PHASE 3 PROJECT

LESLEY KAMAMO

Overview

Stepping into Billionaire Territory

This project requires the use Classification Model to generate insights for a given Agency.

This agency provides all sorts of social support in the different industries, mostly volunteer work. In a bid to increase their profits for the next financial year, their have been looking through different datasets that will enable them make a choice of industriy investment.

Business Understanding

BUSINESS PROBLEM

BUSINESS OBJECTIVES

The objectives of this project based on the dataset chosen is to find out:

1. The Wealth Status of Billionaires (whether self-made or not)

2. What industry/sources is more inclined to produce billionaires?

3. Demographic analysis of billionaires (age, gender, country)

4. Provide classification to the wealth status of billionaires based on the features.

5. Provide insights into which industry are likely to produce billionaires in future (logistic regression)

Data Understanding

The dataset represents historical data on billionaires for recent past years, hence this data will be modified for the purpose of the analysis

The data is contained in a CSV file:

1. `Billionaires Statistics Dataset.csv`: each record represents rank, finalWorth, category, personName, age, country,source, selfMade,status,gender,birthDate,title,residenceStateRegion,birthYear, tax_revenue_country_country,total_tax_rate_country,population_country among other fields.

Data Preparation

1. Loading the Dataset
2. Handling Missing Values
3. Describing the Data

Load the Dataset

Open the csv file as a Dataframe

```
# load the dataset as `billionaire_df`
billionaire_df = pd.read_csv("data/Billionaires Statistics Dataset.csv", index_col=0)

billionaire_df
```

[372]

Handling Missing Values

```
# check for missing values
billionaire_df.isnull().sum()
```

```
billionaire_df.describe()
```

[14] ✓ 0.0s

	finalWorth	age	birthYear	birthMonth	bi

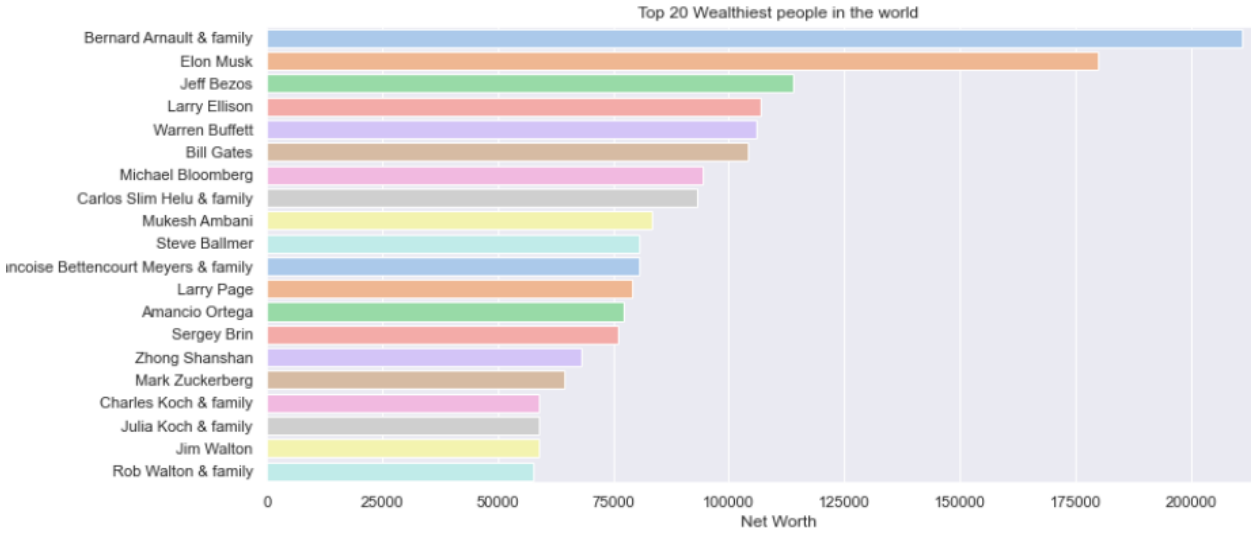
```
# drop the irrelevant columns
billionaire_df = billionaire_df.drop(columns=['category'])

# keep only columns with no missing values
billionaire_df = billionaire_df.dropna(axis=1)
```

