The ExPatApp

Where’s next?

Team

Allison, Jackie, Nicole, Ben, & Thomas

Overview

As we adjust to a post-pandemic world, the American people have changed how and where they work and live. As the American political landscape continues to evolve – and the rise of remote work allows workers to live wherever they find internet access – the idea of emigration from the US has renewed interest.

This analytical project aims to curate an index ranking of countries tailored to Americans interested in trying a new life in a foreign country.

Problems to solve

1. Economic parity: How stable and/or developed is the economy?
2. Health outcomes: Are the people who live there healthy?
3. Political system: Is it democratic? Are the people ‘free’?
4. Education system: What are the average person’s schooling outcomes?
5. Culture: Would an American be welcome there? How happy are the people who live there?

Data Exploration for Country-Level data

- Economy
  - Potential sources: United Nations, World Bank, Newspapers, Academic studies
  - Metrics: UN HDI, GDP/GNI, Income inequality (Gini), CoL, Internet speed, Big Mac Index
- Health
  - Potential sources: UN, World Bank, WHO, OECD, NGOs
  - Metrics: Life expectancy, Happiness index, Quality of life
- Politics
  - Potential sources: UN, World Bank, OECD, The Economist, Freedom House, NGOs
  - Metrics: Human Freedom Index, Democracy Index, Global Freedom Scores
- Education
  - Sources: UN, World Bank, OECD, CIA World Factbook
  - Metrics: Literacy rates, UN HDI, mean/expected years of schooling, prevalence of advanced degrees
- Lifestyle
  - Sources: UN, World Bank, Statista, UNESCO
  - Metrics: Religious freedom index, language, population diversity, climate, Proportion of English-speakers, Climate

The Solution

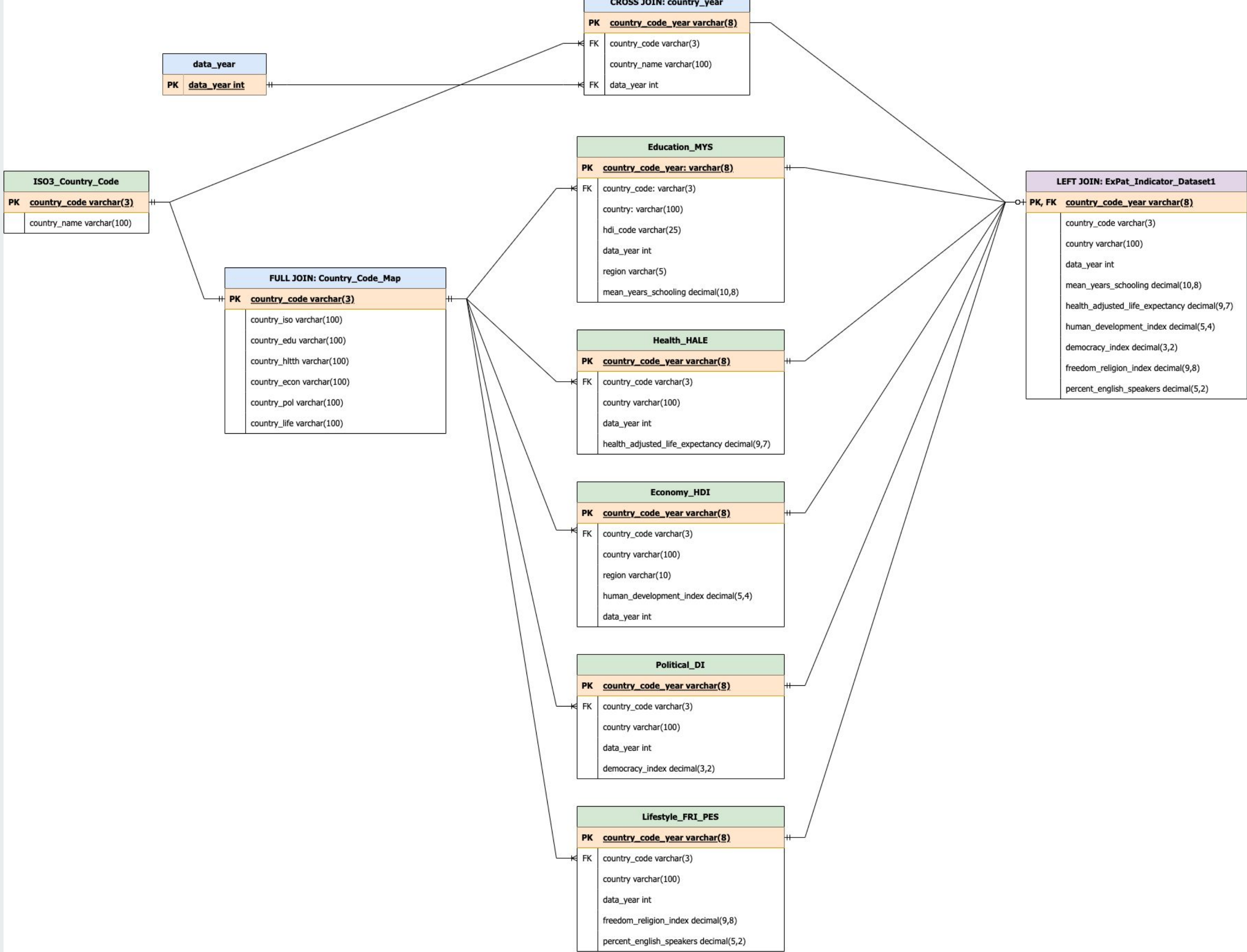
1. Economy: UN Human Development Index scores, incl. GNI per capita
2. Health outcomes: Health Adjusted Life Expectancy (HALE), World Bank/UN
3. Political system: *The Economist's* Democracy Index
4. Education: Literacy rates and mean years of schooling (World Bank)
5. Culture: Freedom of Religion data (UN)

