Intro to Data Mining – PhD Jamir Jafari

The economics of happiness

Final Project

Lilian Sao de Rivera Group #4
4-22-2019

Table of Contents

List of Tables

List of Figures

Introduction

Retiring in the United States is getting harder because of the costs of living. Sites that help people to plan for the future usually states that it is necessary to have one million dollars to live comfortably when people retired. Many Americans do not have enough money to cope with an emergency even if they are educated. The debts that Americans carry may be high due to education loans, house loans, and health expenses. However, some Americans have coped with these problems at the end of their lives by looking for cheaper places to live, where their savings and retirement income can support them a good life. However, these locations may be outside of the country. Before taking any decision, the average American that is willing to take this decision should have information about these locations. This type of decisions not only need to be related to retirement but also about buying a vacation home or having a gap year in an exotic place.

This type of decisions has to do with wellness, and for this reason, the index of happiness is a good indicator to choose a country. If someone decides to go on an adventure, wellness, security, and economic stability are good indicators. The index of happiness created by the Canadian Institute for Advanced Research measures the level of happiness using a poll that measures several aspects of social life. The GDP is the only economic indicator, and the perception of government trust is used, but it should be highlighted the word perception. Thus, there should be another way to predict this happiness. GDP should be kept on the analysis, but other features are included like, the economic group of each country, the GINI indicator, that is the gap between rich and poor people. Is a country with a big gap happy? Do worldwide governance indicators contribute to well-being and happiness of the country? Indeed, good indicators related to economics will allow anyone to analyze the economic stability of a country but also happiness. If the country is happy and then the related features about economic stability are right, then a good investment can be made.

Base in these assumptions the research question can be stated as follows:

> **"Can economic indicators like GDP, GINI, and governance performance indicators predict the status of happiness reported by the Canadian Institute of Advanced Research which is based on polls of perception about happiness and social stability?"**

This research looks also into the contribution of these factors in the index of happiness, if there is a contribution. Although the main goal of the project is to look for evidence that economic and political stability are better predictors of happiness in a country.

This report presents the results of the project into four parts. The first part is the description of the datasets and challenges that were found and solutions to deliver one dataset that would help to make the projections. The second part is the description of the machine learning algorithm that was used and the way the model was applied to the final dataset. The third part is the description of the results that were obtained. Lastly, the conclusions and recommendations are made at the end of the paper.

Datasets Description

The information to classify the level of happiness is contained in four datasets.

Happiness index found in Kaggle web site, although the information is originally from the Canadian Institute for Advanced Research.

Table 1. Happiness Index info

Years of Information	2017
No. Observations	156 countries
No. of Features	12
Type of file	CSV file
Challenge	The only feature to join with other indicators is the name of the country which in some cases varies from the other datasets.

Table 2.Happiness description of features

Country	Name of the Country
Happiness.Rank	Averall rank of the country
Happiness.Score	Score of happiness Range 1 to 8
Whisker.high	95% of score
Whisker.low	5% of score
Economy..GDP.per.Capita.	% of contribution GDP per capita
Family	% of contribution Family relations
Health..Life.Expectancy.	% of contribution
Freedom	% of contribution Perception of freedom
Generosity	% of contribution Perception of Generosity
Trust..Government.Corruption.	% of contribution Perception of Trust
Dystopia.Residual	% of contribution not accounted factors and lowest score previous year.

Governance indicators from the World Bank and the Brookings Institution.

Table 3. Governance indicators general description

Years of Information	1996 thru 2018
No. Observations	216 Countries
No. of Features	7
Type of File	Excel File
Challenge	once excel sheet per indicator with several information columns besides the estimates that are needed for the project. There are missing values.

Table 4. Governance indicators feature description

Voice accountability	Index from -2.5 to 2.5
Political Stability and Absence of Violence/Terrorism	Index from -2.5 to 2.5
Governance Efectiveness	Index from -2.5 to 2.5
Regulatory Quality	Index from -2.5 to 2.5
Rule of Law	Index from -2.5 to 2.5
Control of corruption	Index from -2.5 to 2.5

GDP indicators from the World Bank database

Table 5. GDP dataset general description

Years of Information	2006 thru 2017
No. Observations	264 Countries
No. of Features	63
Type of File	Excel File
Challenge	There are missing values.