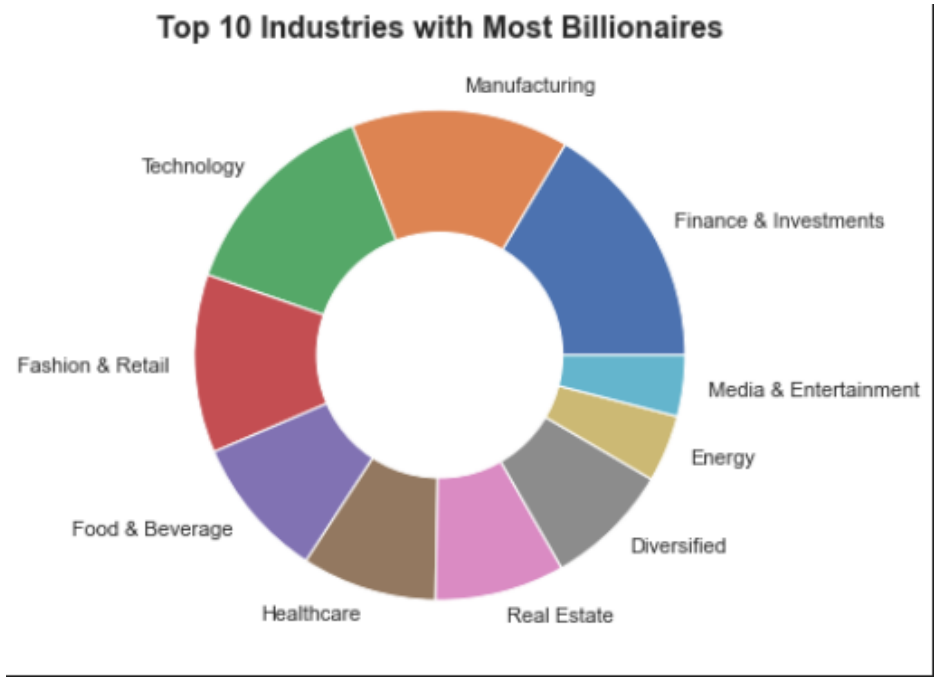
✓ 0.0s

Exploratory Data Analysis

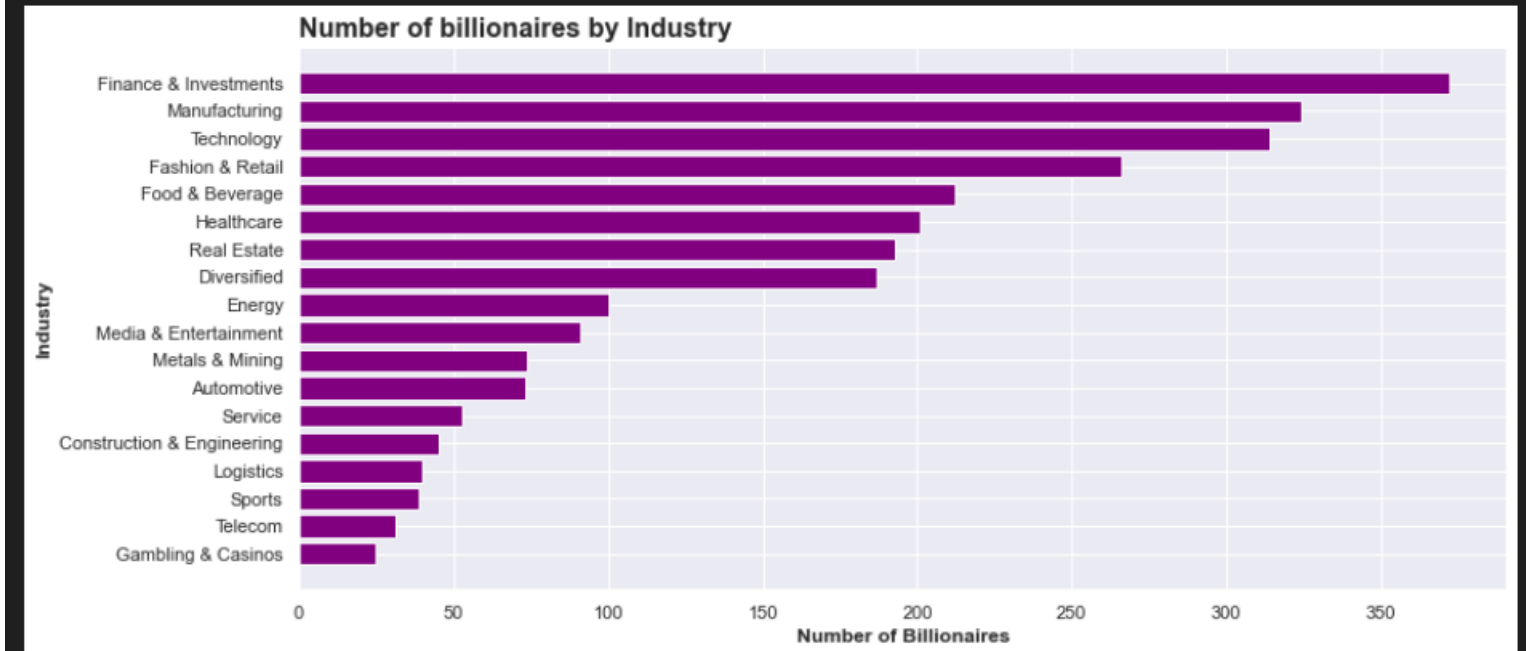
1. Univariate Analysis
2. Bivariate Analysis

