Happiness Predictor

Maria, Manuela and Sindy

Project Pipeline

01

- World Health Organization Dataset
- World Happiness Report Dataset
- ISO classification
- Sunshine hours
- Lat/long for cities

Data Acquisition

02

- Data cleaning
- Transformation
- Data merging

ETL Process

03

- Feature engineering
- Feature selection
- Model training
- Hyperparameter tuning
- Validation

Model Building

04

- Databricks SQL queries
- Tableau visualizations
- Statistical analysis

Analysis

About our project

- The intention of our project was to create a machine model that would be able to predict the happiness of countries around the world and explore different factors that impacted this variable.
- To achieve this, we created and ran two different models on various datasets with different parameters and analyzed the performance of the model using different statistical techniques.
- Our dataset has various variables, including Country, Region, Subregion (Location), Life Ladder, Log GDP per capita, Social support, Healthy life expectancy at birth, Freedom to make life choices, Generosity, Perceptions of corruption, Positive affect, Negative affect, Various Health indicators
- From this data, we developed various visualizations using Tableau and Google Colab that showcase how our model works and further support our conclusion.

Data Acquisition

1. **World Happiness Report 2024**
   Mainly gathers data from the Gallup World poll (GWP) 2005 - 2023
   Features:
   - Log GDP per capita ***(World Development Indicators 2023 and 2024 was forecasted)***
   - Social support
   - Healthy life expectancy at birth ***(WHO Global Health Observatory data last updated on 2020. Data is available for 200,2010, 2015 an 2019 for other periods interpolation and extrapolation are used)***
   - Freedom to make life choices
   - Generosity
   - Perceptions of corruption
   - Positive affect
   - Negative affect

   Target:
   - Life Ladder

Data Acquisition

2. WHO – World health statistics 2024

Monitoring health for the SDGs, Sustainable Development Goals since 2005

Features - 60 metrics related to:

- Mortality-related SDG indicators (Categories of causes of deaths, Maternal and child mortality, Mortality due to injury, Mortality due to chronic diseases, Mortality attributable to environmental risk factors)
- Health-related SDGs (Infectious diseases, Risk factors for health, Metabolic risk factors, Environmental risk factors, Risks to women’s and girls’ health
- Health systems strengthening (Service delivery, Health financing
- Progress towards WHO Triple Billion targets (One billion more people benefitting from UHC)

Target:

- Life expectancy
- Healthy life expectancy
- ***However, the target data wasn’t available for all years on the WHO dataset so we used the data on Healthy life expectancy at birth column in the World Happiness Report 2024***

Data Acquisition

3. **Other datasets**
    - Countries and regions
        - ISO classification alpha-2 / alpha-3
        - Country / region (5) / sub-region (17)
    - Sunshine hours for cities in the world
    - World cities
        - Lat and Lng for main cities around the world

ETL Process

World Happiness Report dataset

- Merge the world_happiness_df with the country_regions_df
- Confirm there aren't any NaN on the "Country name" to find countries with spelling mismatch
- Drop repeat country name column
- Reorganize the order of the columns => Geographic info, year, features, and target

```python
# Merge the world_happiness_df with the country_regions_df
world_happiness_ml_df = pd.merge(world_happiness_df, country_regions_df, left_on="Country name", right_on="name", how="left")
world_happiness_ml_df
```

✓ 0.0s Python

	Country name	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption	Positive affect	Negative affect	name	a
0	Afghanistan	2008	3.723590	7.350416	0.450662	50.500000	0.718114	0.164055	0.881686	0.414297	0.258195	Afghanistan	
1	Afghanistan	2009	4.401778	7.508646	0.552308	50.799999	0.678896	0.187297	0.850035	0.481421	0.237092	Afghanistan	
2	Afghanistan	2010	4.758381	7.613900	0.539075	51.099998	0.600127	0.117861	0.706766	0.516907	0.275324	Afghanistan	
3	Afghanistan	2011	3.831719	7.581259	0.521104	51.400002	0.495901	0.160098	0.731109	0.479835	0.267175	Afghanistan	
4	Afghanistan	2012	3.782938	7.660506	0.520637	51.700001	0.530935	0.234157	0.775620	0.613513	0.267919	Afghanistan	
...	...	...	...	...	...	...	...	...	...	...	...	...	

