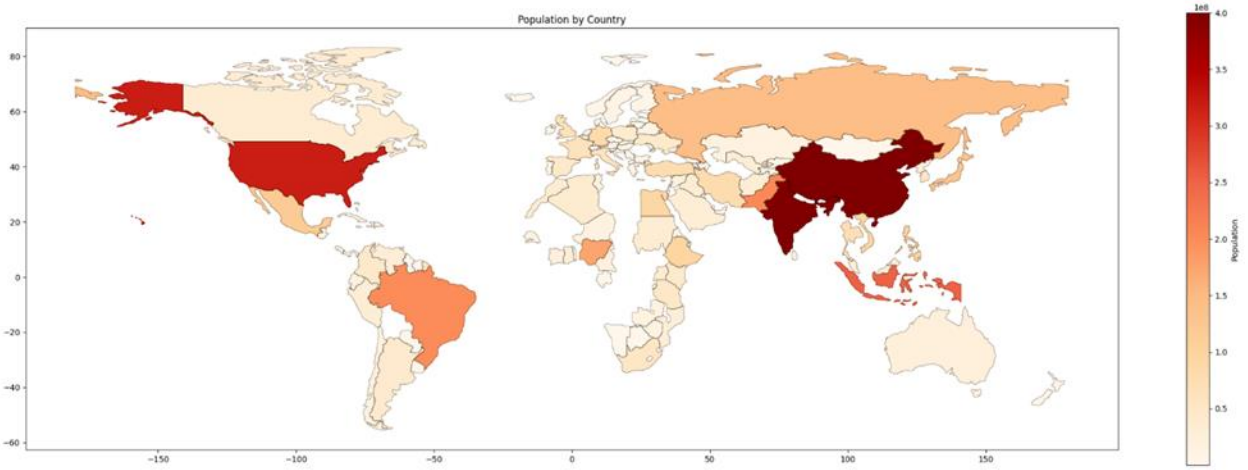

Figure 7.2: Geographic Heat Map of Population

In Figure 7.2, warmer color, again, indicates a higher value: in this case, population. At a glance, it is difficult to see any correlation between this map and the Medals Won map; some countries with large populations (India, Indonesia, Nigeria) are not major players in the Olympics while China and the USA do have both high populations and high medal counts. While it is difficult to discern much from this map about the Olympics, it is still an excellent way to understand the world we are working with and how the country metric is distributed around the globe. As such, we produced similar graphic for our other country metrics as well:

CO2 Emissions –

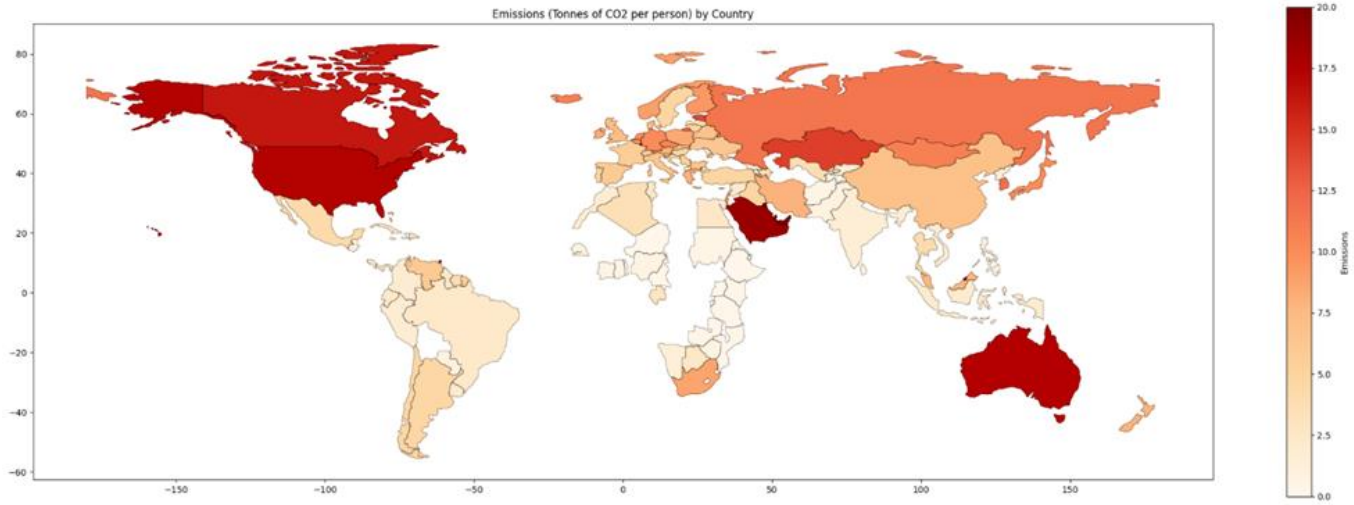

Figure 7.3

Birth Rate –

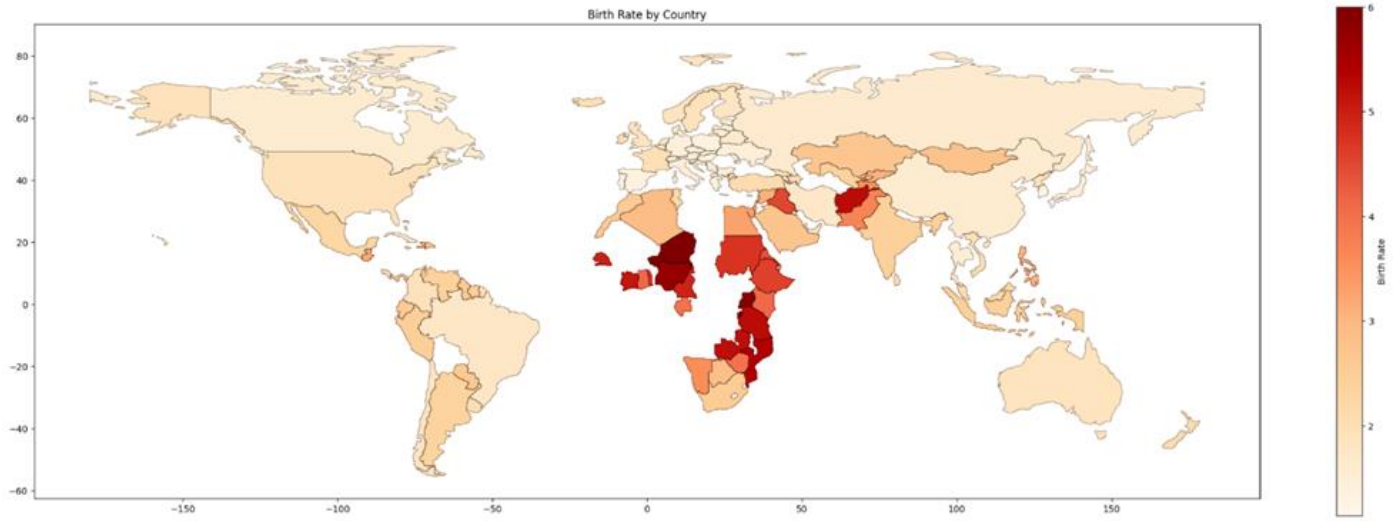

Figure 7.4

Gini Coefficient –

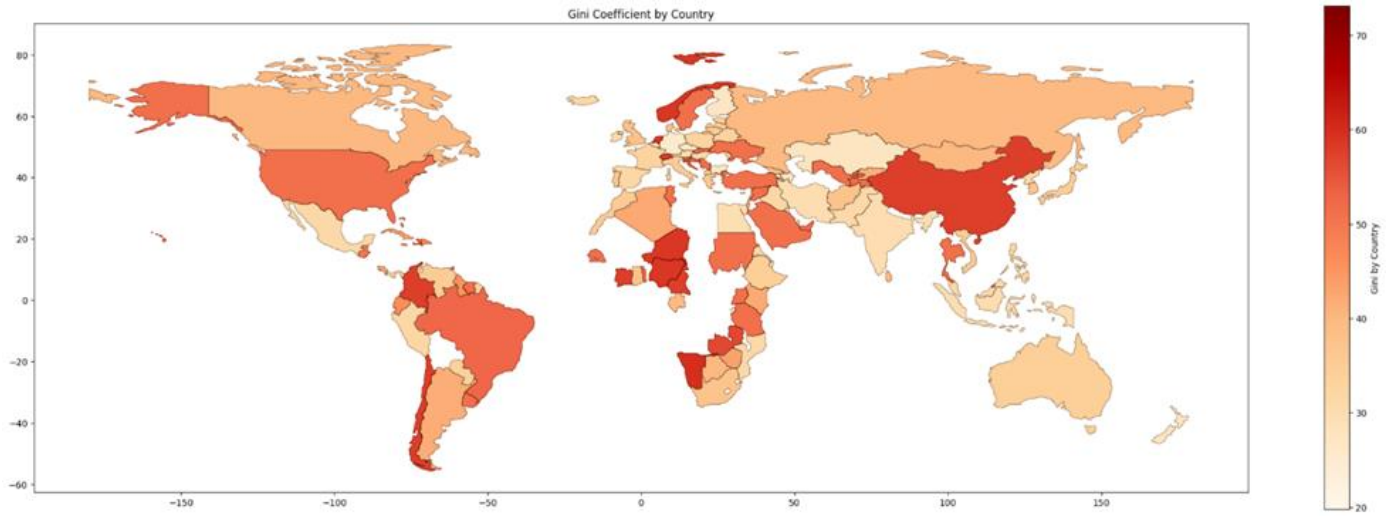

Figure 7.5

Life Expectancy –

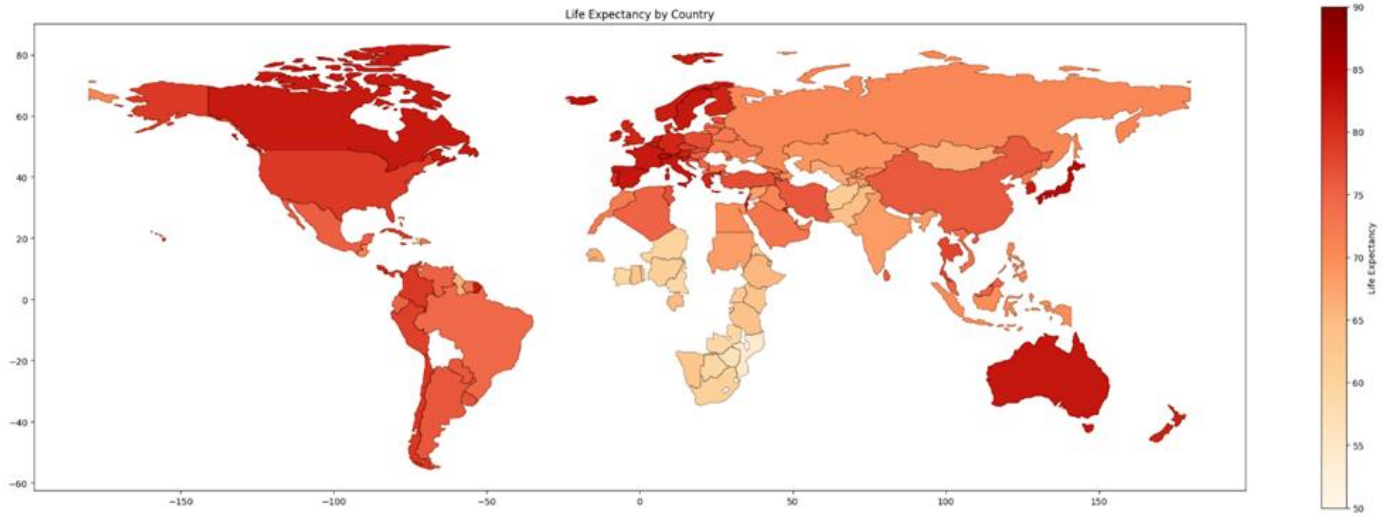

Figure 7.6

Change in GDP (Bluer means more negative, redder means more positive)–

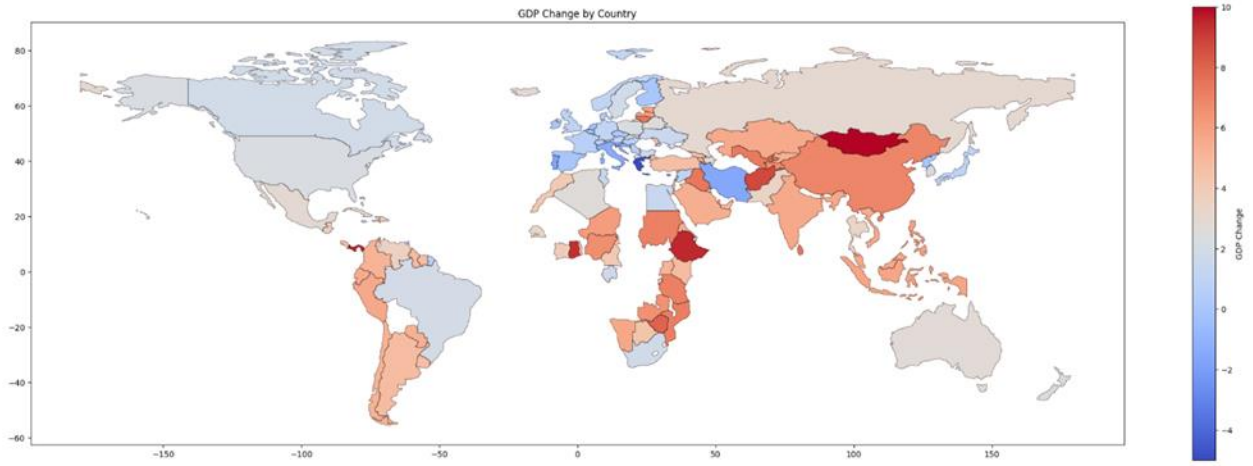

Figure 7.7

Percent Internet Users –