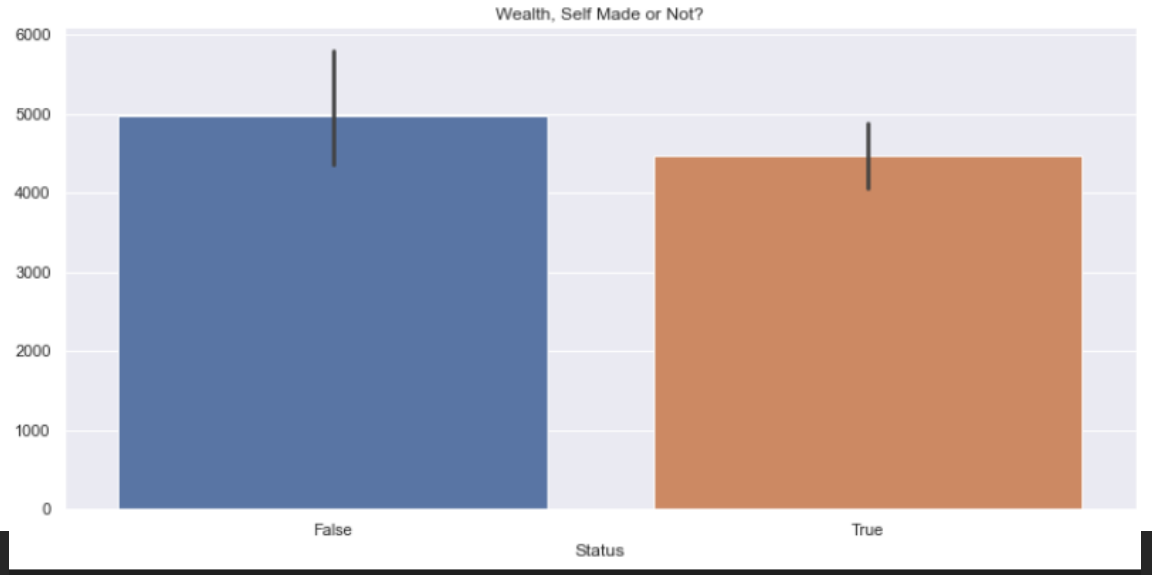
```
(0.5, 1.0, 'Top 20 Wealthiest people in the world'),
(0.5, 0, 'Net Worth'),
(0, 0.5, 'Person')]
```