Table 6. GDP indicator description of features

CountryName	Country Name
CountryCode	ISO Code
Indicator Name	Long name for indicator
IndicatorCode	Code of the indicator
Year1960-2018	GDP for each year, not all countries have information. The amount of the indicator is an amount in dollars

GINI indicators from the World Bank

Table 7. GINI dataset general description

Years of Information	2006 thru 2017
No. Observations	264 Countries
No. of Features	63
Type of File	Excel File
Challenge	There are missing values.

Table 8. GINI indicator description of features

CountryName	Country Name
CountryCode	ISO Code
Indicator Name	Long name for indicator
IndicatorCode	Code of the indicator

Year1960-2018	GINI for each year, not all countries have information. The indicator ranges from 26 to 65

ISO dataset from Wikipedia

Table 9. ISO dataset general description

Years of Information	N/A
No. Observations	264 Countries
No. of Features	9
Type of File	CSV
Challenge	First char has an odd value

Table 10. ISO dataset feature description

Country	Country
Alpha-2-code	ISO 2 CHARS
Alpha-3-code	ISO 3 CHARS
Numeric code	N/A
Link to ISO 3166-2 subdivision codes	N/A
Independent	N/A

This dataset is going to be used to classify the countries in the happiness dataset. The GINI and GDP and the governance indicators have the ISO in place. These three datasets were created under the World Bank supervision.

Methodology

Data Cleaning, Transformation, Imputation of missing data

Each dataset has its own characteristics, although the GINI and the GDP have a similar structure. Only the dataset corresponding a happiness is a csv, the rest of them are excel files so a unique approach has to perform on each one. We describe the general producer to clean each dataset.

ISO Dataset

1. The dataset was loaded with pd.read_csv
2. The first character of the country name was eliminated, these data was obtained from the Wikipedia site.
3. The data then is ready to be used for the happiness dataset.

```
ISO = pd.read_csv(path_files+'ISOCODES.csv')
q=ISO.iloc[:,0]
m=q.apply(lambda x: x[1:len(x)])
ISO.iloc[:,0] =  m
```

Happiness Dataset

1. The dataset was loaded with pd_read.csv
2. A subset of the data was obtained with only the feature country from the dataset
3. This subset was merge using the left option to find the correct ISO codes in the ISO dataset.
4. The resulting datasets was scanned to look for nulls, the name of the country for this null values was changed to match the names on the ISO database.
5. Then the merge was made again to get the correct codes.
6. The final result was merged with the happiness score.

```
happiness_subset["Country"]= y["Country"]
happiness_subset =  pd.merge(happiness_subset, result_final, on="Country",
how='left')
print(happiness_subset.head())
```

GDP dataset

1. The dataset was loaded with pd.read_csv skipping the first lines
2. Some values are empty so the column of most recent year previous to data missing were copied.
3. The data then was filtered by year , only the year 2017 was taken from the dataset.
4.

```
def GDP_fill_null_values():
    for i in GDP_index_null:
        row_GDP = pd.DataFrame(GDP.iloc[i, 4:])
        row_GDP1 = row_GDP[i].fillna(method='ffill')
        GDP.iloc[i, 4:] = row_GDP1


GDP_fill_null_values()

GDP_subset = GDP[["Country Code", "2017"]]

GDP_index_null = GDP_subset[GDP_subset["2017"].isnull()].index
```

5. This was done for the countries with no data in 2017

GINI dataset

1. The dataset was loaded with pd.read_csv skipping the first lines
2. Some values are empty so the column of most recent year previous to data missing were copied.
3. The data then was filtered by year , only the year 2017 was taken from the dataset.

```
def GINI_fill_null_values():
    for i in GINI_index_null:
        row_GINI = pd.DataFrame(GINI.iloc[i, 4:])
        row_GINI1 = row_GINI[i].fillna(method='ffill')
        GINI.iloc[i, 4:] = row_GINI1


GINI_fill_null_values()

GINI_subset = GINI[["Country Code", "2017"]]
GINI_subset.columns = ["Country_code", "GINI"]
GINI_index_null = GINI_subset[GINI_subset["GINI"].isnull()].index
```

```
GINI_subset = GINI_subset[GINI_subset["GINI"].notnull()]
```