Methodology

Tools

- Google Drive
- GitHub
- Postgres
  - pgAdmin
- Python, Pandas
  - SQLAlchemy
  - scikit-learn
- Tableau Public

SQL ERD


CROSS JOIN: country_year
PK country_code_year varchar(8)
FK country_code varchar(3)
country_name varchar(100)
FK data_year int
data_year
PK data_year int
ISO3_Country_Code
PK country_code varchar(3)
country_name varchar(100)
FULL JOIN: Country_Code_Map
PK country_code varchar(3)
country_iso varchar(100)
country_edu varchar(100)
country_hltth varchar(100)
country_econ varchar(100)
country_pol varchar(100)
country_life varchar(100)
Education_MYS
PK country_code_year: varchar(8)
FK country_code: varchar(3)
country: varchar(100)
hdi_code varchar(25)
data_year int
region varchar(5)
mean_years_schooling decimal(10,8)
Health_HALE
PK country_code_year varchar(8)
FK country_code varchar(3)
country varchar(100)
data_year int
health_adjusted_life_expectancy decimal(9,7)
Economy_HDI
PK country_code_year varchar(8)
FK country_code varchar(3)
country varchar(100)
region varchar(10)
human_development_index decimal(5,4)
data_year int
Political_DI
PK country_code_year varchar(8)
FK country_code varchar(3)
country varchar(100)
data_year int
democracy_index decimal(3,2)
Lifestyle_FRI_PES
PK country_code_year varchar(8)
FK country_code varchar(3)
country varchar(100)
data_year int
freedom_religion_index decimal(9,8)
percent_english_speakers decimal(5,2)
LEFT JOIN: ExPat_Indicator_Dataset1
PK, FK country_code_year varchar(8)
country_code varchar(3)
country varchar(100)
data_year int
mean_years_schooling decimal(10,8)
health_adjusted_life_expectancy decimal(9,7)
human_development_index decimal(5,4)
democracy_index decimal(3,2)
freedom_religion_index decimal(9,8)
percent_english_speakers decimal(5,2)

