Using random forest regression to determine what leads to world happiness

Baiqing Lyu
Boston University
`baiqing@bu.edu`

Shengyao Luo
Boston University
`jaxluo@bu.edu`

Hanzi Yu
Boston University
`hanziyu@bu.edu`

Juncheng Zhang
Boston University
`morganz@bu.edu`

Abstract

By using the world happiness dataset provided by Kaggle, we wanted to see what the world needed to improve on the most for immediate happiness gain. We used a random forest regressor to determine the effects of scaling certain features and their correlation to world happiness.

1 Introduction

Advancement of modern technology has directly improved the living condition for the whole-world population dramatically. One of the metrics being taken is happiness, which creates one of the measurement to elevate the wellness of someone's life. The World Happiness Report is a landmark survey of the state of global happiness, reporting these 7 factors: real GDP per capita, social support, healthy life expectancy, freedom to make life choices, generosity, perceptions of corruption. In our project, we want to find out which factor/factors could enhance the happiness score which correlates to achieving a higher quality of life. All code, reports and supplementing information can be located at `https://github.com/BaiqingL/CS542-Final`

2 Background

Modern society and most countries on earth allows people to live in a cleaner and more comfortable situation compared to centuries ago. The World Happiness Report is a landmark survey of the state of global happiness. The Report is a publication of the United Nations Sustainable Development Solutions Network. It contains articles and rankings of national happiness, based on respondent ratings of their own lives, which the report also correlates with various (quality of) life factors. The report primarily uses data from the Gallup World Poll. Data is collected from people in over 150 countries. Each variable measured reveals a populated-weighted average score on a scale running from 0 to 10 that is tracked over time and compared against other countries. The scores are based on answers to the main life evaluation question asked in the poll. This question, known as the Cantril ladder, asks respondents to think of a ladder with the best possible life for them being a 10 and the worst possible life being a 0 and to rate their own current lives on that scale. The scores are from nationally representative samples for the years 2013-2016. These variables currently include:real GDP per capita, social support, healthy life expectancy, freedom to make life choices, generosity, perceptions of corruption, below are the brief introductions of those variables from the data-set:

2.1 Real GDP per capita

GDP per capita is a measure of a country's economic output that accounts for its number of people.

2.2 Social support

Social support means having friends and other people, including family, turning to in times of need or crisis to give you a broader focus and positive self-image. Social support enhances the quality of life and provides a buffer against adverse life events.

2.3 Healthy life expectancy

Healthy Life Expectancy is the average number of years that a newborn can expect to live in “full health” — in other words, not hampered by disabling illnesses or injuries.

2.4 Freedom to make life choices

Freedom of choice describes an individual’s opportunity and autonomy to perform an action selected from at least two available options, unconstrained by external parties.

2.5 Generosity

Generosity is defined as the residual of regressing the national average of responses to the question, “Have you donated money to a charity in past months?” on GDP capita.

2.6 Perceptions of corruption

The Corruption Perceptions Index (CPI) is an index published annually by Transparency International since 1995, which ranks countries “by their perceived levels of public sector corruption, as determined by expert assessments and opinion surveys.”