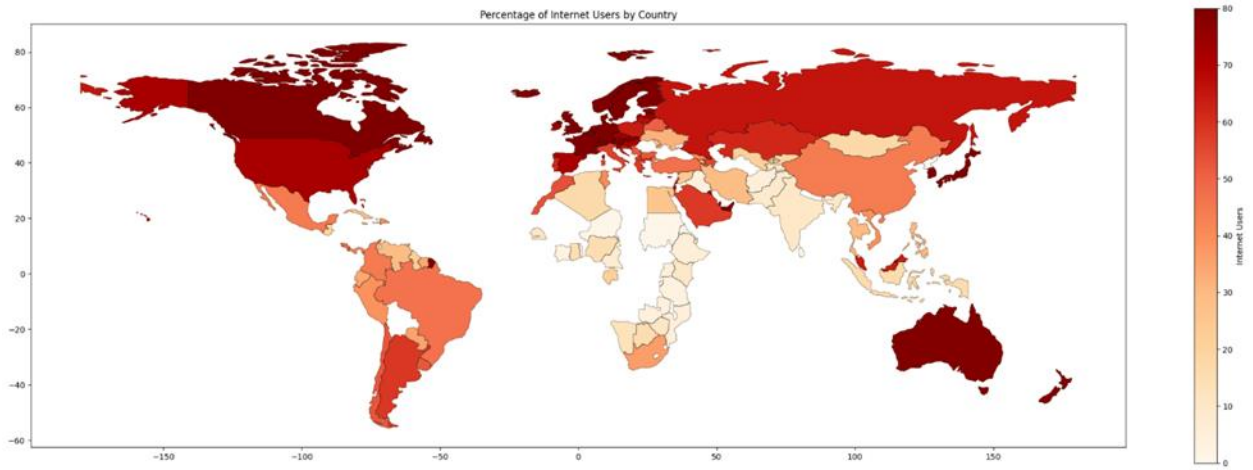

Figure 7.8

These charts are excellent for understand the world, but in order to relate them to our Olympic set, we needed to relate them to medals in a less pretty format. To do this, we could select a metric and a collection of countries and create a scatter plot of their medals won against the chosen metric at that time. For instance, change in GDP is pictured in figure 8.