4. This was done for the countries with no data in 2017

Governance Indicators dataset

1. Each indicator was read from the worksheet in the Excel workbook using pd.read_excel
2. The name of the columns where created from the rows, one row contained the year and the other row the name of the column, for example : row0 = 1997 row1 = Estimate, the final result was Estimate_2017.

```
def creates_colnames(estimate_cols):
    columns = []
    len_cols = estimate_cols.shape[1]
    for i in range((len_cols)):

        row1 = str(estimate_cols.iloc[1, i])
        row0 = str(estimate_cols.iloc[0, i])

        if pd.isnull(row0) or row0 == 'nan':
            row0 = ""
        if pd.isnull(row1):
            row1 = ""
        if row0 == "":
            final_row_name = row1
        else:
            final_row_name = row1 + "_" + row0
        columns.append(final_row_name)

    return (columns)
```

3. The columns of the datasets where name with this information.
4. Then a subset was obtained filtering only the columns with name 'Estimate*'

```
# Voice
col_names = ["Country/Territory","WBCode"]
col_names1= [col for col in voice.columns if 'Estimate' in col]
col_names.extend(col_names1)
voice_subset = voice[col_names]
```

5. The null values were identified from the results and were filled with the most recent value.

```
def ESTIMATE_fill_null_values(index_null, EST_dataset):
    for i in index_null:
        # if data_v == 1:
        row_EST = pd.DataFrame(EST_dataset.iloc[i, 2:])
        row_EST1 = row_EST[i].fillna(method='ffill')
        # if data_v == 1:
        EST_dataset.iloc[i, 2:] = row_EST1
```

6. This was done for each worksheet.

7. There were 6 final dataframes that were merge together to get one last database for governance. All of them contained the IS0 code.

```
# Joining all the values
#Voide
next_subset= voice_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","VoiceandAccountability"]
final_dataset = next_subset.copy()
#PoliticalStabilityNoViolence
next_subset= political_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","PoliticalStabilityNoViolence"]
final_dataset = pd.merge(final_dataset, next_subset, on="Country_code", how="inner")
#Effectiveness
next_subset= effectiveness_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","GovermentEffectiveness"]
final_dataset = pd.merge(final_dataset, next_subset, on="Country_code", how="inner")
#Regulatory
next_subset= regulatory_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","RegulatoryQuality"]
final_dataset = pd.merge(final_dataset, next_subset, on="Country_code", how="inner")
#Ruleoflaw
next_subset= ruleoflaw_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","RuleofLaw"]
final_dataset = pd.merge(final_dataset, next_subset, on="Country_code", how="inner")
#ControlofCorruption
next_subset= corruption_subset[["WBCode","Estimate_2017"]]
next_subset.columns = ["Country_code","ControlofCorruption"]
final_dataset = pd.merge(final_dataset, next_subset, on="Country_code", how="inner")
```

Data Integration and Normalization

The previous process created four dataframes with ISO codes and the information for 2017 year.

1. The name of the ISO code was changed in the four dataframes so they could be merged in a more transparent way.
2. The four dataframes were joined together in a resulting data.

```
final_happiness = happiness_subset.copy()
final_happiness = pd.merge(final_happiness,GDP_subset, on="Country_code")
final_happiness = pd.merge(final_happiness,GINI_subset, on="Country_code")
final_happiness = pd.merge(final_happiness,final_dataset, on="Country_code")
#print(final_happiness)
```

3. This data was normalize using the scikit-learn package so an EDA analysis could be more significant for the final user.

```
# Create the Scaler object
scaler = preprocessing.StandardScaler()
# Fit your data on the scaler object
scaled_happ = scaler.fit_transform(numerical_data)
scaled_happ = pd.DataFrame(scaled_happ, columns=happ_columns)

str_data = final_happiness[['Country','Country_code','Happiness.Scale']]
ff_happiness = pd.concat([str_data,scaled_happ], axis=1)
```

4. The information was binned into four categories : “Happy”, “Med Happy”, “Low Happy”, “Not Happy”. This clustering is used in the prediction models.

