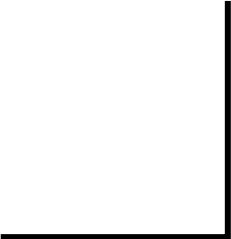




The X-Files Problem

By Oscar Aguilar
September 2020



Index

- I. Introduction to the problem
- II. Data Preprocessing
- III. Descriptive statistics
- IV. Visualization and Findings
- V. Conclusion and next steps

Introduction and Problem

Society has related UFO culture to a matter of belief; however, there is plenty of data that registers UFO sightings. Therefore, due to these empirical observations, UFO sightings and research should be handled as any scientific problem. Gallup, a consulting company, performed a poll in 2019 at the U.S., they found that 33% of Americans believed that the UFO sightings recorded in their country were indeed alien-related; nonetheless, 60% of Americans showed skepticism, and 7% were not sure. Out of the total poll-participants, 16% stated that they have personally witnessed a UFO visit.



This problem consists on evaluating how rare UFO sightings are. In order to do so, data collected by the company Infinito is going to be used. The final scope of this project is to find where in the globe is there a higher chance to live an UFO experience.

Data Preprocessing

Data shape: 80,332 rows, 11 columns

One column has observations in hours/min and another one in seconds; these values are equivalent between each other e.g. 1 min & 60 seconds. Which is more consistent?

#Variable types
df.dtypes

datetime	object
city	object
state	object
country	object
shape	object
duration (seconds)	object
duration (hours/min)	object
comments	object
date posted	object
latitude	object
longitude	float64
dtype: object	

#Null Values
df.isnull().sum()

datetime	0
city	0
state	5797
country	9670
shape	1932
duration (seconds)	0
duration (hours/min)	0
comments	15
date posted	0
latitude	0
longitude	0
dtype: int64	

A

Numerical attributes as objects, e.g. latitude and longitude

B

NaN values: missing data

```
print('Unique value in hours/min')
print(len(df['duration (hours/min)'].unique().tolist()))
print('Unique value in seconds')
print(len(df['duration (seconds)'].unique().tolist()))
```

Unique value in hours/min

8349

Unique value in seconds

706

"In the case of multivariate analysis, if there is a larger number of missing values (25-30%), then it can be better to drop those cases (rather than do imputation) and replace them."

```
#Drop missing values, and redundant column
df.dropna(inplace=True)
df.drop(['duration (hours/min)'], axis=1, inplace=True)
```

df.isnull().sum()

datetime	0
city	0
state	0
country	0
shape	0
duration (seconds)	0
comments	0
date posted	0
latitude	0
longitude	0
dtype: int64	

```
#Change of variable types
df['duration (seconds)'] = df['duration (seconds)'].astype(float)
df['latitude'] = df['latitude'].astype(float)
df['longitude'] = df['longitude'].astype(float)
df.dtypes
```

datetime	object
city	object
state	object
country	object
shape	object
duration (seconds)	float64
comments	object
date posted	object
latitude	float64
longitude	float64
dtype: object	

B

A

Data shape: 66,516 rows, 10 columns

Data Preprocessing

C

In order to deal with date variables (date time and date posted), both columns were handled in order to have DD, MM and YYYY in different columns (For both variables).

	City	State	Country	Shape	Duration (s)	Comments	Latitude	Longitude	Month	Day	Year	Month_posted	Day_posted	Year_posted
0	San marcos	Tx	Us	Cylinder	2700.0	This event took place in early fall around 194...	29.883056	-97.941111	10	10	1949	4	27	2004
1	Edna	Tx	Us	Circle	20.0	My older brother and twin sister were leaving ...	28.978333	-96.645833	10	10	1956	1	17	2004
2	Kaneohe	Hi	Us	Light	900.0	As a marine 1st lt. flying an fj4b fighter/att...	21.418056	-157.803611	10	10	1960	1	22	2004
3	Bristol	Tn	Us	Sphere	300.0	My father is now 89 my brother 52 the girl wit...	36.595000	-82.188889	10	10	1961	4	27	2007
4	Norwalk	Ct	Us	Disk	1200.0	A bright orange color changing to reddish colo...	41.117500	-73.408333	10	10	1965	10	2	1999
...