2.7 Algorithm decision

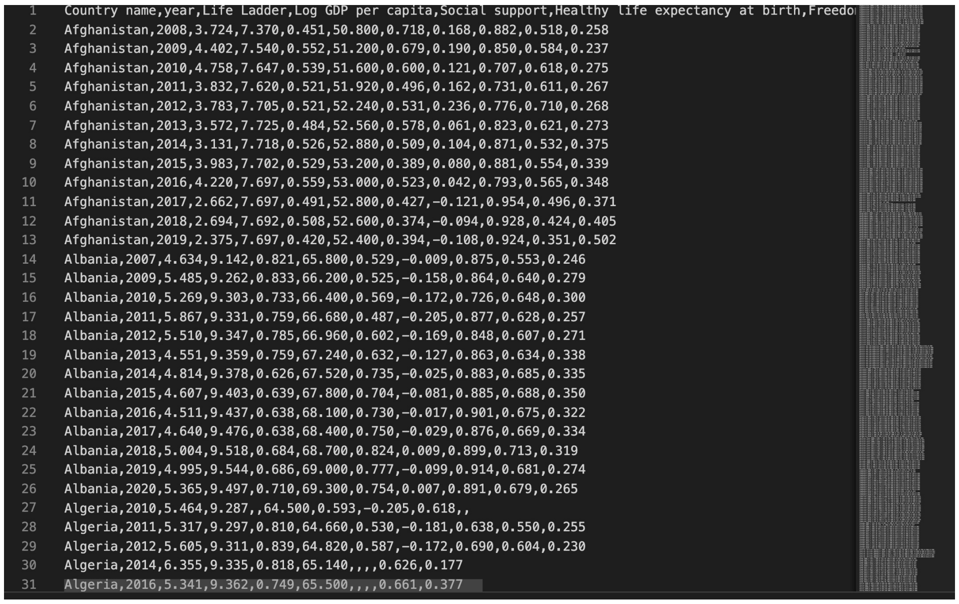
There were a lot of different algorithms we could use, but we were able to rule out some methods like linear regression or methods that assume the data-set contained a linear relationship. The reasoning for this design choice is due to the context of our data. Since in human psychology, there is a tendency for people to return back to homeostasis after reaching a certain level of happiness or satisfaction. This means for example, the same amount of money to a person with none to begin with will have different impacts if it was given to a wealthy person. This discrepancy in happiness generated by the same resources could be extrapolated onto a global scale. As countries with the highest amount of wealth or highest level of happiness in terms of individual features does not show a happiness score that is equally as high compared to countries with less, it shows a non linear relationship between happiness and the amount of resources available.

The decision to rule out any algorithms that assumes a linear relationship between data points leads us to another possible approach which is the random forest regressor. We know that by constructing a random forest, we can harvest the low bias and high variance property of individual decision trees and create an aggregation of outputs for the forest. Which in result, gives us a more comprehensive and regulated output that will more closely align with actual happiness than a reflection of a over fitted model.

3 Data processing

3.1 Data overview

A sample of the whole data set is shown at figure 1. There are currently 166 countries and regions included within. For each row, the first element records the name of the country, the second element indicates the year of the data. Our Y value is "Life Ladder", i.e. the happiness value, as the third element of each row and the rest are deterministic features such as generosity, GDP, social support etc,