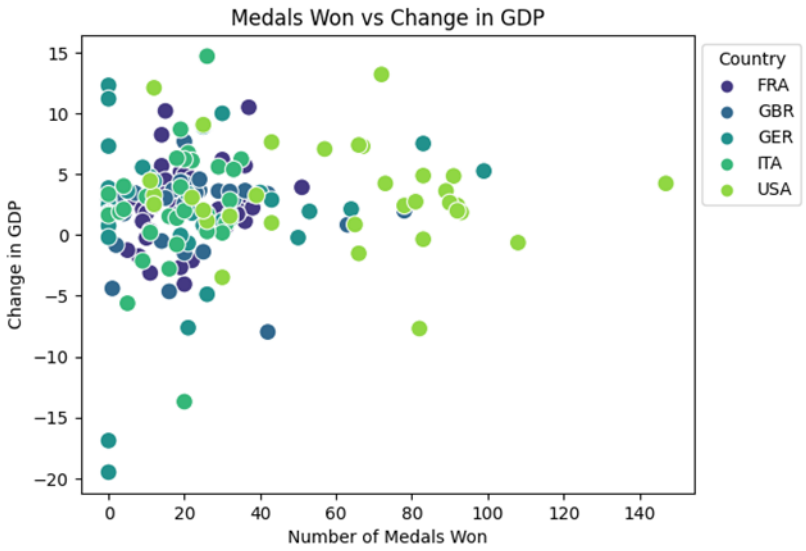

Figure 8: Change in GDP Against Number of Medals Won

From this graph, we can see that most of the medals ever won correspond to a positive change in GDP. It was encouraging to know that there is some correlation, but to explore this to its fullest extent, we needed to quantify the connection between our metrics and Olympic medals with machine learning.

5. **Machine Learning**

Code for the following segments can be found in ***Machine_Learning.ipynb***

To find and quantify the correlation between our country metrics and Olympics success, we attempted several different linear regression techniques from the SciKitLearn library. We felt his approach matched our data and goals best as we could attempt to predict medals won as the output Y of some model with an input vector X made up of our country metrics and the year, which we could easily create by indexing the columns of our dataset. To measure the success of our models, we used the R^2 value to measure the success of the model and the RMSE (Root Mean Squared Error) to measure the loss of each model. Our priority when selecting the best model was maximizing the R^2.

Before getting into our process for machine learning, we should introduce this graph in Figure 9.

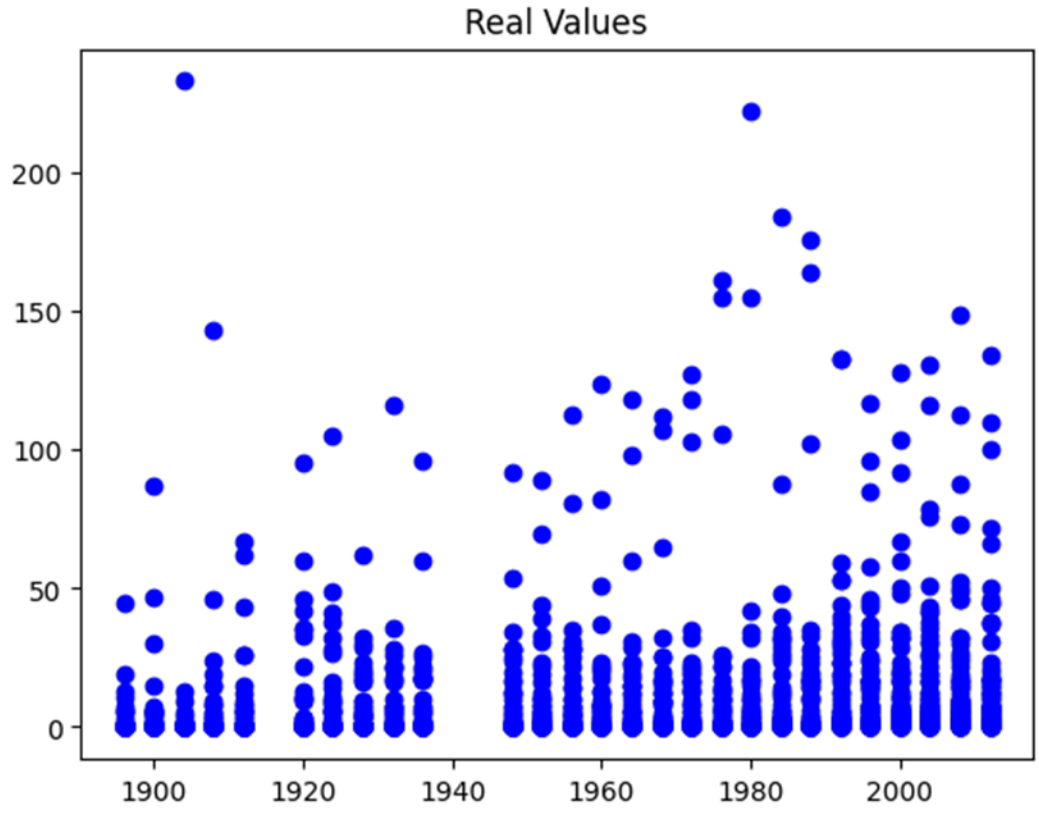

Figure 9: Real Value of Medals Won in Each Year, for Each Country

This is the actual medals won by each country in each year. Each blue dot represents a country's medal count (Y) for that year (X). You may notice that there is no delineator for which dot

belongs to which country, and this is acceptable for this segment because country was not included as a predictor. This chart will be referenced later to understand the performance of our models.