5.
```
def create_scale(x):
    '''
    Create scale to bin the happiness score
    :param x:
    :return:
    '''
    scale = ((x >= 5.5) & (x < 6.5)) * 2 + (x >= 6.5) * 1 + ((x >= 4.5) & (x <
5.5)) * 3 + ((x >= 0) & (x < 4.5)) * 4

    return scale


mbin = happiness["Happiness.Score"]
happiness_subset['Happiness.Scale'] = mbin.apply(create_scale)
```

6. Finaly the dataset was saved into a csv file to be used by the application for the EDA analysis and the ML algorithms

```
ff_happiness.to_csv(path_files+"final_happiness_dataset.csv", index=False)
```

ML Methodology

An application is PyQT5 was created to perform the ML algorithms and to provide a tool easy to manipulate for the final user. This IDE provides opportunities to manipulate the parameters an to analyze the information in a more interactive way. This means that a user of the model would have the chance to look into the information by using graphs and make inform decisions about some parameters in the ML algorithms. A Decision Tree ML algorithm along with Random Forest classifier were chosen to perform the prediction. This model was chosen because the target is clustered in four bins and it is expected that the model predicts if a country is happy base on the n factors. A regression algorithm could be a good option too, but it is important to analyze which variables are important to the model and the Radom Forest gives these results.

Decision Tree Model

The depth was set by default in three, because there only eight features to analyze. This parameter can be changed to get other results, but the model showed that this is one of the best values. The other value that was set to be changed, so there can be made more experiments on the data was the size of the test datasets. The default value was set in .30 which in several trials showed the best performance. The features can be changed also to evaluate how the model performs with or without some features. The gridi function that used was “entropy”, because most of the data is continuous and also to avoid confusion with GINI index used as feature which has a completely different meaning from the GINI function in the DT algorithm.

```
# split the dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X_dt, y_dt, test_size=vtest_per,
random_state=100)
# perform training with entropy.
```

```
# Decision tree with entropy
self.clf_entropy = DecisionTreeClassifier(criterion="entropy", random_state=100,
max_depth=vmax_depth, min_samples_leaf=5)

# Performing training
self.clf_entropy.fit(X_train, y_train)

# predicton on test using entropy
y_pred_entropy = self.clf_entropy.predict(X_test)
```

Random Forest Classifier

In this case the features can be modified and the value of the test sample. The contribution of the variables that shows the RT algorithm is a useful tool to experiment with the features to be used by the model. The algorithm is simple and because it is implemented in such a way that is parametrizable by a user, a random state of 100 was used, so the results were always the same under the same combination of features and sample size.

```
# split the dataset into train and test

X_train, X_test, y_train, y_test = train_test_split(X_dt, y_dt, test_size=vtest_per,
random_state=100)

# perform training with entropy.
# Decision tree with entropy

#specify random forest classifier
self.clf_rf = RandomForestClassifier(n_estimators=100, random_state=100)

# perform training
self.clf_rf.fit(X_train, y_train)
```

Results

PYQT5 Application

The purpose of the applications was to create and end user tool to analyze the final dataset and to manipulate which features were included in the model. The other purpose was to include a way to change parameters of the ML algorithms without changing the code