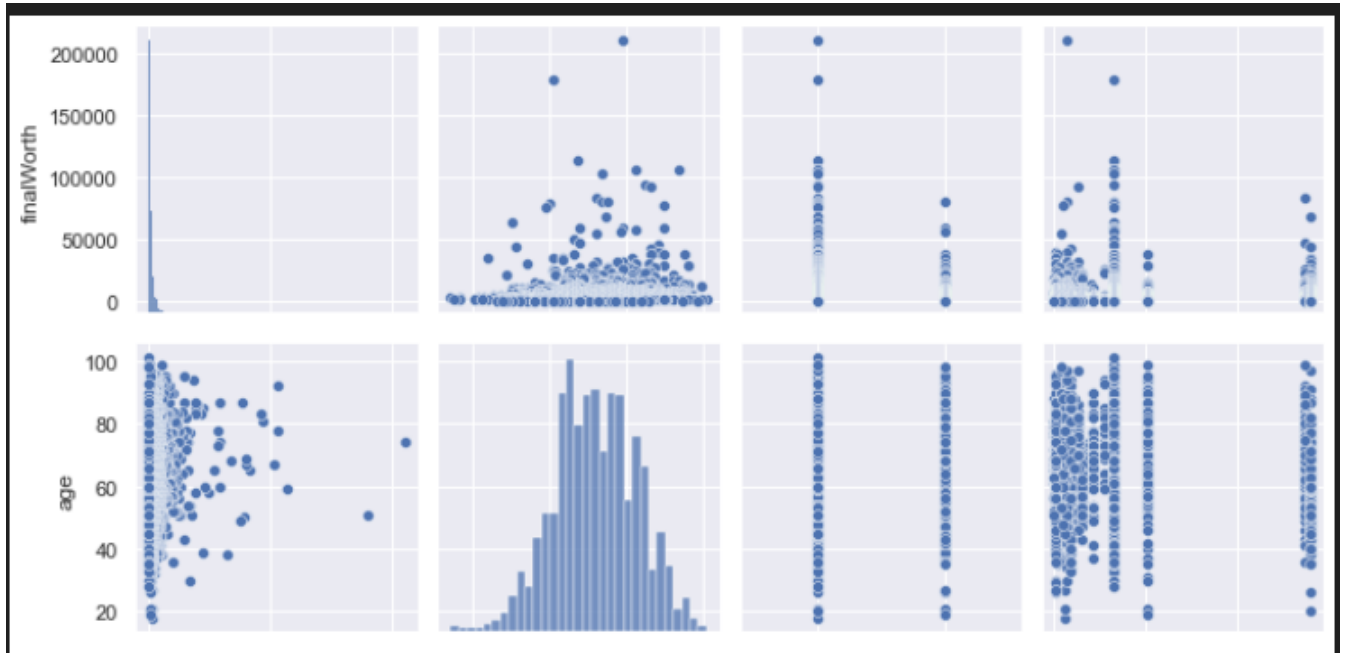


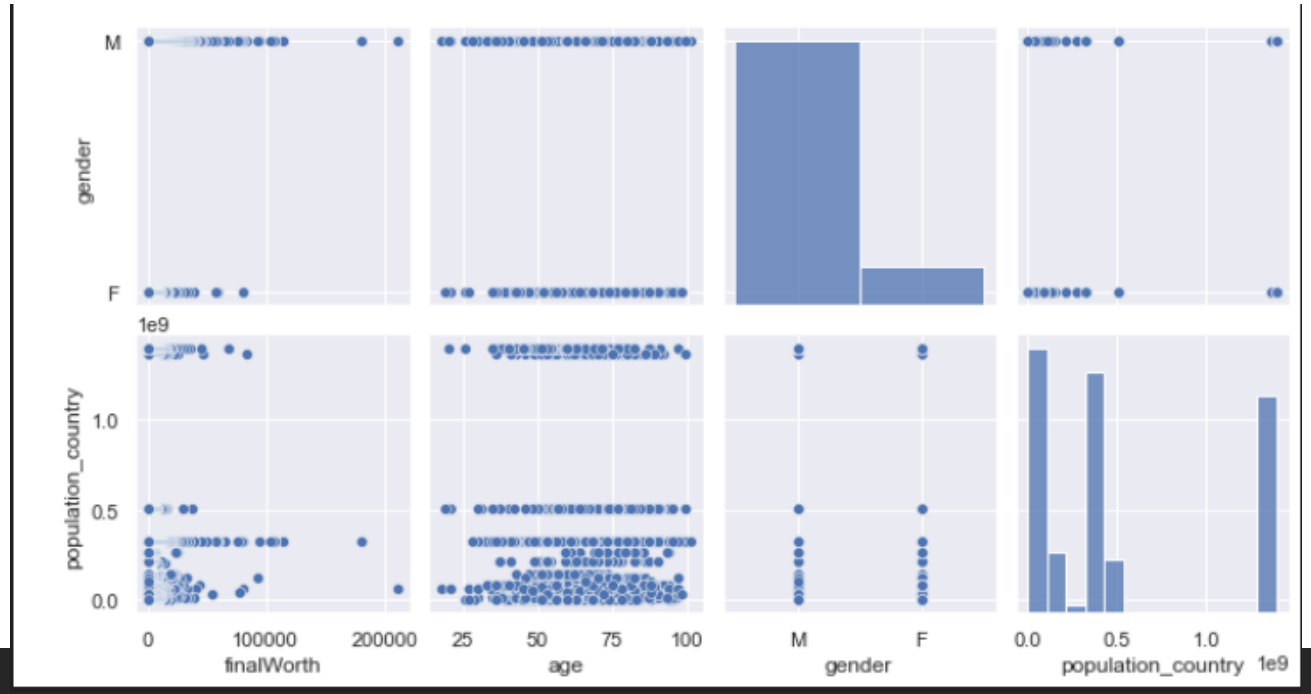



```
Text(0, 0.5, 'Industry')
```

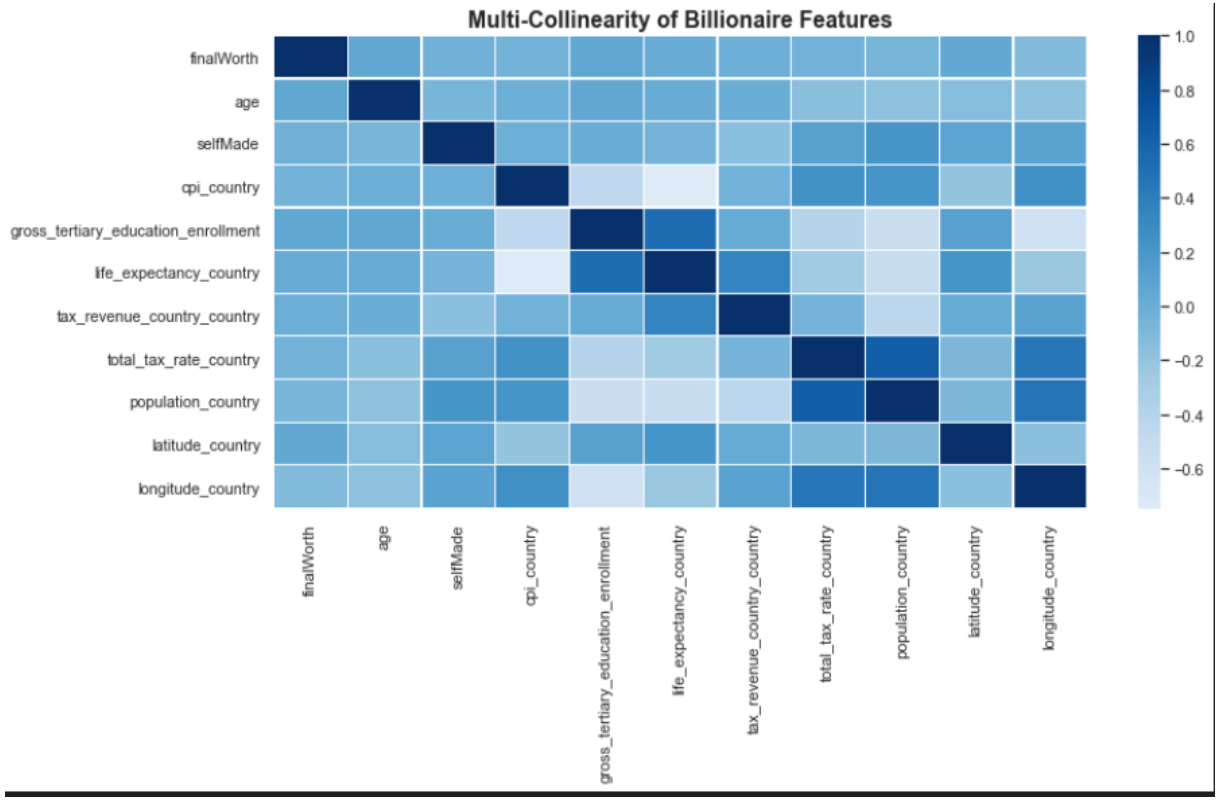


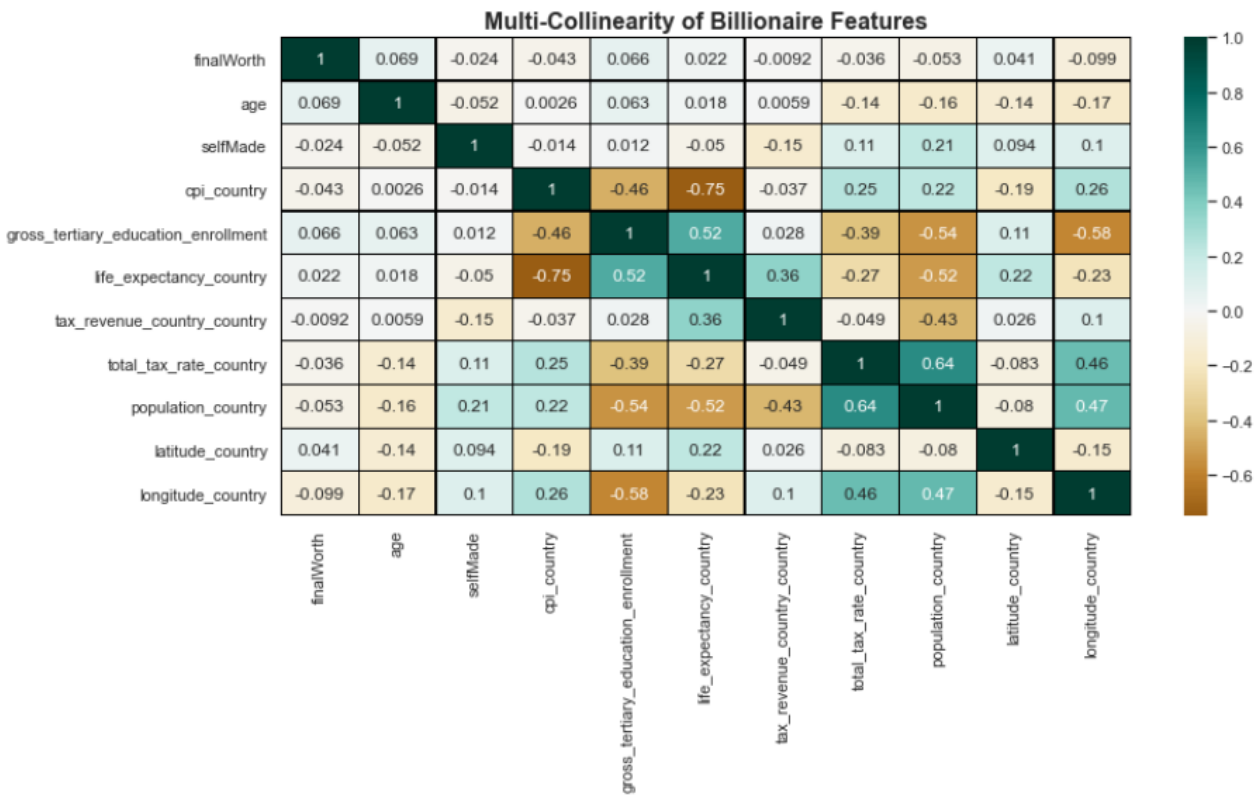



finalWorth
200000
150000
100000
50000
0
age
100
80
60
40
20


gender
M
F
1e9
population_country
1.0
0.5
0.0
0
100000
200000
finalWorth
25
50
75
100
age
M
F
gender
0.0
0.5
1.0
1e9
population_country

Multi-Collinearity





Modelling

Modelling

1. Data Splitting
2. Preprocess the Data
3. Model Training
4. Model Fitting
5. Model Evaluation

Logistic Regression