ETL Process

WHO dataset

- All the data was in rows, not columns. We had to pivot the final version of the df for the ***features to be columns***.
- Drop unneeded columns (Questions are codified in IND_CODE column).
  - IND_CODE = ["WHOSIS_0001", "WHOSIS_0002"] are the targets. (The Healthy Life Expectancy at Birth (years) and Life Expectancy at Birth (years))
    - The metrics are only available on the dataset for 2021.
- Merge the WHO df with the Country - Regions file.
  - Identify the region/sub-region NaN on the merge df and define if there is a way to work with those records.

	alpha-3	name	region	sub-region
1	AFR	NaN	NaN	NaN
53	EMR	NaN	NaN	NaN
62	EUR	NaN	NaN	NaN
71	GLOBAL	NaN	NaN	NaN
146	AMR	NaN	NaN	NaN
170	SEAR	NaN	NaN	NaN
199	WPR	NaN	NaN	NaN

AFR

AFR is a region, not a country => Algeria, Angola, Benin, Botswana, Burkina Faso, Burundi, Cameroon, Cape Verde, Central African Republic, Chad, Comoros, Ivory Coast, Democratic Republic of the Congo, Equatorial Guinea, Eritrea, Ethiopia, Gabon, Gambia, Ghana, Guinea, Guinea-Bissau, Kenya, Lesotho, Liberia, Madagascar, Malawi, Mali, Mauritania, Mauritius, Mozambique, Namibia, Niger, Nigeria, Republic of the Congo, Rwanda, São Tomé and Príncipe, Senegal, Seychelles, Sierra Leone, South Africa, South Sudan, Eswatini, Togo, Uganda, Tanzania, Zambia, Zimbabwe.

- Region: Africa
- Sub Region: Sub-Saharan Africa

EMR

Eastern Mediterranean Region does not have a single dedicated ISO code as it is a geographical region encompassing multiple countries, each with their own ISO code.

Afghanistan, Bahrain, Djibouti, Egypt, Iran, Iraq, Jordan, Kuwait, Lebanon, Libya, Morocco, Oman, Pakistan, Qatar, Saudi Arabia, Somalia, Sudan, Syria, Tunisia, United Arab Emirates, Yemen.

Afghanistan, Iran, Pakistan

- Region: Asia
- Sub Region: Southern Asia

Bahrain, Iraq, Jordan, Kuwait, Lebanon, Oman, Qatar, Saudi Arabia, Syrian Arab Republic, United Arab Emirates, Yemen

- Region: Asia
- Sub Region: Western Asia

Egypt, Libya, Morocco, Sudan, Tunisia

- Region: Africa
- Sub Region: Northern Africa

Djibouti, Somalia

- Region: Africa
- Sub Region: Eastern Africa

ETL Process

- ○ Filter by NaN on the "Country name" to find countries with spelling mismatch.
  - ■ Replace the country name on the world_happiness_df
  - ■ Add Kosovo and update region and sub-region for Taiwan in the country_regions_df
- We had to merge the Life Expectancy at Birth (years) column from the World Happiness data set to get a ***target column***.
  - ○ Count NaN on the target column "Healthy life expectancy at birth"
  - ○ **Unfortunately we will lost 32.3%** of the records for the ML model. (WHO special regions and target NaN)

```python
# Aggregate values by avg of DIM_1_CODE
who_aggregated_df = who_ml_df.groupby(["DIM_GEO_NAME", "DIM_GEO_CODE", "DIM_TIME_YEAR", "IND_CODE"], as
who_aggregated_df
```

✓ 0.0s Python

	DIM_GEO_NAME	DIM_GEO_CODE	DIM_TIME_YEAR	IND_CODE	VALUE_NUMERIC
0	Afghanistan	AFG	2016	SDGFPALL	42.099998
1	Afghanistan	AFG	2018	HWF_0006	4.520000
2	Afghanistan	AFG	2018	HWF_0014	0.292000
3	Afghanistan	AFG	2018	SDGIPV12M	35.000000
4	Afghanistan	AFG	2018	SDGIPVLT	46.000000
...	...	...	...	...	...

```python
# Pivot the who_aggregated_df DataFrame
who_pivot_df = who_aggregated_df.pivot_table(index=["DIM_GEO_NAME", "DIM_GEO_CODE", "DIM_TIME_YEAR"],
                                              columns="IND_CODE",
                                              values="VALUE_NUMERIC")
# Reset index
who_pivot_df.reset_index(inplace=True)

who_pivot_df.head()
```