The design of the applications is as follows

EDA Analysis

Happiness Distribution

Scatter Plots and regression line for each feature against the score

Correlation plot of the features

Dashboards for Machine Learning Algorithms

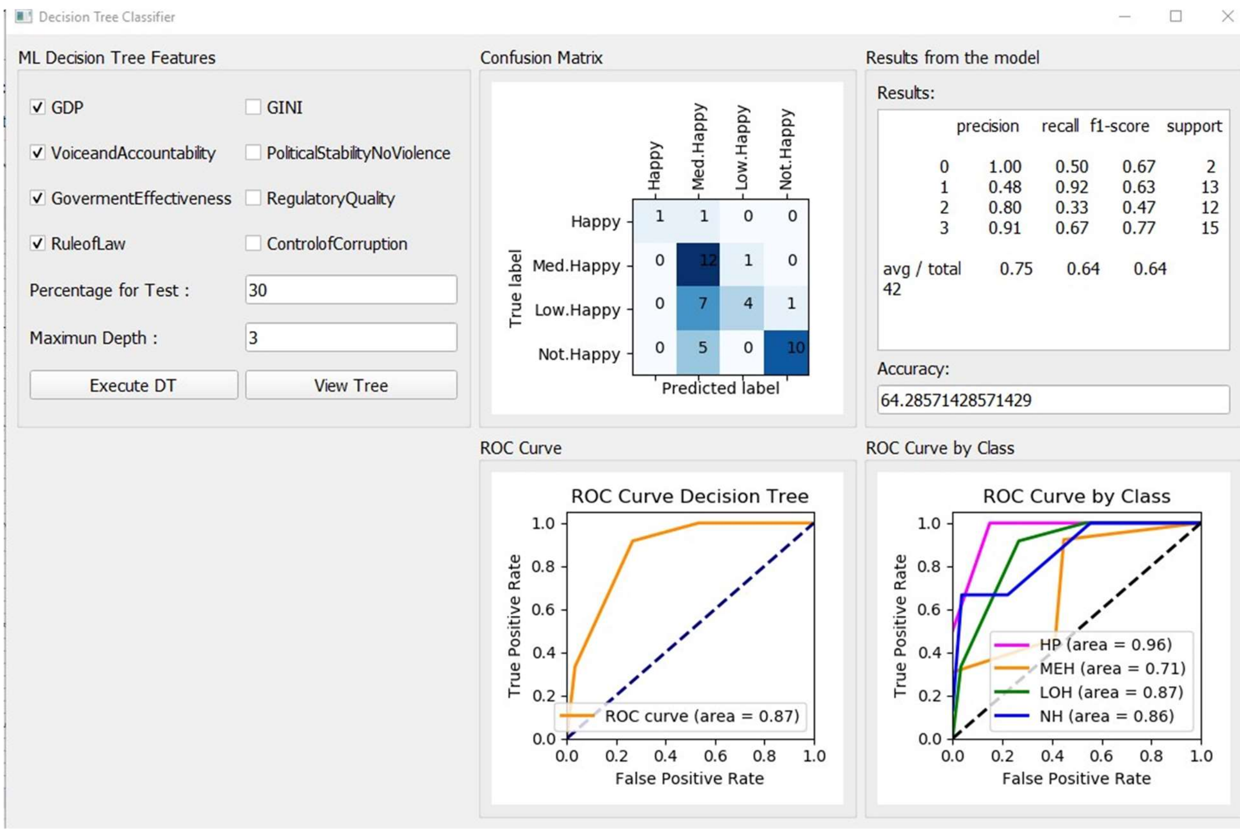
Decision Tree Dashboard

Random Forest Dashboard

Parameters are introduced in most of the options. Thus, the results can be changed on run time by an end user. This characteristic makes the dashboard useful for analysis and generation of insights.

Below is an example of the dashboard for the Decision Tree classifier. The features and parameters are located on top left corner. They can be changed to create results on the fly. The rest of the boxes represent results for the Decision Tree algorithm.

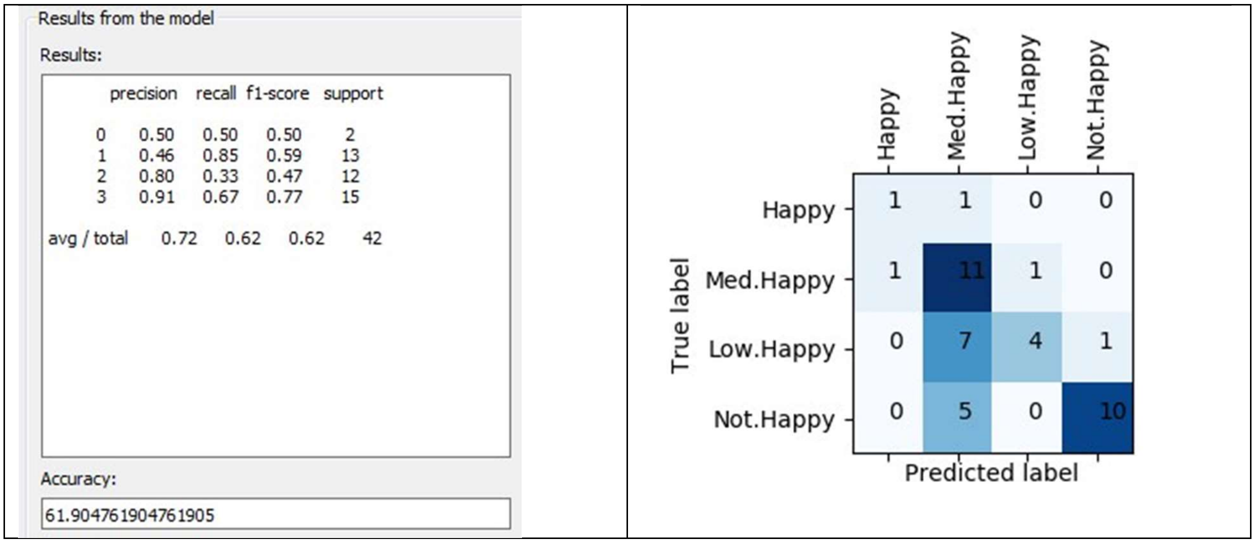
Figure 1. PyQt5 Dashboard for Decision Tree