```
from sklearn.model_selection import train_test_split

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

✓ 0.7s

```
# convert the categorial variable to numeric values

from sklearn.preprocessing import OneHotEncoder
# one-hot encoder
features_encoder = OneHotEncoder(handle_unknown='ignore')

X_train_encoded = features_encoder.fit_transform(X)
X_test_encoded = features_encoder.fit_transform(X)
```

✓ 0.0s

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(max_iter=5000, multi_class='multinomial', solver='lbfgs')

#Train the model
model.fit(X_train_encoded, y_train_encoded)
```

✓ 0.4s

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Make predictions
y_pred = model.predict(X_test_encoded)

# Evaluate the model
accuracy = accuracy_score(y_test_encoded, y_pred)
precision = precision_score(y_test_encoded, y_pred, average='weighted')
recall = recall_score(y_test_encoded, y_pred, average='weighted')
f1 = f1_score(y_test_encoded, y_pred, average='weighted')

print({"Accuracy": accuracy,"Precision": precision,"Recall": recall,"f1_score": f1})
```

Modelling

1. Data Splitting
2. Preprocess the Data
3. Model Training
4. Model Fitting
5. Model Evaluation

Decision Tree Classifier

```
# 1. Create a decision tree classifier

from sklearn.tree import DecisionTreeClassifier

# Initialized DecisionTree
dt_classifier  = DecisionTreeClassifier(max_depth=3, min_samples_split=100, random_state = 42)

# 2. Train a decision tree classifier
dt_classifier.fit(X_train_encoded, y_train_encoded)

# 3. Make predictions
yd_pred = dt_classifier.predict(X_test_encoded)
```

✓ 0.2s

```
from sklearn.metrics import accuracy_score
dt_accuracy = accuracy_score(y_test_encoded, yd_pred)