The Splits

Our methods were forms of supervised machine learning; as such, we needed to establish data splits. We wanted a test set to fit the model, a validation set to measure the model, and an ultimate test set which would verify the model's success. We settled on a 70-15-15 split, and seeded the train_test_split() function with random_state=42 and made the splits which were the same for all models. The Train Set and the Validation Set were used in each machine learning attempt, while the Test set was reserved to verify our top performing models after all machine learning was completed (this turned out to be a good choice).

The Models

We first established a baseline model which assumed there was no difference between any countries. With this approach, the available medals in each year were distributed equally among all of the countries in our dataset. So, with the input of year, this model will predict every country winning their fair share of the medals, giving Figure 10.

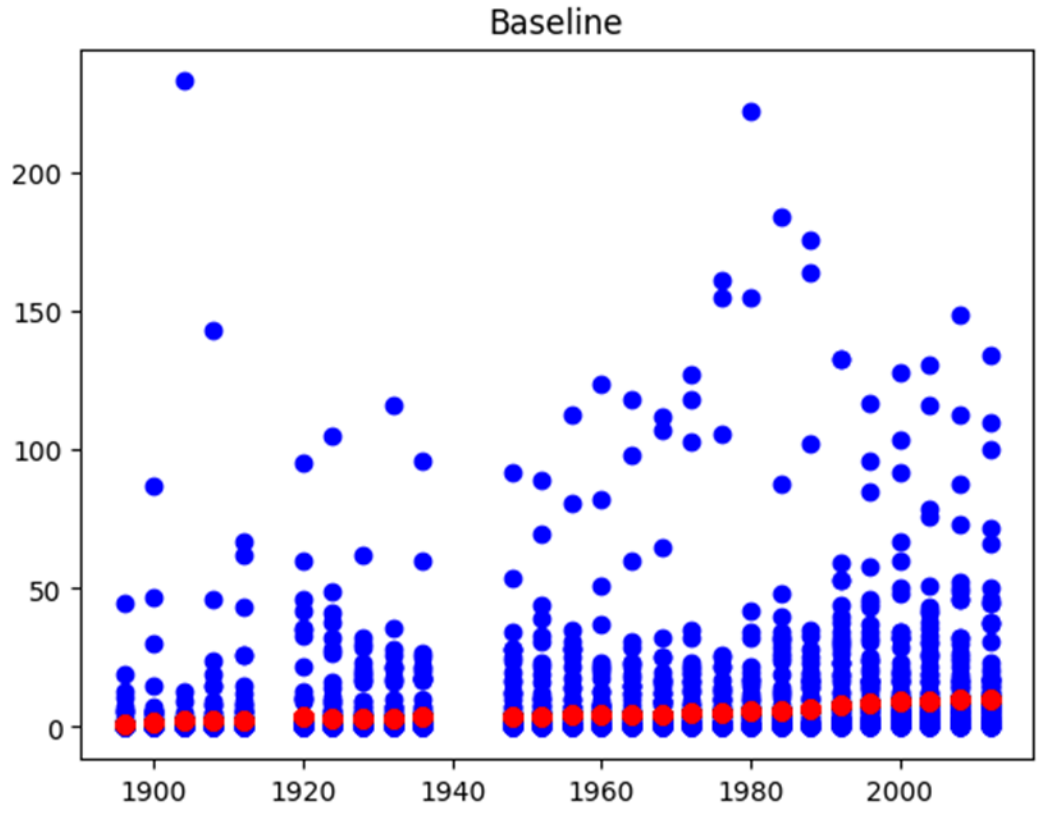

Figure 10: Baseline Model vs Real Values

The red dots represent the predicted values. While there appears to be far too few predictions, all 124 countries are predicted to have the same value, and so they are stacked directly on top of each other. This model appears to find the center of mass in the chart, but miss the higher medal counts completely. This is reflected in the R^2 value and the RMSE, which were 0.0336 and 18.7 respectively (all models' scores will be reported in a table later in the report). While these numbers are bad, it is to be expected from out least intelligent approach.

To improve upon this, we tried several linear regression models including: multiple simple Linear Regressions, a Neural Network, and a Random Forest Regressor. These models were also paired with a handful of preprocessing steps to see if we could improve the outcome.

The first improved model we attempted was a simple Linear Regression with an order 2 polynomial feature expansion. This model achieved an R^2 score of 0.219 with a RMSE of 16.8. An improvement to be sure, but we still wanted to increase the R^2 score.

Our next attempt was a feature reduction, using a recursive support machine to only select the best features and run them through a Linear Regression. The support machine chose GDP, Children, Emissions, and Internet Users to use as inputs. The result was even worse performance, achieving an R^2 score of 0.148 and a RMSE of 17.6.

Next, we tried a straight Linear Regression with no feature manipulation, which gave similar results to the polynomial expansion: an R^2 of 0.229, and a RMSE of 16.7.