ExPat Database in SQL

- There are seven static tables in the database. These tables were created using the Create_Schema SQL script and importing the following CSV tables:
  - Economy_HDI
  - Education_MYS
  - Health_HALE
  - Lifestyle_FRI_PES
  - Poltical_DI
  - ISO3_Codes: ISO 3166-1 alpha-3 codes are three-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest.
  - Data_Year: We focused on data from years 2000 - 2022.
- In order to join all the source datasets together, we created a country code map table by performing full joins with the country name in the ISO3_codes table and the country names of the source datasets (see next slide).
  - Manual updates were also made to an exported copy of the country code map table to ensure each country name in all 5 source tables had a corresponding ISO3 code.

SQL code to create country code map table

```sql
-- create country code map since not every country is spelled the same in every source dataset
select distinct a.country_code,
       a.country country_iso,
       b.country country_edu,
       c.country country_hlth,
       d.country country_econ,
       e.country country_pol,
       f.country country_life
 into country_code_map
 from iso3_country_codes a
    full join education_mys b on a.country = b.country
    full join health_hale c on a.country = c.country
    full join economy_hdi d on a.country = d.country
    full join political_di e on a.country = e.country
    full join lifestyle_fri_pes f on a.country = f.country;
```

SQL code to create ExPat Indicator Dataset

- ➔ Using a *WITH* query, we created a common table expression (CTE) named country_year.
  - ◆ This table represents all the distinct combinations of country code/name and data year by performing a *cross join* between the ISO3_codes table and data_year table (see figure).
- ➔ By performing *left joins* between the country_code_year column of the country_year (i.e, CTE query explained above) and country_code_year columns of all the source data tables, created the merged dataset to be used for our machine learning model (see figure).

```
with country_year as (
    select a.country_code,
           a.country_iso country,
           b.data_year,
           concat(a.country_code, '_', b.data_year) country_code_year
    from country_code_map a cross join data_year b
)
select a.country_code_year,
       a.country_code,
       a.country,
       a.data_year,
       b.mean_years_schooling,
       c.health_adjusted_life_expectancy,
       d.human_development_index,
       e.democracy_index,
       f.freedom_religion_index,
       f.percent_english_speakers
into expat_indicator_dataset1
from country_year a
    left join education_mys b on a.country_code_year = b.country_code_year
    left join health_hale c on a.country_code_year = c.country_code_year
    left join economy_hdi d on a.country_code_year = d.country_code_year
    left join political_di e on a.country_code_year = e.country_code_year
    left join lifestyle_fri_pes f on a.country_code_year = f.country_code_year;
```

Database interface

The project database interfaces with the project by using the merged source data (i.e., ExPat Indicator Dataset) as the input data for the machine learning model. A connection string via the psycopg2-binary package can potentially be used to connect PostgresSQL and Python (see figure below). For testing purposes, however, we are currently importing the CSV version of the dataset into Python for ease of use.

```
# A cell for importing the data. The commented lines are how we would do this with SQL, but for now we are using a static file to test our algorithm.
#
# !pip install psycopg2-binary
# db_string = f"postgresql://postgres:{db_password}@127.0.0.1:5432/expat_data"
# engine = create_engine(db_string)
#
# If we were really doing this with SQL, we would also put our data table into SQL with a line like the following:
#
# df_expat.to_sql(name='expat', con=engine)
#
# ...but because we are not using the actual SQL database at this point, that line doesn't appear in the following code.
#
# Import static .csv file which was exported from our database
df_expat = pd.read_csv("https://drive.google.com/uc?export=download&id=1A6xzq-o2HFz83j8deFjX3fIdnEwGTvU1")
df_expat
```