Decision Tree Results

The results in the Fig.2 below shows that for all the features, with 30 percent sample and a depth of three the model has a result of 61 with a precision of 71. This accuracy is low for what is expected. The model does not seem to present a good prediction. We can see in the confusion matrix plot also in Figure 2, that there are two missing value in the medium level that should be classified as high. This is the principal value for us in this case.

Figure 2. DT Results and Confusion Matrix



Results from the model

Results:

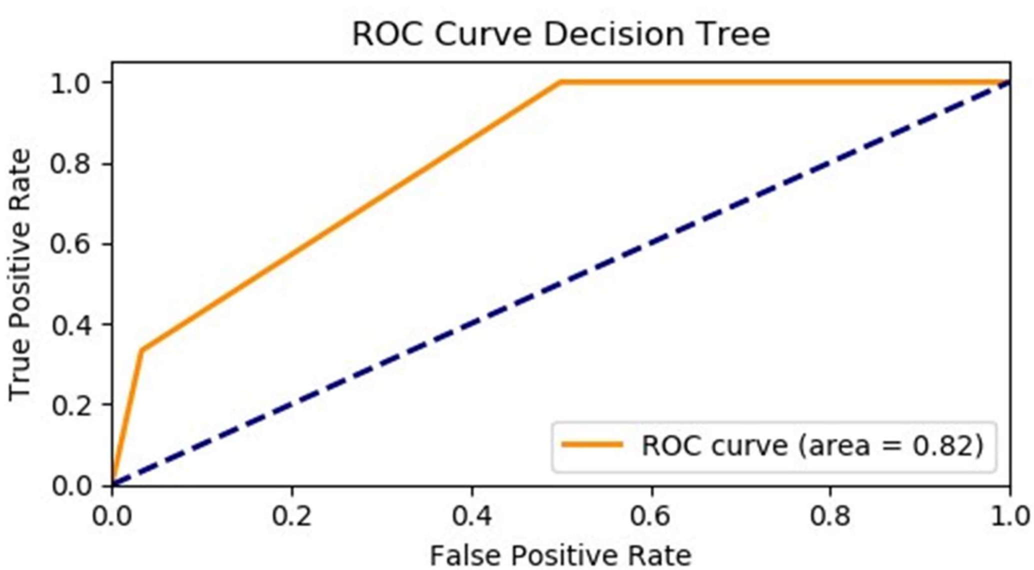
	precision	recall	f1-score	support
0	0.50	0.50	0.50	2
1	0.46	0.85	0.59	13
2	0.80	0.33	0.47	12
3	0.91	0.67	0.77	15
avg / total	0.72	0.62	0.62	42

Accuracy:

61.904761904761905

The ROC curve for the model shows ROC curve 0.82 as area under the curve this means that in most cases makes a good result, but still if we see the data in the Figure 1, the Happiness is not well predicted which do not give enough information.

Figure 3. DT ROC Curve



Random Forest Results

For the Random Forest model the results are better. The reports in Figure 4 below show that the precision is 0.75 and the general accuracy is 71% percent. This model seems more suitable than the Decision Tree model.

Figure 4. RF- Results s

Results from the model

Results:

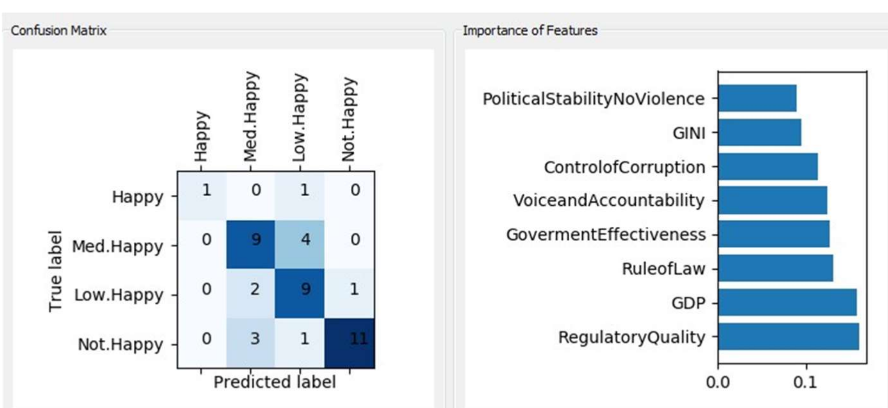
	precision	recall	f1-score	support
0	1.00	0.50	0.67	2
1	0.64	0.69	0.67	13
2	0.60	0.75	0.67	12
3	0.92	0.73	0.81	15
avg / total	0.75	0.71	0.72	42