✓ 0.0s Python

IND_CODE	DIM_GEO_NAME	DIM_GEO_CODE	DIM_TIME_YEAR	AMR_INFECT_ECOLI	AMR_INFECT_MRSA	FINPROT
0	Afghanistan	AFG	2016	NaN	NaN	
1	Afghanistan	AFG	2018	NaN	NaN	
2	Afghanistan	AFG	2019	NaN	NaN	
3	Afghanistan	AFG	2020	NaN	NaN	
4	Afghanistan	AFG	2021	NaN	NaN	

```python
# Count NaN on the target column "Healthy life expectancy at birth"
target_nan = who_country_nan_ml_df.loc[who_country_nan_ml_df["Healthy life expectancy at
print(f'Total amount of records: {who_country_nan_ml_df["DIM_GEO_NAME"].count()}')
print(f'Count target value NaN: {target_nan["DIM_GEO_NAME"].count()}')
```

✓ 0.0s Python

```
Total amount of records: 1369
Count target value NaN: 442
```

ETL Process

Sunshine Hours dataset

- Merge the sunshine_hours_df with the world_cities_df to get the latitude and longitude
- Filter by NaN on the "country" to find countries with spelling mismatch
  - Replace the country on the world_cities_df
  - Replace the city on the world_cities_df or the sunshine_hours_df

<u>All the datasets were exported to CSV files for ML modeling and tableau</u>

```python
# Drop "country" and "city_ascii" columns
tableau_mapping_df = tableau_mapping_df.drop(columns=["country", "city_ascii"])
tableau_mapping_df
```

Python

	Country	City	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year	lat	lng
0	Ivory Coast	Gagnoa	183.0	180.0	196.0	188.0	181.0	118.0	97.0	80.0	110.0	155.0	171.0	164.0	1823.0	6.1333	-5.9333
1	Ivory Coast	Bouaké	242.0	224.0	219.0	194.0	208.0	145.0	104.0	82.0	115.0	170.0	191.0	198.0	2092.0	7.6833	-5.0167
2	Ivory Coast	Abidjan	223.0	223.0	239.0	214.0	205.0	128.0	137.0	125.0	139.0	215.0	224.0	224.0	2296.0	5.3167	-4.0333
3	Ivory Coast	Odienné	242.0	220.2	217.3	214.7	248.8	221.8	183.5	174.5	185.4	235.8	252.0	242.6	2638.6	9.5000	-7.5667
5	Benin	Cotonou	213.9	210.0	223.2	219.0	213.9	141.0	136.4	148.8	165.0	207.7	243.0	223.2	2345.2	6.3667	2.4333
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Random Forest vs Gradient Boost Models

We did a regressor model for both random forest and gradient boost models because we were predicting numerical values.

Random Forest	Gradient Boost
How it works: ● Parallel training of independent trees ● Each tree is trained on a random subset of data and features ● Reduces overfitting through ensemble averaging When to use: ● Smaller datasets ● Need interpretability ● Want robust, generalized performance ● Less time for hyperparameter tuning	How it works: ● Sequential tree building ● Each tree corrects errors of previous trees ● Focuses on minimizing residual errors When to use: ● Large, structured datasets ● Able to invest in tuning ● Want maximum predictive performance ● Complex, non-linear relationships

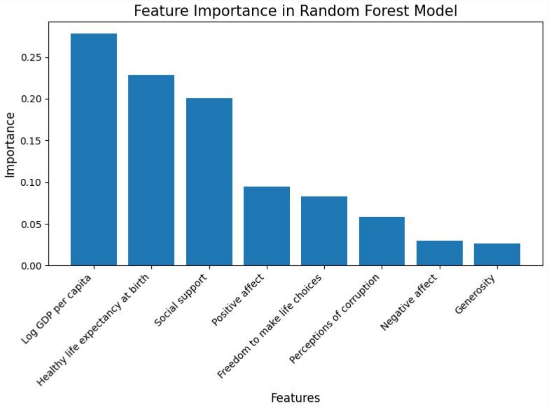
Happiness Data

Parameters:

```
rf_model = RandomForestRegressor(
  n_estimators= 500,
  max_depth= 10,
  min_samples_split= 10,
  max_features= 'sqrt',
  min_samples_leaf= 1,
  random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
Performance Comparison:
   Training R² Score: 0.9306
   Testing R² Score:  0.8400
```