	country_code_year	country_code	country	data_year	mean_years_schooling	health_adjusted_life_expectancy	human_development_index	democracy_index	freedom_religion_index	percent_english_speakers
0	ABW_2000	ABW	Aruba	2000	NaN	NaN	NaN	NaN	NaN	NaN
1	AFG_2000	AFG	Afghanistan	2000	NaN	46.622245	NaN	NaN	NaN	NaN
2	AGO_2000	AGO	Angola	2000	NaN	46.013173	NaN	NaN	NaN	NaN
3	AIA_2000	AIA	Anguilla	2000	NaN	NaN	NaN	NaN	NaN	NaN
4	ALA_2000	ALA	Åland Islands	2000	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
5745	WSM_2022	WSM	Samoa	2022	NaN	NaN	NaN	NaN	NaN	NaN
5746	YEM_2022	YEM	Yemen	2022	NaN	NaN	NaN	NaN	NaN	NaN
5747	ZAF_2022	ZAF	South Africa	2022	NaN	NaN	NaN	NaN	NaN	NaN
5748	ZMB_2022	ZMB	Zambia	2022	NaN	NaN	NaN	NaN	NaN	NaN
5749	ZWE_2022	ZWE	Zimbabwe	2022	NaN	NaN	NaN	NaN	NaN	NaN

5750 rows × 10 columns

Machine Learning Model

- ➔ To prepare the data for a ML algorithm, dropped all unnecessary columns for analysis (including year and country name); after this was done,
- ➔ Used the raw data to compile a DataFrame the rows of which were the most current index measures available for a specific country, so that the data would be most relevant to an expat moving to that country in 2022.
- ➔ Once the latest data was collected, we proceeded to rescale our numerical indices for PCA analysis.
- ➔ Feature engineering was fairly minimal; needed to drop columns variation across which would not support our clusters, like year.
- ➔ Choosing to create "latest" country profiles for each country out of some data that might be out of date inspired us to create a "fudge factor" parameter
    - ◆ Any time old data must be substituted for new, up-to-date data, the fudge factor counter becomes a more negative number;
    - ◆ This captured the spirit of our algorithm because positive variation in the fudge factor tracks a positive/desirable feature of a country for expats: namely, that up-to-date data is available for that country.

Machine Learning Model

```
df_expat
```

	country_code	country	data_year	human_development_index	health_adjusted_life_expectancy	mean_years_schooling	freedom_religion_index
0	AFG	Afghanistan	2019	0.511	54.111275	3.930000	0.273744
1	AGO	Angola	2019	0.581	56.745929	5.173993	0.455960
2	ALB	Albania	2019	0.795	68.859483	10.145730	0.684292
3	ARE	United Arab Emirates	2019	0.890	64.379104	12.111220	0.350558
4	ARG	Argentina	2019*	0.845	66.791514	10.940601	0.790884
...	...	...	...	...	...	...	...
157	VNM	Viet Nam	2019	0.704	65.741530	8.320000	0.273744
158	YEM	Yemen	2019*	0.470	58.586660	3.200000	0.138944
159	ZAF	South Africa	2019*	0.709	56.177157	10.240646	0.605027
160	ZMB	Zambia	2019*	0.584	55.081757	7.152016	0.516698
161	ZWE	Zimbabwe	2019*	0.571	53.557018	8.466800	0.486934

Machine Learning Model

- ➔ Because we employed an unsupervised ML model, training and testing sets were not necessary for us.
- ➔ Our choice of an unsupervised machine learning model for our project has one clear downside, which is that it is difficult to ascertain the "accuracy" of our suggestions for users; without supervised learning (aka a verifiable outcome, training & testing sets, etc.) it is difficult to verify our cluster output.
- ➔ There are, however, helpful benefits to this unsupervised approach: countries that are surprisingly similar to the US can be revealed without preconception.