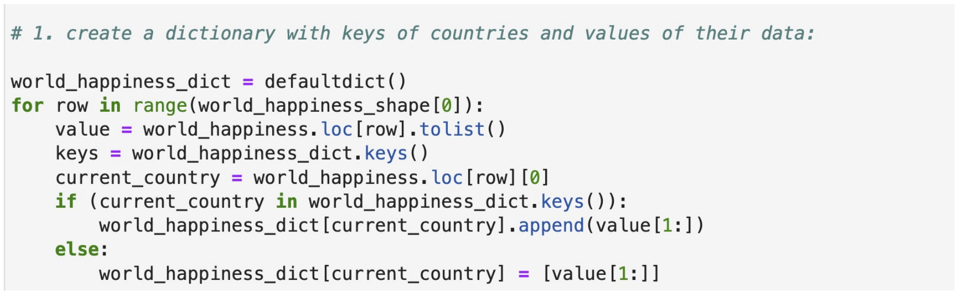


```
Country name,year,Life Ladder,Log GDP per capita,Social support,Healthy life expectancy at birth,Freedo
Afghanistan,2008,3.724,7.370,0.451,50.800,0.718,0.168,0.882,0.518,0.258
Afghanistan,2009,4.402,7.540,0.552,51.200,0.679,0.190,0.850,0.584,0.237
Afghanistan,2010,4.758,7.647,0.539,51.600,0.600,0.121,0.707,0.618,0.275
Afghanistan,2011,3.832,7.620,0.521,51.920,0.496,0.162,0.731,0.611,0.267
Afghanistan,2012,3.783,7.705,0.521,52.240,0.531,0.236,0.776,0.710,0.268
Afghanistan,2013,3.572,7.725,0.484,52.560,0.578,0.061,0.823,0.621,0.273
Afghanistan,2014,3.131,7.718,0.526,52.880,0.509,0.104,0.871,0.532,0.375
Afghanistan,2015,3.983,7.702,0.529,53.200,0.389,0.080,0.881,0.554,0.339
Afghanistan,2016,4.220,7.697,0.559,53.000,0.523,0.042,0.793,0.565,0.348
Afghanistan,2017,2.662,7.697,0.491,52.800,0.427,-0.121,0.954,0.496,0.371
Afghanistan,2018,2.694,7.692,0.508,52.600,0.374,-0.094,0.928,0.424,0.405
Afghanistan,2019,2.375,7.697,0.420,52.400,0.394,-0.108,0.924,0.351,0.502
Albania,2007,4.634,9.142,0.821,65.800,0.529,-0.009,0.875,0.553,0.246
Albania,2009,5.485,9.262,0.833,66.200,0.525,-0.158,0.864,0.640,0.279
Albania,2010,5.269,9.303,0.733,66.400,0.569,-0.172,0.726,0.648,0.300
Albania,2011,5.867,9.331,0.759,66.680,0.487,-0.205,0.877,0.628,0.257
Albania,2012,5.510,9.347,0.785,66.960,0.602,-0.169,0.848,0.607,0.271
Albania,2013,4.551,9.359,0.759,67.240,0.632,-0.127,0.863,0.634,0.338
Albania,2014,4.814,9.378,0.626,67.520,0.735,-0.025,0.883,0.685,0.335
Albania,2015,4.607,9.403,0.639,67.800,0.704,-0.081,0.885,0.688,0.350
Albania,2016,4.511,9.437,0.638,68.100,0.730,-0.017,0.901,0.675,0.322
Albania,2017,4.640,9.476,0.638,68.400,0.750,-0.029,0.876,0.669,0.334
Albania,2018,5.004,9.518,0.684,68.700,0.824,0.009,0.899,0.713,0.319
Albania,2019,4.995,9.544,0.686,69.000,0.777,-0.099,0.914,0.681,0.274
Albania,2020,5.365,9.497,0.710,69.300,0.754,0.007,0.891,0.679,0.265
Algeria,2010,5.464,9.287,,64.500,0.593,-0.205,0.618,,
Algeria,2011,5.317,9.297,0.810,64.660,0.530,-0.181,0.638,0.550,0.255
Algeria,2012,5.605,9.311,0.839,64.820,0.587,-0.172,0.690,0.604,0.230
Algeria,2014,6.355,9.335,0.818,65.140,,,,0.626,0.177
Algeria,2016,5.341,9.362,0.749,65.500,,,,0.661,0.377
```

Figure 1: data overview

3.2 Data processing

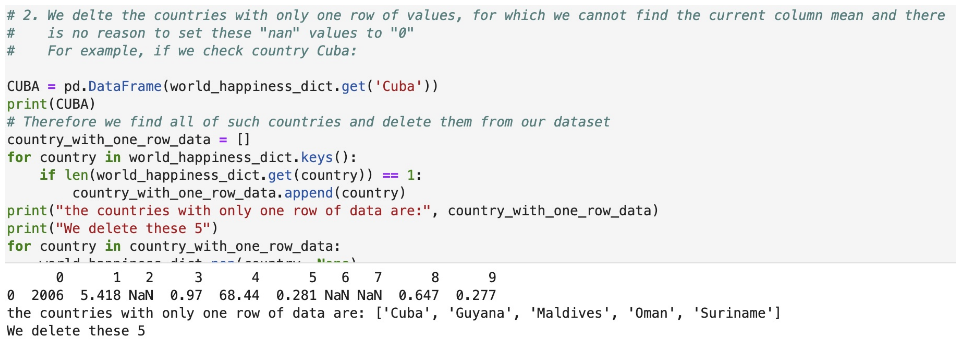
We initially create a dictionary in which keys are country names and values are the data belong to the countries.



```python
# 1. create a dictionary with keys of countries and values of their data:

world_happiness_dict = defaultdict()
for row in range(world_happiness_shape[0]):
    value = world_happiness.loc[row].tolist()
    keys = world_happiness_dict.keys()
    current_country = world_happiness.loc[row][0]
    if (current_country in world_happiness_dict.keys()):
        world_happiness_dict[current_country].append(value[1:])
    else:
        world_happiness_dict[current_country] = [value[1:]]
```

Figure 2: saving data to dictionary

Notice that there are several countries of which data have only one columns. In order to avoid the inaccuracy, we find out the total 5 countries with this feature and delete them before the data processing.