Accuracy:

71.42857142857143

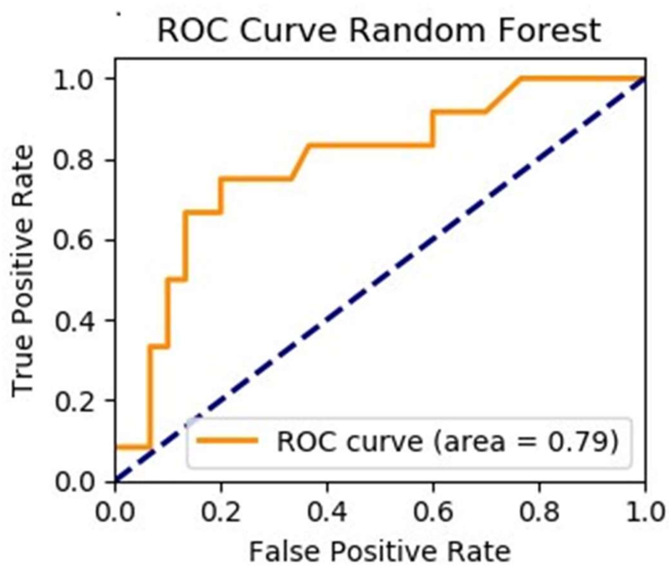
In the confusion matrix the classification seems more accurate and the prediction of the happiness seems more accurate in the Figure 4 below, than the Decision Tree prediction. The Figure 4 shows also the importance of the variable GDP and Regulatory Quality are the ones with most contribution to the model.

Figure 5. RF – Confusion Matrix and Importance of Features



The Figure 6 below shows 79% of area under the curve which continues to be a better prediction model. we can see that the line is well on the left upper corner. Random Forest shows with the same parameters used int DT continues to deliver a better outcome.

Figure 6. ROC Random Forest



Summary and Conclusions

The Decision three model does not seem a good candidate for this model, it could be possible that a regression model could be more suitable for the present information. However, the Random Forest approach seems a good model. The final information contains only 140 records with complete information with only one year. The amount of information can be a problem. There also seems that are missing features that are contributing to the information. In the original happiness dataset was a health and life expectancy factor that was not accounted for in this model. There also education factors that can be included to create a better result.

The original score was created using a poll and GDP, the purpose of this project was to generate similar results using only economic indicators. Thus, there is some uncounted information that is missing that needs to be included so the model is more reliable.

However, the Random Forest is a better solution for this project that the Decision Tree, but it could be useful to use a regression model to experiment on the final results.

References

Graham, C. (2017, December 1). *The human costs of the productivity paradox in the USA: Insights from metrix of well-being.* Retrieved from Brookings: https://www.brookings.edu/research/the-human-costs-of-the-productivity-paradox-in-the-usa-insights-from-metrics-of-well-being/

Helliwell, J.F., Layard, R., Sachs, J.D. (2017). Word Happiness Report 2017. *Canadian Institute for Advanced Research.* Retrieved from https://s3.amazonaws.com/happiness-report/2017/HR17.pdf

Kaggle. (2017). Global indicators happiness. Retrieved from https://www.kaggle.com/dgscharan/data-set-for-happines

VenederPlas, J. (Dec, 2016). Python Data Science Handbook. *O'Reilly Media Inc*.

World Bank.(2017). GDP per country. *World Bank Data.* Retrieved from

World Bank.(2017). GINI per country. *World Bank Data.* Retrieved from

World Bank.(2018). World Wide Governance Indicators. *World Bank Data and Brookings Institution.* Retrieved from http://info.worldbank.org/governance/wgi/index.aspx#reports