- ➔ Since our ML algorithm uses hierarchical clustering rather than K-Means, it doesn't depend on a random seed, which seems appropriate for a big decision like which country to move to.
- ➔ *Note: For this latest analysis we dropped the column related to "percent of English speakers" in a country, because the data was missing information from so many countries.*
    - ◆ *Our group agreed, however, that this is an important data point to consider for expats, and we found that there is better and more up-to-date data available for this measure using the CIA World Factbook; we will add this back in in future analysis.*

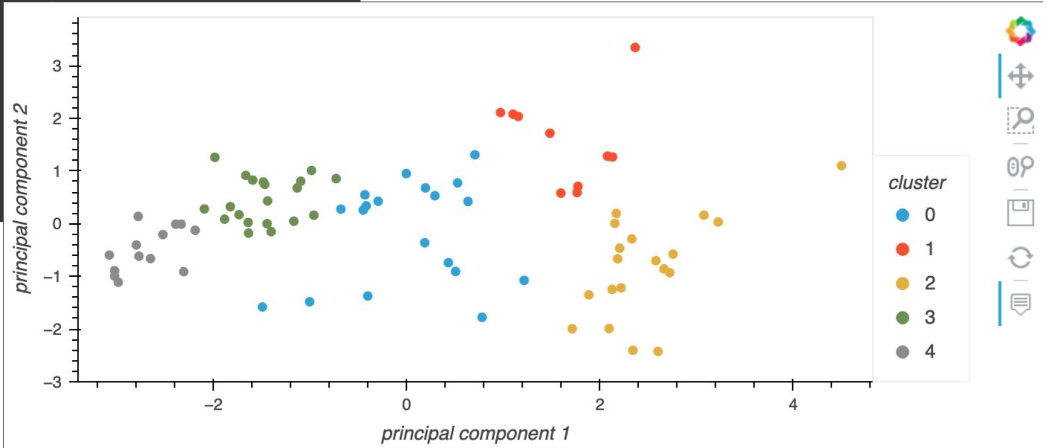
Machine Learning Model

```
US_index = df_expat_nonull[df_expat_nonull['country_code']=='USA'].index.values.astype(int)[0]
US_cluster_label = df_expat_nonull.at[US_index,'cluster']
USlike_cluster = []

for index, row in df_expat_nonull.iterrows():
  if row['cluster'] == US_cluster_label:
    USlike_cluster.append(row['country'])

USlike_cluster
```

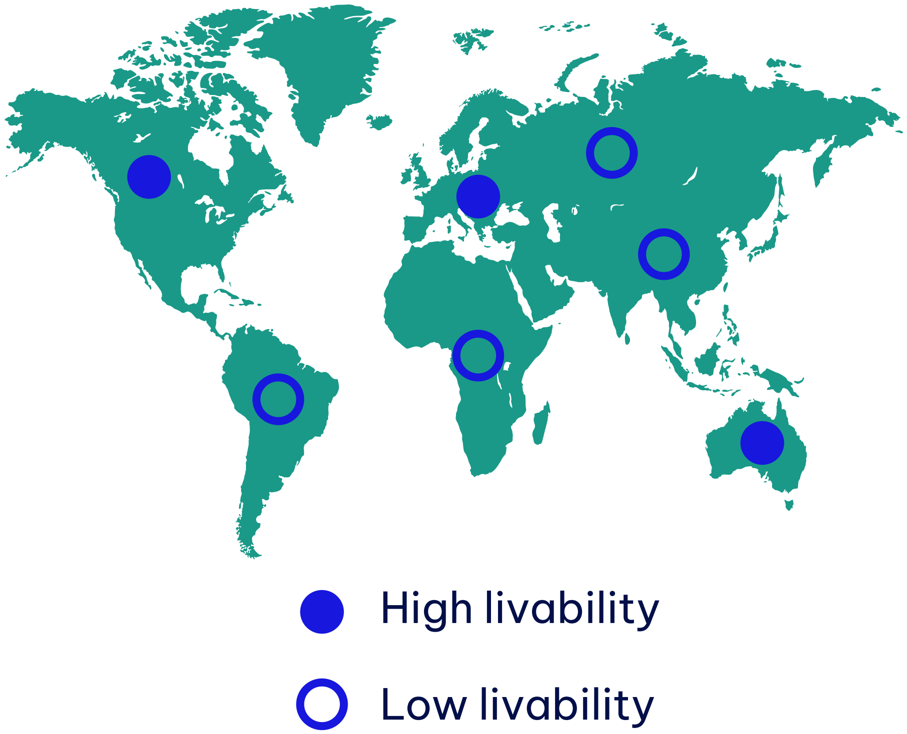
```
['Australia',
 'Canada',
 'Switzerland',
 'Denmark',
 'United Kingdom',
 'Ireland',
 'Israel',
 'Luxembourg',
 'Netherlands',
 'Norway',
 'New Zealand',
 'Slovenia',
 'United States of America']
```