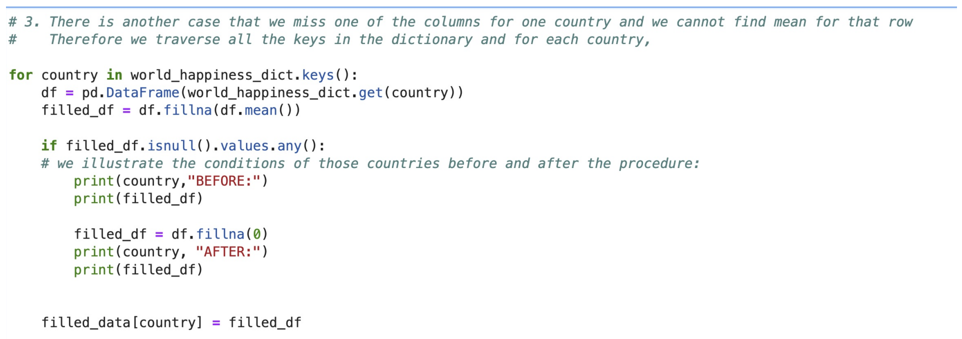


```python
# 2. We delte the countries with only one row of values, for which we cannot find the current column mean and there
#    is no reason to set these "nan" values to "0"
#    For example, if we check country Cuba:

CUBA = pd.DataFrame(world_happiness_dict.get('Cuba'))
print(CUBA)
# Therefore we find all of such countries and delete them from our dataset
country_with_one_row_data = []
for country in world_happiness_dict.keys():
    if len(world_happiness_dict.get(country)) == 1:
        country_with_one_row_data.append(country)
print("the countries with only one row of data are:", country_with_one_row_data)
print("We delete these 5")
for country in country_with_one_row_data:
```

```
      0      1   2     3      4      5   6   7      8      9
0  2006  5.418 NaN  0.97  68.44  0.281 NaN NaN  0.647  0.277
the countries with only one row of data are: ['Cuba', 'Guyana', 'Maldives', 'Oman', 'Suriname']
We delete these 5
```

Figure 3: exceptions in data

Finally, we fill all the remaining "nan" values with the local mean of that column from that country. Specifically, there exist one situation that a whole column of data is missing, and we fill the whole column with 0 value. In this way, all the "nan" values are filled with the appropriate values.

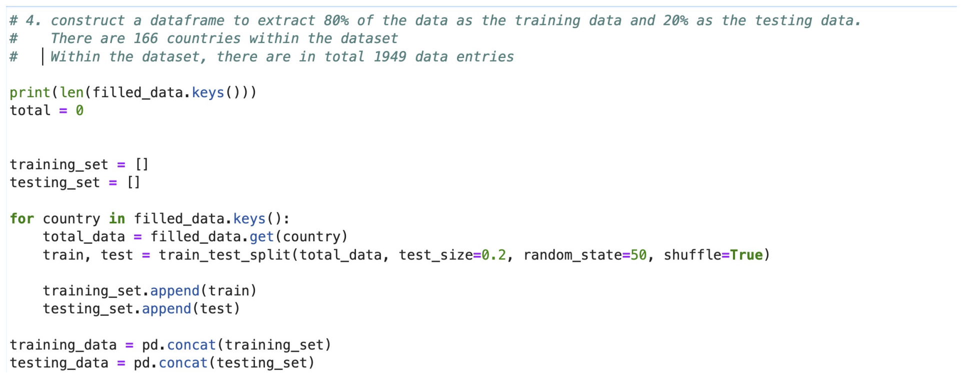


```
# 3. There is another case that we miss one of the columns for one country and we cannot find mean for that row
#    Therefore we traverse all the keys in the dictionary and for each country,

for country in world_happiness_dict.keys():
    df = pd.DataFrame(world_happiness_dict.get(country))
    filled_df = df.fillna(df.mean())

    if filled_df.isnull().values.any():
    # we illustrate the conditions of those countries before and after the procedure:
        print(country,"BEFORE:")
        print(filled_df)

        filled_df = df.fillna(0)
        print(country, "AFTER:")
        print(filled_df)


    filled_data[country] = filled_df
```