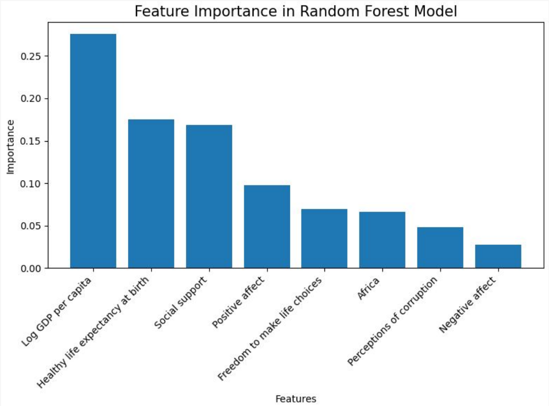
Happiness Data w/ Regions

Parameters:

```
rf_model = RandomForestRegressor(
  n_estimators= 500,
  max_depth= 10,
  min_samples_split= 10,
  max_features= 'sqrt',
  min_samples_leaf= 1,
  random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
Performance Comparison:
   Training R² Score: 0.9308
   Testing R² Score:  0.8760
```



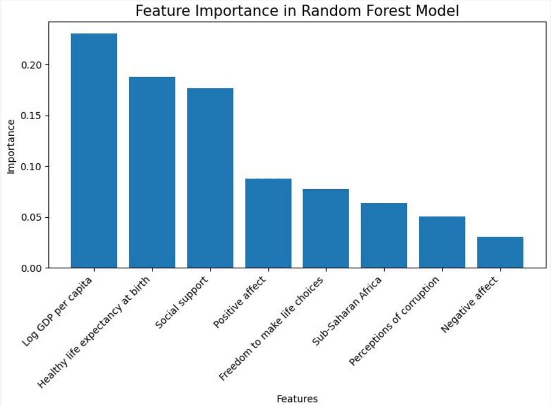
Happiness Data w/ Sub Regions

Parameters:

```
rf_model = RandomForestRegressor(
  n_estimators= 500,
  max_depth= 10,
  min_samples_split= 10,
  max_features= 'sqrt',
  min_samples_leaf= 1,
  random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
Performance Comparison:
   Training R² Score: 0.9231
   Testing R² Score:  0.8740
```



Random Forest Model

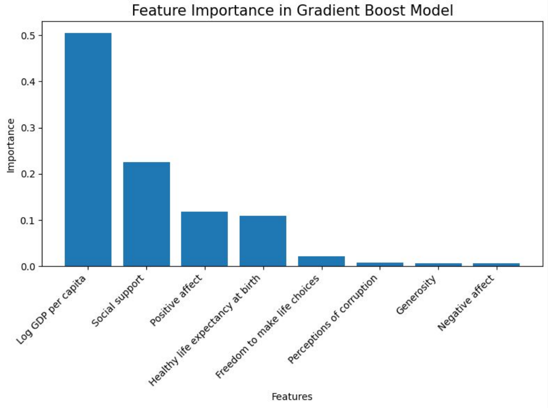
Happiness Data

Parameters:

```
gb_regressor =GradientBoostingRegressor(
    n_estimators= 300,
    learning_rate= 0.01,
    max_depth= 3,
    min_samples_split= 2,
    loss='squared_error' ,
    random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.8501
   Testing R² Score:  0.8285
```



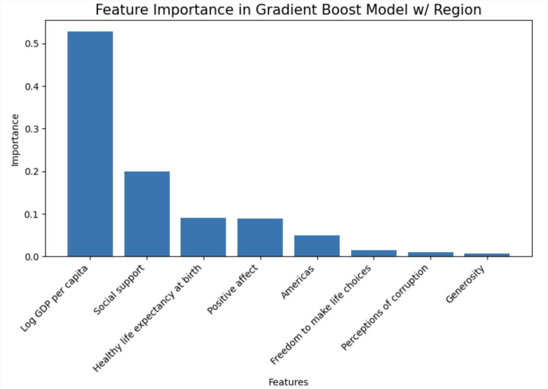
Happiness Data w/ Regions

Parameters:

```
gb_regressor1 =GradientBoostingRegressor(
    n_estimators= 300,
    learning_rate= 0.01,
    max_depth= 3,
    min_samples_split= 2,
    loss='squared_error' ,
    random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.8601
   Testing R² Score:  0.8224
```