print({'Accuracy': dt_accuracy})
```

✓ 0.0s

Findings

1. From The age distribution analysis, Most Billionaires in the world lie in the age group bracket of 50 -60 years
2. The top five countries of citizenship for most billionaires include : United States, China, India, Germany, Russia
3. The top industry with most number of billionaires in the world are: Finance and Investments, Manufacturing, Technology, Fashion & Retail, Food & Beverage
4. The major sources of wealth include: Investments, Real Estate, Software, Pharmaceuticals
5. On the question on whether Most billionaire acquired their wealth themselves or Inherited, we have seen that most of them have inherited

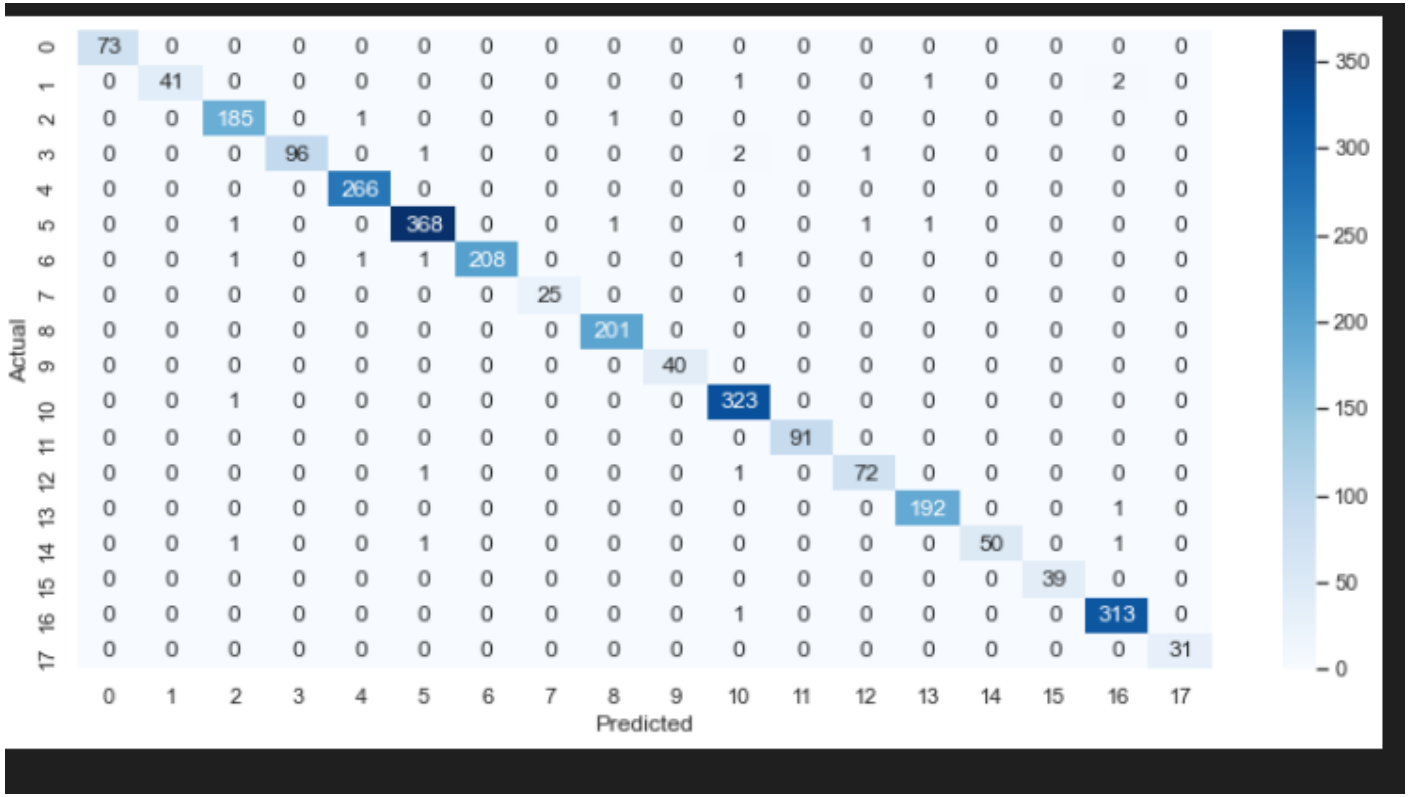
Recommendations

I would recommend investing in the Finance & Investments, Technolgy industries, as it shows potential of growing, not only for the younger generation but also the aging.

Conclusion

From the project above, we are able to see that the industry with a high number of wealth distributed is the Finance & Investments industry. Hence we are able to conclude that, based on the wealth age distribution, wealth is accumulated over the years, having best invested in the right industry.

The various logistic modelling, present an almost perfect accuracy score based on the variables indicated. The technologies used for this project include: Data Preparation, Exploratory Data Analysis methods, Data Modelling



THANK YOU

20TH OCTOBER 2023