Figure 4: filling "nan" values

3.3 Generate training and testing data

We then combine all the data as an aggregate and extract 80% of data as our training data and the remaining 20% as our testing data.



```
# 4. construct a dataframe to extract 80% of the data as the training data and 20% as the testing data.
#    There are 166 countries within the dataset
#    Within the dataset, there are in total 1949 data entries

print(len(filled_data.keys()))
total = 0


training_set = []
testing_set = []

for country in filled_data.keys():
    total_data = filled_data.get(country)
    train, test = train_test_split(total_data, test_size=0.2, random_state=50, shuffle=True)

    training_set.append(train)
    testing_set.append(test)

training_data = pd.concat(training_set)
testing_data = pd.concat(testing_set)
```

Figure 5: generating the training and testing data

4 Experimentation

4.1 Brief overview

We need to determine the optimal amounts of trees that should exist within the random forest regressor and the amount of nodes required for each tree. The reasoning for being against arbitrarily increasing tree and node count is because each increment accounts for more processing power, and the gain achieved by adding more trees has diminishing returns.

4.2 Implementation

Through multiple iterations of adding the tree amount and increasing the possible nodes per tree, results show that roughly 300 trees with 150 nodes produces the most economical result. This testing is done through incremental changes to the tree and node counts until the improvements observed has decreased substantially.

We show a segment of a certain decision tree within the forest, and it is revealed that each tree is a binary tree data structure. Each tree contains nodes that checks a certain feature, with the resulting decision being used to determine which levels of the node the prediction should progress to, or a happiness value is directly returned. The figure below attempts to show a glimpse of what a decision tree's logic of prediction is:

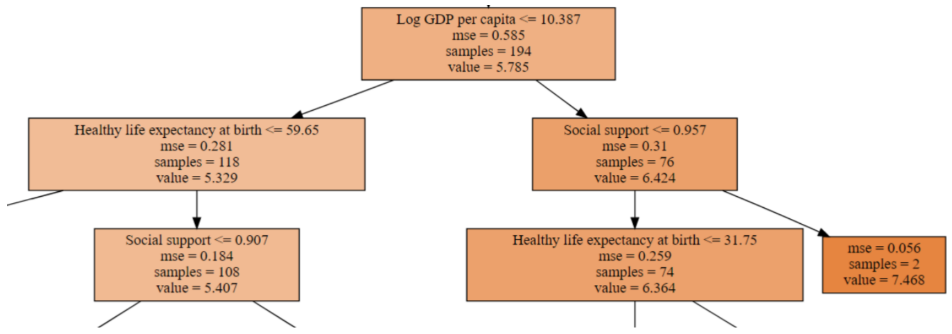


Figure 6: a segment of a decision tree

Each tree's decision is then aggregated together, with a resulting average value being the final output and prediction made by the model.

4.3 Visualization

Another goal we have during experimentation is to see if we can visualize the model in some way and dig deeper into how a random forest regressor works. To do this, we need to take out individual trees from the model and graph out the relationship between nodes.

Luckily, there are already established libraries that can help us achieve this task. All we needed to do is to generate a single tree's binary relational data, and plot that data with mature dependencies like matplotlib. It is important to note that due to the massive size of each tree, a SVG object needed to be produced as any image that comes out of the data would be in low resolution. However, after visualizing the trees we can see the exact numbers and decisions each node takes in order to reach to the next feature analyzing stage or directly output a result, which can allow us to fine tune some exact numbers if it was needed.

5 Results

5.1 Model performance

We first need to measure the performance of the model, we took three approaches in an attempt to see how well it performs against the testing data-set that we have split.

The first approach is a more straightforward naive way of measuring performance. To do this, given that we have n samples in the testing set we took the following:

$$\sum_{n=1}^{n} \frac{abs(predict_n - truth_n)}{truth_n}$$

The resulting sum is then averaged over the length of the data-set, and we concluded the predicted happiness score had a 3% difference on average compared to the actual happiness score.

Another method we used to test for errors is to calculate the root mean square error of the random forest regressor model, to do this, we can use the standard formula:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(\frac{d_i - f_i}{\sigma_i}\Big)^2}$$

This resulted in a MSE loss that was observed to be around 0.19, which showed a wider disparity between the model and its actually performance.

Finally, we move on to calculate the coefficient of determination R^2 for the model, to do this, we need to establish a few things.