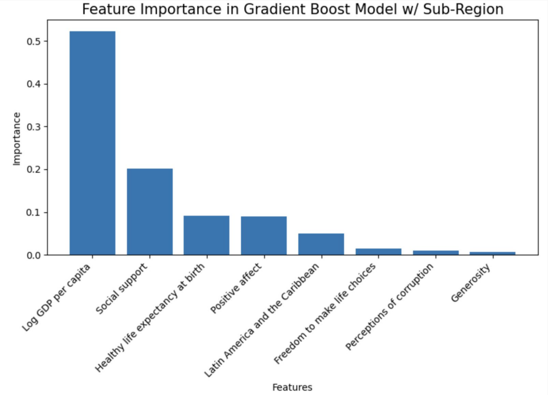
Happiness Data w/ Sub Regions

Parameters:

```
gb_regressor2
=GradientBoostingRegressor(
    n_estimators= 300,
    learning_rate= 0.01,
    max_depth= 3,
    min_samples_split= 2,
    loss='squared_error' ,
    random_state= 42
)
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.8620
   Testing R² Score:  0.8232
```



Gradient Boost Model

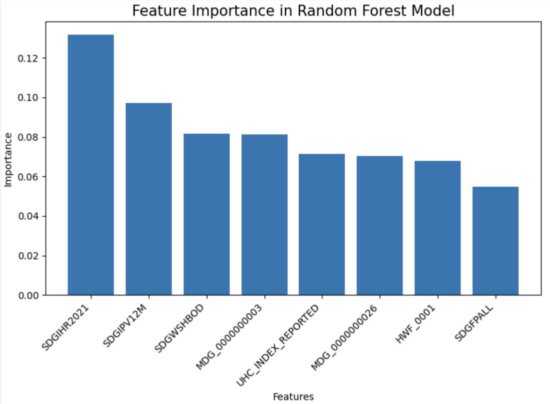
WHO Data

Parameters:

```
rf_regressor = RandomForestRegressor(
    n_estimators=500,
    random_state=78
    )
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.9232
   Testing R² Score:  0.5588
```



SDGIHR2021 - Average of 15 International Health Regulations core capacity scores
SDGIPV12M - Proportion of ever-partnered women and girls aged 15-49 years subjected to physical and/or sexual violence by a current or former intimate partner in the previous 12 months (%)

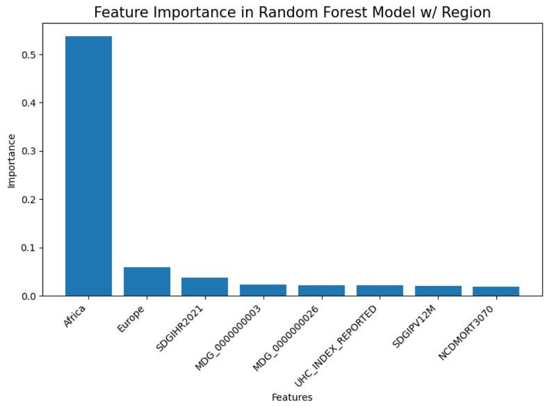
WHO Data w/ Regions

Parameters:

```
rf_regressor1 = RandomForestRegressor(
    n_estimators=500,
    random_state=78
    )
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.9400
   Testing R² Score:  0.7566
```



SDGIHR2021 - Average of 15 International Health Regulations core capacity scores
MDG_0000000003 - Adolescent birth rate (per 1000 women)

