World Happiness Analysis

Melissa Marin | Noelia Vasquez | Mariana Isidoro | Verania Calderon

Introduction

World happiness, why? We hope to find the understanding of the factors that contribute to other countries happiness.

Determining the journey of how to unpack the data we chose Postgres as our database. Using supervised machine learning. And Tableau to create our visuals.

Data Sources:

- World-Happiness-report.2005-2020.csv
- MortalityDataWHR2021C2.csv
- Wikipedia-iso-country-codes

Research Question:

- What is the main well being factor that contributes to increase the happiness around the world?
- What can the least happiest of countries improve to achieve the same level of happiness in their respective countries?
-

World Happiness

Analysis

- Reviewing the state of happiness from most countries.
- Extracting factors that contribute to happiness.
- Comparing countries numbers to identify key factors to their state of happiness.

Technologies

1. PostgresSQL
2. Python
3. Tableau

Language & Tools

1. CodeTools: VSCode
2. quickdatabasediagrams

Communication

1. Slack
2. Zoom

Data Sources

The columns following the happiness score estimate the extent to which each factor –

A. Log GDP per capita,
B. Social support,
C. Population
D. Healthy life expectancy at birth,
E. Freedom to make life choices,
F. Generosity ,
G. Perceptions of corruption,
H. Positive affect,
I. Negative affect,
J. Confidence in national government.

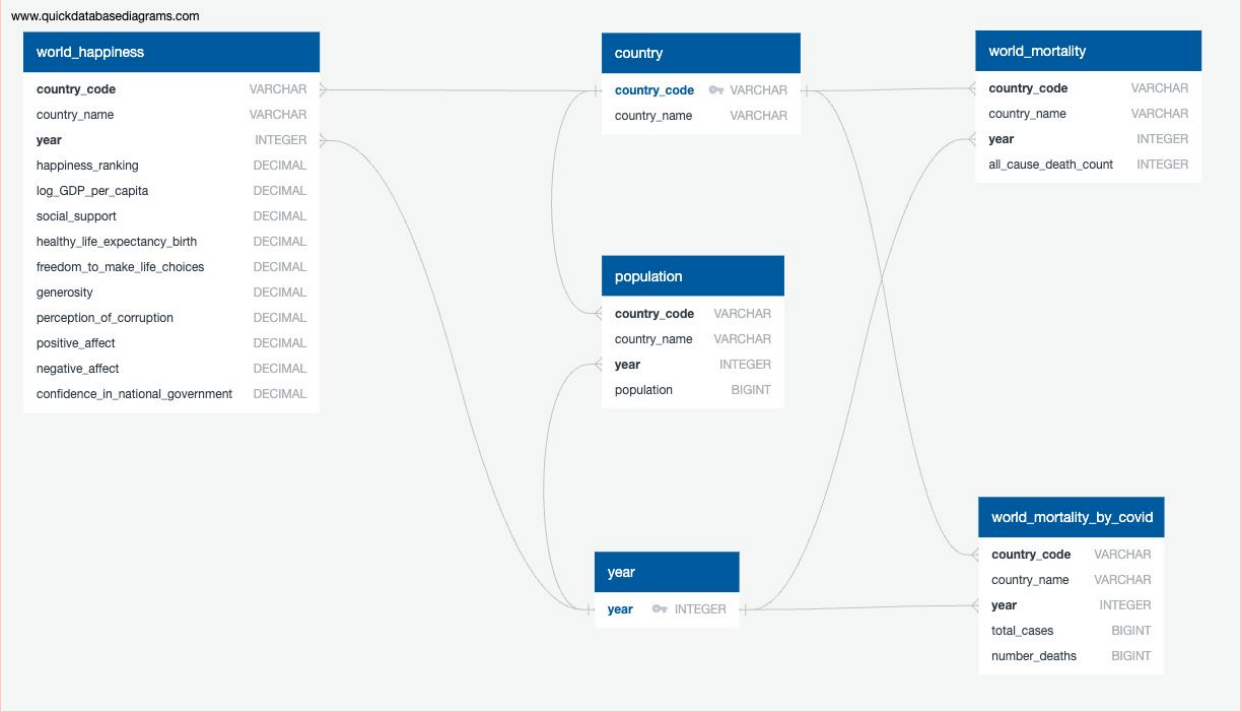
MortalityDataWHR2021C2.csv

World-Happiness-report.2005-2020.csv

- We found our datasets on Kaggle and reflected on existing models

Database Model

Postgres as our relational database management system.