First establish that:

$$u = \sum_{n=1}^{n} (truth_n - predict_n)^2$$

$$v = \sum_{n=1}^{n} (truth_n - mean_n)^2$$

Afterwards, the coefficient of determination R^2 can be calculated as:

$$R^2 = 1 - \frac{u}{v}$$

This is a more common way to score performance of a random forest regressor model. For our case, we scored a 0.84, as the ideal score is as close to 1 as possible, it showed a decent performance in how our model handles data points within the real world.

Now, we can move on to use the model in giving us predictions and insights about the data-set.

5.2 Most significant features

Before we discuss the results derived from the trained model, it is important to first note some assumptions. First of which is that we are assuming the world as a whole acts as a super organism instead of regional partitions that each have their own unique differences. It is also important to note the euro-centric view of happiness that is given within the data-set, as there are many assumptions on what features would contribute to happiness. Now, with those assumptions in mind, let's proceed on to the results.

Beginning with the most important features that are required to improve the world happiness score, we turn to individual feature scaling. Through incremental changes to each feature individually, we generate a happiness score prediction. The highest happiness score generated by each feature scale is then taken as the result. For our experiment, we chose to scale each feature by 1%, 10%, 50%, and 100%. The results are shown below.

Table 1: Most significant improvement in scaling individual features

Improvement	Best Feature
1%	Social Support
10%, 50%, 100%	Log GDP per Capita

The model gave the prediction for if there was immediate improvement to the world's average happiness score, the focus should be on social support and related support programs for the vulnerable people. However, if there were to be longer term and more significant improvements in happiness, the average economic situation should be improved instead.

5.3 Same scale happiness

We then combine all the data as an aggregate and extract 80% of data as our training data and the remaining 20% as we also wanted to find out how much scaling a single feature requires in order to elevate world happiness by the same amount. To achieve that, we set a static happiness goal and took the world features average. Then each feature is individually scaled until the happiness goal can be achieved or a local upper limit has been observed. Here are the results we observed:

Table 2: Scaling needed to increase world average happiness by 5%

Feature	Scale increase
Log GDP per Capita	6%
Social Support	8%
Healthy life expectancy at birth	7%
All other features	$undefined$

The reasoning for undefined data will be explained in the next section, as we discuss some inherent limitations that plague the random forest regressor model.

6 Limitations

One direct limitation of random forest regression is that the predictions the model can make is limited to the upper and lower bounds of the training data set. The reasoning behind this limitation is due to how decisions are being trained, it is important to note that each tree's decision is based off of values given by training data. This conclusion means each tree's final output is limited by the range of data it has seen, and causes the entire forest to be unable to extrapolate on new data in foreign ranges.

Going back to the results section, this is why some features cannot be determined. Since with all the other features set at a static level, increasing one single feature can easily hit the maximum happiness score that can be produced, causing a upper bound limit which will never output high enough values. This is a critical limitation of random forest regression, as it prevents the model itself to generate accurate predictions out of the training data range.

Another limitation of random forest regression is the fact that each prediction requires every single feature to be present, with no exception for missing features. This is because in the forest of decision trees, some trees may depend on the feature that is absent. If we ignore the decision trees that rely on the missing value, the rest of the tree aggregation prediction may output bad results. That is due to the outcome of certain over-fitted portions of the forest with no regularization trees, as those trees may have been dependent on the features that were not provided to the model.

7 Additional work

There is a lot of room for additional work, we believe this model can be improved in its data sanitation, training algorithm and model complexity.

The first problem that needs to be addressed is the model's inability to extrapolate out to new, unseen data ranges. Clearly, a random forest regressor would no longer be able to fit that demand, therefore a new type of algorithm would need to be used. Noting the non-linear relationship between features and happiness, it is also important to view the context of the data-set and recognize that the world in which we live in right now cannot be considered an utopia. That means with the increase of certain features, happiness is destined to increase alongside with it, creating a short term linear relationship amongst the two. It is important however to also note that this linear relationship would stop at some point, as the wealthiest countries on earth does not have the equal multiplication of happiness compared to some of the poorest.

This leads to another sector that could use some improvement, and that is data sanitation. The world happiness data-set contains many non existent values, and countries with whole features missing. Our method of implication data is considered to be more naive since we cannot easily assume an average of a column to be a somewhat accurate measure of the given country. The best way is to use algorithms that can handle missing values, an addition of a KNN model to estimate missing values might be an interesting way to resolve this problem, as we can take advantage of ensemble learning to fill in gaps and downfalls for our data-set and training methodologies.