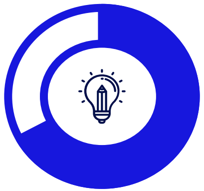


Dashboard

ExPatDash

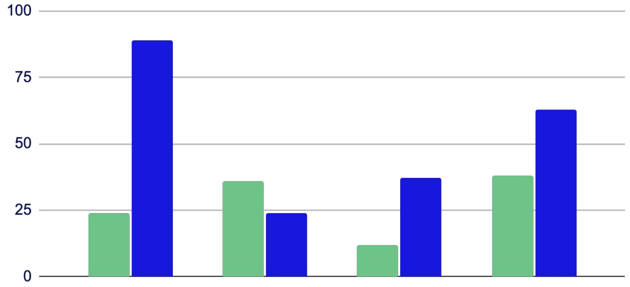
- Using Tableau Public, the project dashboard will enable the user to dig down into some nitty-gritty comparisons between countries in the cluster the ML algorithm returns.
- The features available for data visualization in Tableau may include:
  - Map overlays highlighting the countries most suited for emigration from America
  - Graphs highlighting the specific metrics behind the indicators for top-ranked countries
  - Filters that enable the user to choose specific factors that affect the rankings
  - Charts allowing them to compare 2 or more countries' data

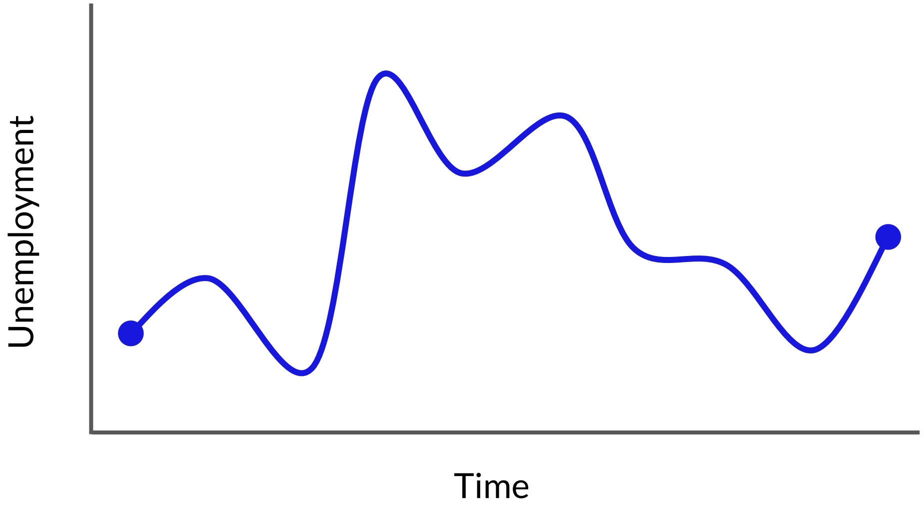




Economic Development	Education System
70%	50%
High GNI per capita	Mid education outcomes

ANALYSIS





Mock-up of features for Tableau dashboard