Machine learning

```
# Creating the independent and depend variables
y = data_model['Life Ladder']
X = data_model[['Log GDP per capita','Social support', 'Healthy life expectancy at birth','Freedom to make life choices','Generosity' ,
                'Perceptions of corruption','Positive affect', 'Negative affect','Confidence in national government']]

# Splitting the data into train and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.6, random_state=1)
```

✓ 0.2s

Supervised Machine Learning

1. Independent and Dependent variables
   - Our independent variables: *Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity , Perceptions of corruption, Positive affect, Negative affect,Confidence in national government*.
   - The dependent: *ladder_score*.
   - We have defined our dependent and independent variables and splitting them into test and train data.

2. Model Testing

- After analyzed the Data Sources we decided on Supervised Machine Learning model.
- We calculated the Mean Absolute Error (MAE):
- In order to choose the best model, we calculated the r2_Score value for both models.
- We can conclude from the table(right), the models with the minimum MAE are Random Forest Regressor and XGBoost.

```
# Creating a predefined function to test the models
def modelfit(model):
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    mae = metrics.mean_absolute_error(y_test, preds)
    return (round(mae,4))
```

✓ 0.1s

	Model	MAE
0	Linear Regression	0.4354
1	Random Forest Regressor	0.3280
2	XGBoost	0.3620
3	Decision Tree	0.4749
4	Bayesian Linear Model	0.4179

Continuation to our Model Testing

```
xg_reg_model = xgb.XGBRegressor(objective ='reg:linear', learning_rate = 0.1, n_estimators=5000)
xg_reg_model.fit(X_train, y_train)
xg_predictions = xg_reg_model.predict(X_test)
xg_predictions
```

```
r2_score(y_test, xg_predictions)
✓ 0.2s
0.7928957647069097
```

Above is the test and train set into XGBoost Model.

(Left) Results.

2.2.Random Forest Regressor Model

```
# Create a random forest regresor.
rf_model = RandomForestRegressor(n_estimators=128, random_state=78)
rf_model = rf_model.fit(X_train, y_train)
rf_predictions= rf_model.predict(X_test)
rf_predictions
✓ 0.1s
```

(Above) We fit the test and train set into Random Forest Regressor Model

(Right) Results

```
r2_score(y_test, rf_predictions)
✓ 0.2s
0.810495830809052
```

3. Applying Random Forest Regressor Model

Here we can see the predicted values generated for Random Forest Regressor Model:

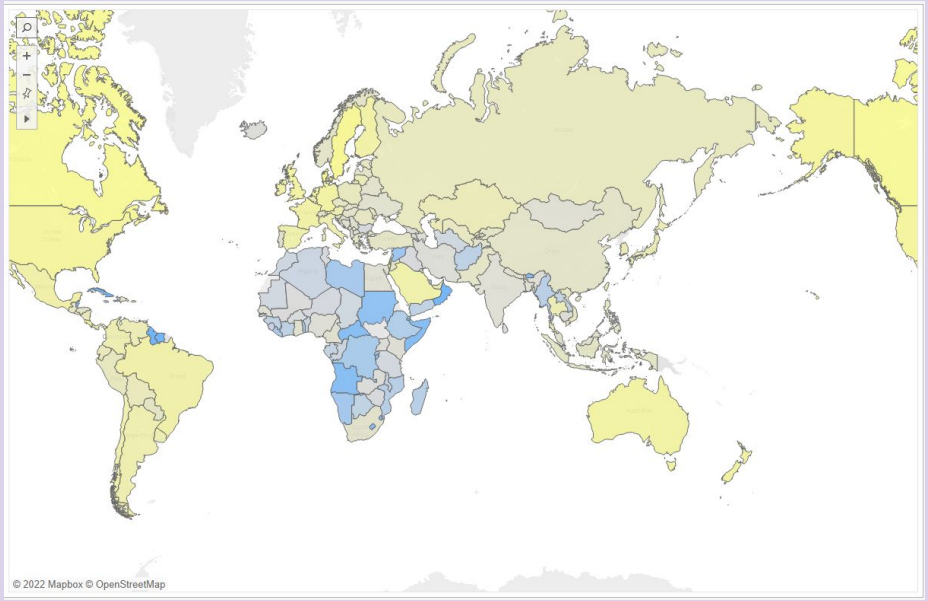
```
# Create a random forest regresor.
rf_model = RandomForestRegressor(n_estimators=128, random_state=78)
rf_model = rf_model.fit(X_train, y_train)
rf_predictions= rf_model.predict(X_test)
rf_predictions
```