Next, we combined the first two feature manipulations and standardize the medal winnings by adding 1 to every medal count. The hope was that his would learn better with fewer zeros in the Y values, but it still only yielded an R^2 of 0.219 and a RMSE of 16.8.

The best of these models is plotted in Figure 11.

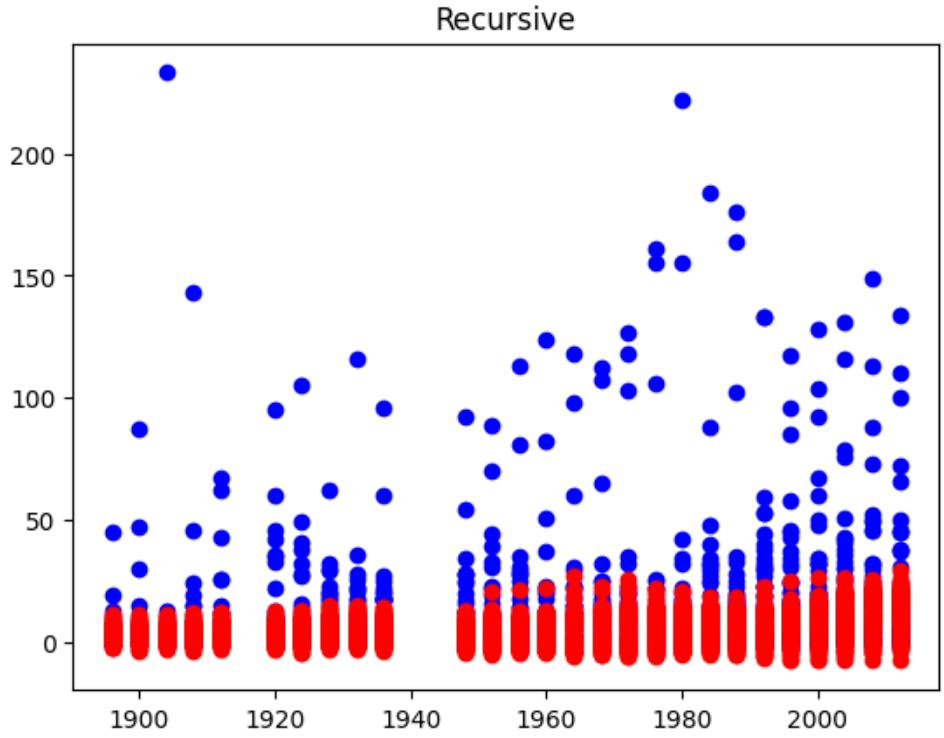

Figure 11: Recursive Feature Selection vs Real Values

This model is exceptionally bad at capturing outliers, and even appears to occasionally predict negative medals won, which is not a possible outcome of the Olympics.

Simple Linear Regressions were not working, so we moved on to a Neural Network with 3 perceptrons per 32 layers. This model yielded a significant improvement, achieving an R^2 of 0.696 and an RMSE of 10.5.

We wanted to push it further, and so we tried a Random Forest Regressor, which turned out to be our strongest model with an R^2 of 0.754 and a RMSE of 9.46. To see if we could do any better, we once again attempted some standardization, per the recommendation of our professor, and added 1 to every medal won. The Random Forest Regressor did marginally worse with this change with an R^2 of 0.750 and a RMSE of 9.53.

---

6. **Results and Analysis**

---

*The following segments continue through **Machine_Learning.ipynb***

With the Random Forest Regressor and the Neural Network performing the best, we chose to pit them against the Test Set, which no model had yet interreacted with. The Neural Network perform incredibly poorly with an R^2 of -573223199042301 and a RMSE of 460178201. Luckily, the Random Forest Regressor achieved an R^2 of 0.733 and a RMSE of 9.93 on the Test Set, cementing it as our strongest model. Looking at the Neural Network predictions on a graph, it is understandable why it performed so poorly on the Test Set (Figure 12).