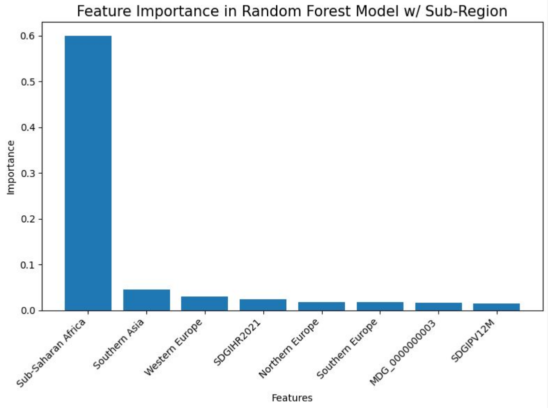
WHO Data w/ Sub Regions

Parameters:

```
rf_regressor2 = RandomForestRegressor(
    n_estimators=500,
    random_state=78
    )
```

Results:

```
Overfitting Diagnostics:
 Performance Comparison:
   Training R² Score: 0.9443
   Testing R² Score:  0.7706
```



SDGIHR2021 - Average of 15 International Health Regulations core capacity scores
MDG_0000000003 - Adolescent birth rate (per 1000 women)

Random Forest Model

ANOVA

Log GDP

ANOVA Results for Log GDP per capita:

F-statistic: 1246.9248

p-value: 0.0000

Statistically significant

Life Expectancy at Birth

ANOVA Results for Healthy life expectancy at birth:

F-statistic: 1013.0566

p-value: 0.0000

Statistically significant

Positive Affect

ANOVA Results for Positive affect:

F-statistic: 262.1806

p-value: 0.0000

Statistically significant

Freedom to make Life Choices

ANOVA Results for Freedom to make life choices:

F-statistic: 307.0536

p-value: 0.0000

Statistically significant

Social Support

ANOVA Results for Social support:

F-statistic: 874.1956

p-value: 0.0000

Statistically significant

Negative Affect

ANOVA Results for Negative affect:

F-statistic: 100.4659

p-value: 0.0000

Statistically significant

Generosity

ANOVA Results for Generosity:

F-statistic: 44.8425

p-value: 0.0000

Statistically significant

Perceptions of Corruption

ANOVA Results for Perceptions of corruption:

F-statistic: 134.8096

p-value: 0.0000

Statistically significant

SQL Queries

Top 10 happiest countries

```sql
%sql
WITH country_averages AS (
    SELECT `Country name`, ROUND(AVG(`Life Ladder`), 2) AS avg_life_ladder
    FROM happiness
    GROUP BY `Country name`
)
SELECT `Country name`, avg_life_ladder
FROM country_averages
ORDER BY avg_life_ladder DESC
LIMIT 10;
```

Table

	Country name	1.2 avg_life_ladder
1	Denmark	7.66
2	Finland	7.62
3	Iceland	7.47
4	Norway	7.46
5	Netherlands	7.44
6	Switzerland	7.44
7	Sweden	7.37
8	Canada	7.3
9	New Zealand	7.26
10	Australia	7.24

Top 10 countries with highest GDP

```sql
%sql
WITH country_averages AS (
    SELECT `Country name`, ROUND(AVG(`Log GDP per capita`), 2) AS avg_GDP
    FROM happiness
    GROUP BY `Country name`
)
SELECT `Country name`, avg_GDP
FROM country_averages
ORDER BY avg_GDP DESC
LIMIT 10;
```

Table

	Country name	1.2 avg_GDP
1	Luxembourg	11.64
2	Qatar	11.55
3	Singapore	11.37
4	Ireland	11.17
5	Switzerland	11.13
6	United Arab Emirates	11.12
7	Norway	11.07
8	United States	10.98
9	Kuwait	10.94
10	Hong Kong S.A.R. of Chi...	10.91

Top 10 countries with highest life expectancy

```sql
%sql
WITH country_averages AS (
    SELECT `Country name`, ROUND(AVG(`Healthy life expectancy at birth`), 2) AS avg_life_expectancy
    FROM happiness
    GROUP BY `Country name`
)
SELECT `Country name`, avg_life_expectancy
FROM country_averages
ORDER BY avg_life_expectancy DESC
LIMIT 10;
```

Table

	Country name	1.2 avg_life_expectancy
1	Japan	73.54
2	Singapore	72.92
3	Switzerland	72.17
4	South Korea	72
5	Israel	71.9
6	Iceland	71.87
7	Cyprus	71.75
8	France	71.64
9	Sweden	71.55
10	Spain	71.54


questions
Del
ScrLk
Ins
NumLk
Shift
PgUp
PgDn
End
Home