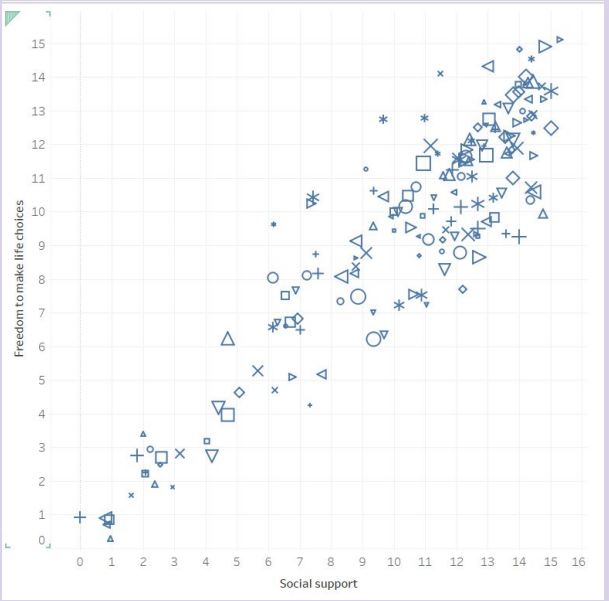
✓ 0.1s

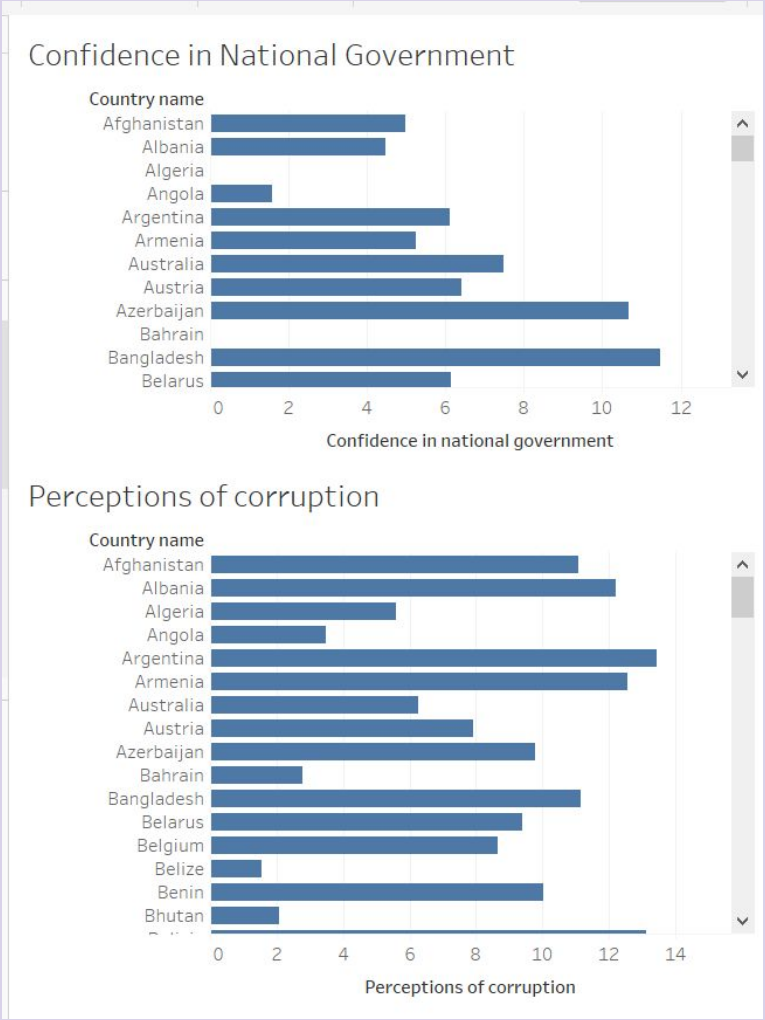
```
array([5.24339532, 5.22115167, 5.82301676, 4.50335405, 4.16570305,
       4.67768635, 5.56302996, 4.18622058, 7.01520208, 4.7821355 ,
       6.99405451, 5.29720026, 4.18685016, 5.2562597 , 7.12793378,
       6.21760345, 5.31306281, 4.75284224, 4.76048971, 7.02421015,
       6.70869939, 6.01395375, 6.4692332 , 6.31652859, 5.96567863,
       4.11109183, 5.68445745, 4.81161454, 5.41482535, 5.30500049,
       5.6316678 , 6.94334947, 5.18157767, 4.81080868, 4.20061303,
       4.94059947, 6.0862235 , 5.57674178, 5.42009015, 5.19388478,
       5.53502683, 4.77176941, 5.60765356, 5.58866989, 4.75688841,
       6.78233653, 5.71008884, 4.77124838, 4.30662296, 4.2475578 ,
       4.8359778 , 4.04351933, 6.63905638, 6.91982763, 4.0731257 ,
       6.55773674, 4.80851352, 5.76807824, 4.23090092, 5.19092772,
       4.05776509, 4.81363655, 7.02470612, 6.37854647, 7.09973475,
       6.94568077, 5.38386142, 6.02390924, 6.47789114, 4.34081595,
       4.10404161, 5.6848231 , 4.79393805, 4.84663585, 4.72536508,
       5.82280659, 4.85171699, 5.92602942, 4.02440187, 6.81860065,
       6.16702673, 5.71274022, 5.28838136, 5.98600262, 5.98201707,
       5.84630829, 4.63133258, 6.00768626, 7.01285248, 5.42761835])
```