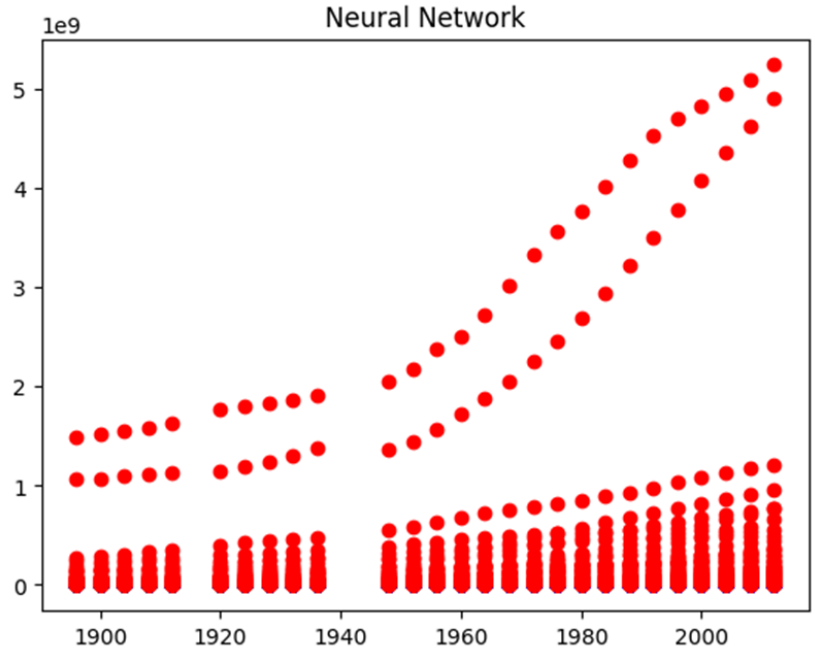

Figure 12: Neural Network Predictions vs Real Values

While the shape of Figure 12 looks decent, the fact that there are no blue dots is concerning. Upon examining the scale, it becomes clear why: this model is predicting millions of medals won, which is an absurd amount. Our assumption is that it is overfit to the Training Set, and so generalizes very poorly. The Random Forest Regressor, however, actually appears fairly consistent, as its metrics suggest (Figure 13).

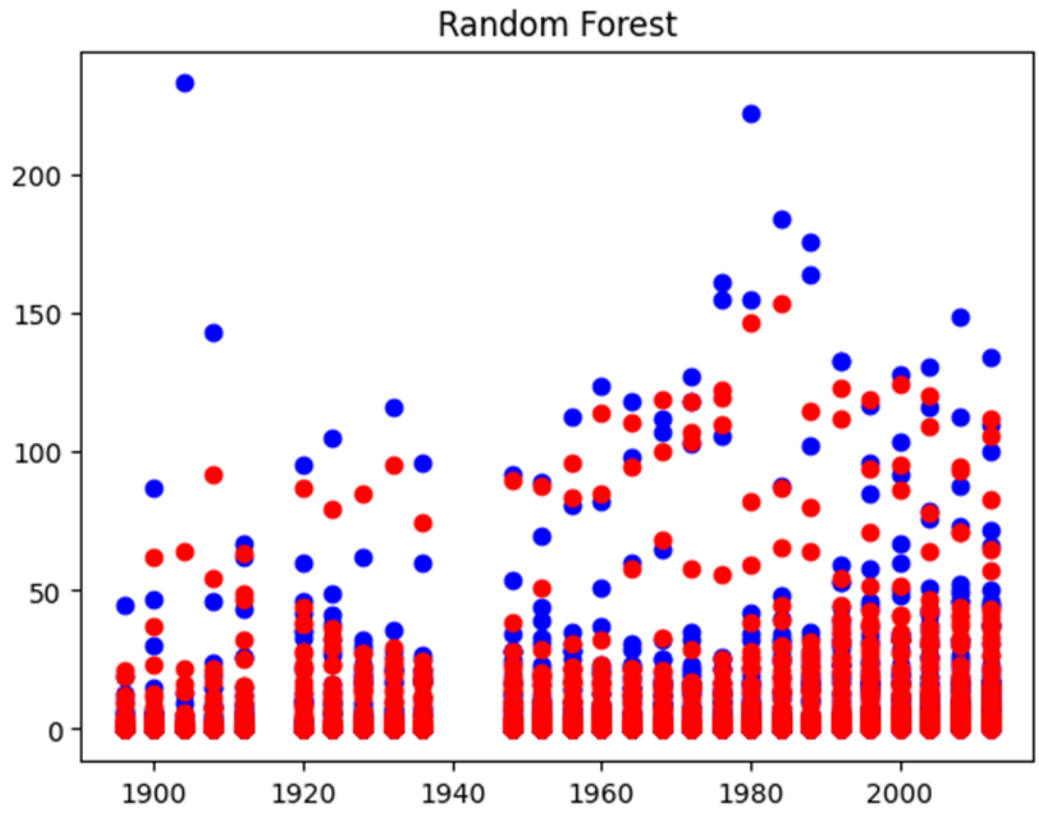

Figure 13: Random Forest Regressor Predictions vs Real Values

Its biggest failings appear to be predicting high performers. In fact, we found some of the extreme outliers in the blue true values correspond to the historical context discussed earlier. For instance, the blue dot above 200 medals in 1904 is the USA in the year that the USA hosted. We believe this sort of anomaly is unreasonable for a linear regression to model, and so are satisfied with calling the Random Forest Regressor our best model.

Before we dive into the analyzing this model, here is the full table of all our machine learning attempts is pictured in Figure 14.

Model	R^2 (Validation)	RMSE (Validation)	R^2 (Test)	RMSE (Test)
Forest	0.754	9.46	0.732	98.9
Forest +1	0.750	9.53		
NN	0.696	10.5	-5.7*10^14	460178201
Linear	0.229	16.7		
Poly	0.219	16.8		
Poly + 1	0.219	16.8		
Recursive	0.148	17.6		
Baseline	0.0336	18.7		

Figure 14: Table of Models and their Performance

Now, with our best model selected, it was time to explore the feature weights the model was using to predict medals. The Random Forest Regressor has a field called feature_weights, which is an array of float values for each input feature, which will all sum up to equal 1. A higher weight means the associated feature was a more important predictor than something with a lower weight. Figure 15 is a table of these values sorted with the most influential features on top.