Visualizations

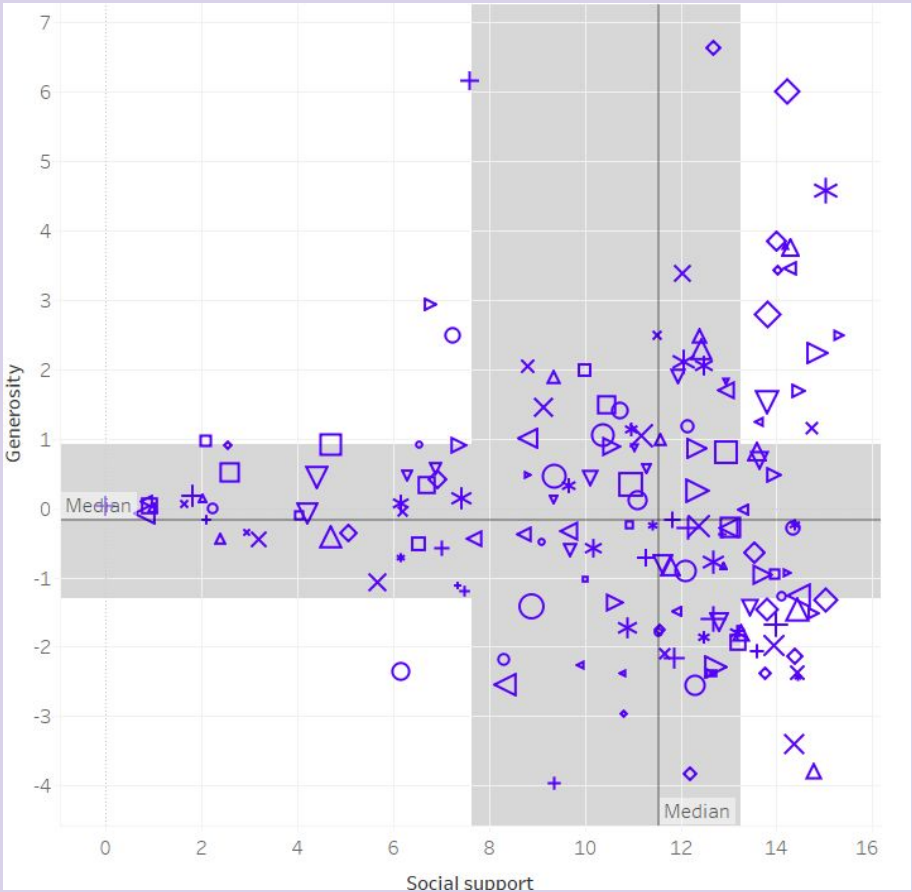
Dashboard:
https://public.tableau.com/app/profile/mariana.isidoro/viz/Rainbow_16619176290710/Colors?publish=yes







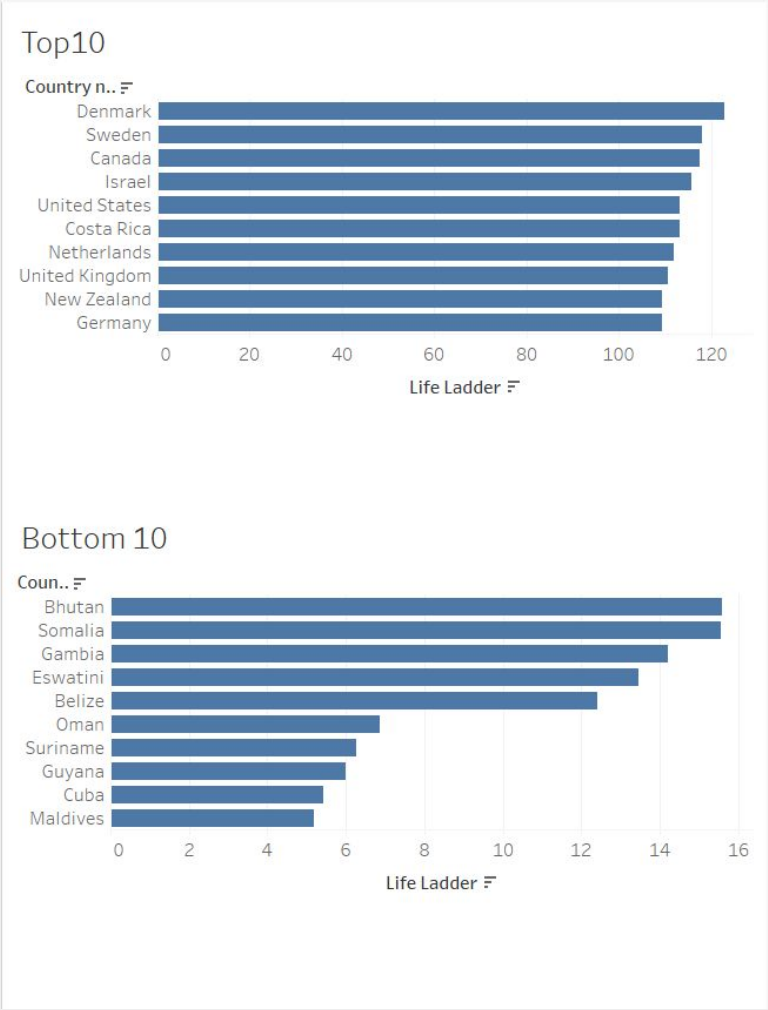
Story Country by Topics Interactive:https://public.tableau.com/app/profile/mariana.isidoro/viz/CountrybyTopic/CountybyTopic?publish=yes



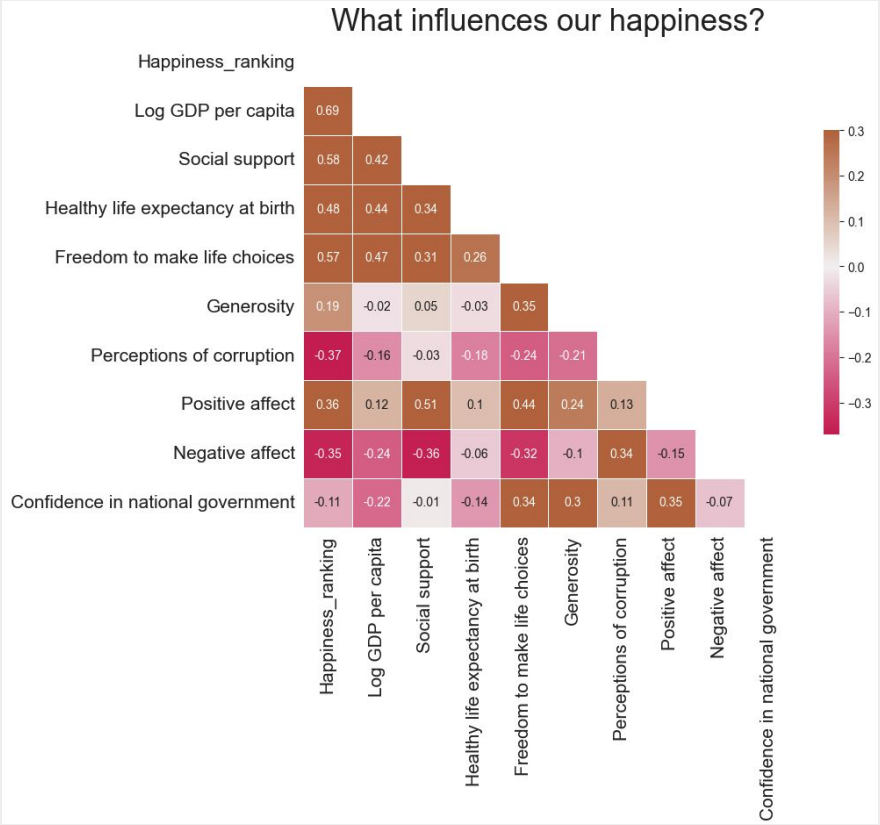
Story:
https://public.tableau.com/app/profile/mariana.isidoro/viz/Happiness_Rainbow/Happiness_Rainbow

Top and Bottom 10 Countries

Measured by Life Ladder.



Correlations Matrix



Recommendation for future analysis

1. Predicting that our top five countries will still remain in the top ten seeing how that been the trend for the past ten years.
2. Learn more about positive psychology to help determine what leads to better perceptions of personal happiness

Questions