Feature	Weight
Emissions	0.37
Population	0.32
Children	0.1
Gini	0.06
Life Span	0.06
dGDP	0.044
Year	0.037
Internet%	0.009

Figure 15: Feature Weights of the Random Forest Regressor

Together, Emissions and Population make up 69% of the predictive weight. Population acting as a good predictor was not surprising as it holds one of the simplest explanations: with no other factors, more people would mean more participants which means more medals. Emissions on the other hand, is more complicated to break down. We have loosely associated Emissions with economy (GDP would have been a better representation, but we could only find change in GDP for our timeframe) because Emissions are created by expensive entities: industry, infrastructure, transportation, and agriculture. So, higher Emissions means more money at the country's disposal. Because Olympic athletes are expensive to train, this is a reasonable outcome. Both of these results are corroborated by the exploratory maps presented earlier; the same countries with dark red colorings for Population or Emissions also tend to have dark red colorings for Medals Won. While this observation was previous just a guess, these numbers confirm the relationship.

The next best predictor, albeit much weaker than Emissions and Population, is Children per Woman, with a predictive weight of 10%. Looking at the maps from the exploratory analysis, this almost appears to be an anti-predictor; where a high value signals worse performance with the darkest colorations on the map corresponding to some of the weakest Olympic teams. This is also likely because the average number of children women give birth to in their lifetime in a country is inversely correlated to factors like that country's health, average education, and wealth.

The other factors fall below a threshold where it is difficult to draw conclusions from them. Percent Internet Users turned out to be the worst predictor, probably because the internet was not around for our entire timeframe. The other factors, Gini Coefficient, Life Span, Change in GDP, and Year are more influential than Percent Internet Users, but still way smaller than the top predictors. Most surprising is Life Expectancy; we anticipated this to act as a metric for the general health of a nation which may have told us something about the health of athletes, but it was barely used to predict medals. Our best guess would be that life spans have become too homogenous across the world to differentiate between country based on it.

7. **Concluding Remarks**

At the onset of this project, we intended to build a good predictor for Olympic success and find what factors acted as the best predictors. We had much more success with the latter; the weights our model used to predict medals tell us a lot about what actually goes into winning at the Olympics: money and people. As for the prediction aspect, even our best model cannot make incredibly accurate predictions. While the RMSE of the Random Forest Regressor is only around 9, on average being wrong by 9 medals is significant. In the worst cases, the difference can be much larger. Despite this, we still believe our model is meaningful for the weights we can extract from it. Its metrics show that it is good enough to recognize the trends in the inputs and their relationship to the output. With the correct purpose in mind, the model was a success, just not at predicting with extreme precision.

If we were to continue this research, or start again, there are a few questions we still wish to explore. First, we are interested in how many athletes actually participated from each country. This could open up an entirely new metric to explore: win rate. If a country sends two athletes and they both win gold, that is notably different than sending three hundred athletes and winning the same medals. This would also resolve the choice we made early in data curation to treat not participating the same as not winning. Second, on a similar note, we wondered if assigning arbitrary weights to each medal would change the results. For instance, if a gold medal were worth 3 points, a silver medal was worth 2, and a bronze medal was worth 1, and we compared the sum of points instead of medals, what would we find?

Now, while our final results are interesting, it is kind of a disappointing conclusion that more people and more money produces better results. Such an outcome almost undermines the integrity of the competition: countries essentially need to buy medals. These predictors, however, are only the best predictors from the selection we curated. Perhaps there are other, better predictors out there. We also studied this at a very coarse level of granularity. If we were to study this on an athlete level, rather than a country level, we would likely find that the Olympics are dominated by exceptional individuals, not their home country’s population. This would be a more acceptable conclusion, but our methods could not reach it.

8. **Chat GPD Acknowledgement**

To assist in our project, we used Chat GPT to generate some code segments. The "conversations" with Chat GPT were sometimes quite long; the discussion here is kept brief. In general, generated code was decent, but not quite usable as is. Outlined below is the general use of Chat GPT in the referenced notebooks with bulleted notes about the changes needed to make the generated code useable. More in depth discussion, as well as specific prompts and associated code can be found in the notebooks containing the generated code.

Chat GPT can be found at: https://chat.openai.com/

Data Cleaning - exact prompts and responses in the ***Olympic_Cleaning.ipynb***

notebook Chat GPT was asked for:

1. functions related to splitting, grouping and adding to a dictionary of dataframes as we went from our initial Kaggle athlete-wise set to our rough_sort which was an event-wise data set.
    - splitting to years, combining winter and summer dfs, adding event counts, adding year column back in
2. geographic region implementation, from dictionary creation to column additions

Machine Learning - exact prompts and responses in the ***Machine_Learning.ipynb*** notebook

Chat GPT was asked to:

1. Generate a Machine Learning model (polyexp + rfe)
    - changed from lasso to linear (because we did linear)
    - retained feature names through the expansion
    - see general structure of ML workflow, comments are helpful
2. use a more general prompt to get our low-level models (baseline, polyexp, rfe, custom)

    - enumerated and differentiable variable names were the nicest part of this step, it kept things to their own blocks which aided the clarity and collaboration potential of gpt generated code.
3. expand to our final model count: baseline, poly, poly+1, rfe, custom, neural net, random forest, random forest +1
    - best part of this was the easy to present score comparison at the bottom.
    - for the +1 variants, it used in-place changes to the df and never took them back. this would've changed the data for each model after the change, but we fixed it by hand.
    - hyperparameters were filled under some choice that it never explained
4. Implement a validation set.
    - late into the development we still had no validation / test split, so using gpt we implemented the same 70/15/15 split for every model on the same randomstate setting.