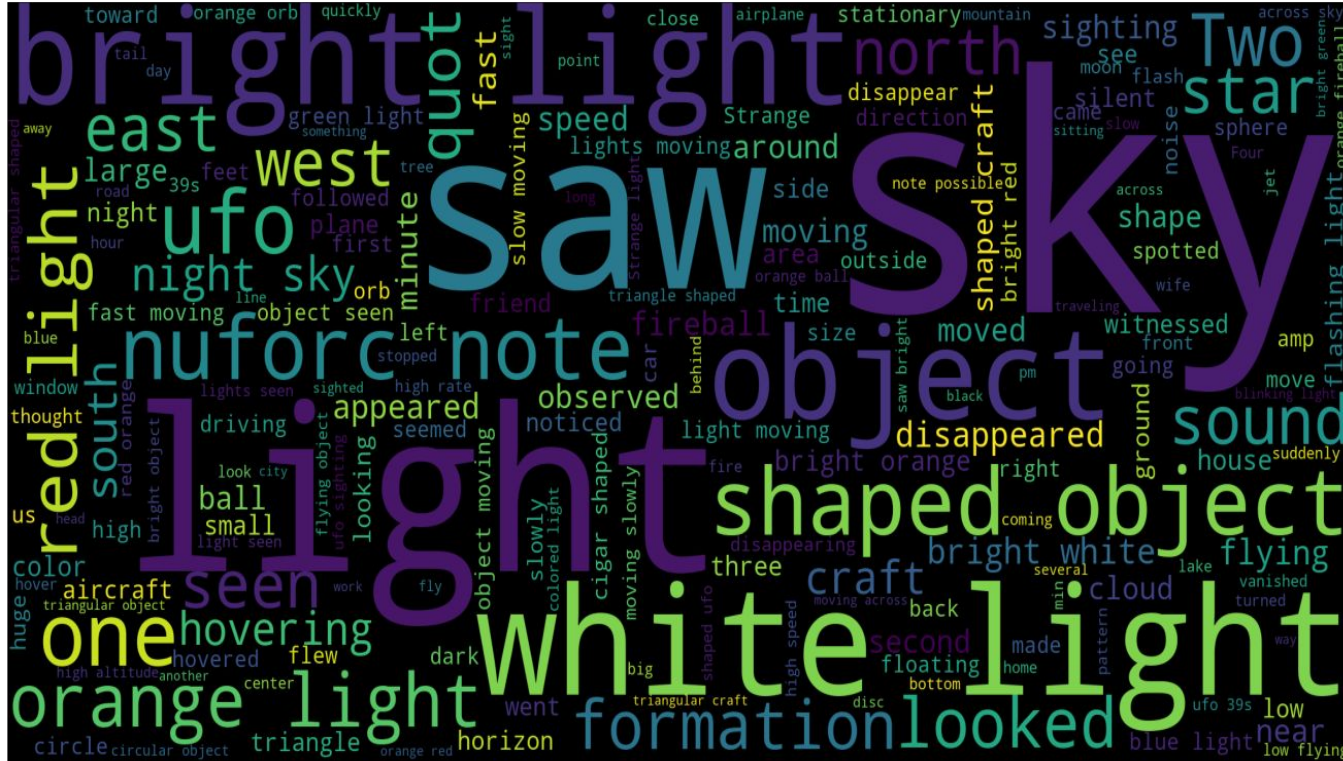
D

In the other hand; headers, locations (City, State, Country), shape and comments were capitalized (1st letter) for formatting.

Descriptive statistics

	City	State	Country	Shape	Duration (s)	Comments	Latitude	Longitude	Month	Day	Year	Month_posted	Day_posted	Year_posted
count	66516	66516	66516	66516	6.651600e+04	66516	66516.000000	66516.000000	66516	66516	66516	66516	66516	66516
unique	11920	67	4	28	NaN	66132	NaN	NaN	12	31	83	12	31	17
top	Seattle	Ca	Us	Light	NaN	Bright light	NaN	NaN	7	15	2012	8	12	2012
freq	471	8683	63553	14130	NaN	15	NaN	NaN	7972	4846	6489	8250	6196	7067
mean	NaN	NaN	NaN	NaN	6.572997e+03	NaN	38.707097	-95.293158	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	5.108910e+05	NaN	5.844058	18.480976	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	1.000000e-02	NaN	-37.813938	-176.658056	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	3.000000e+01	NaN	34.197500	-114.180556	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	1.800000e+02	NaN	39.246111	-89.598750	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	6.000000e+02	NaN	42.336944	-80.397500	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	8.280000e+07	NaN	72.700000	153.099533	NaN	NaN	NaN	NaN	NaN	NaN

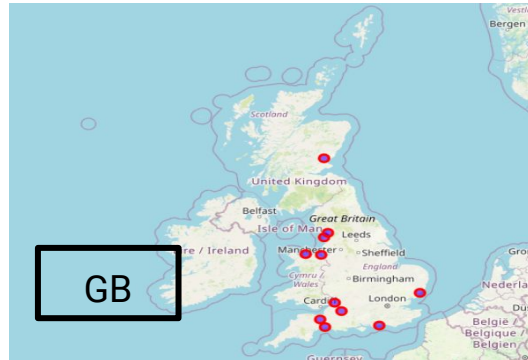
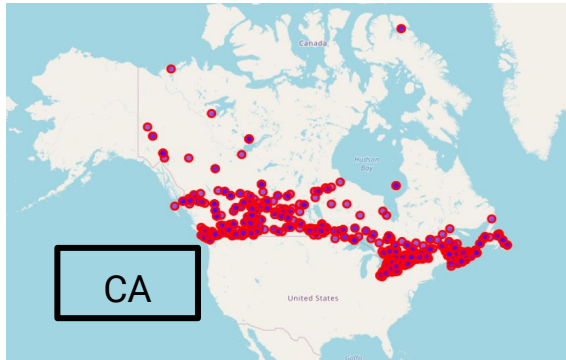
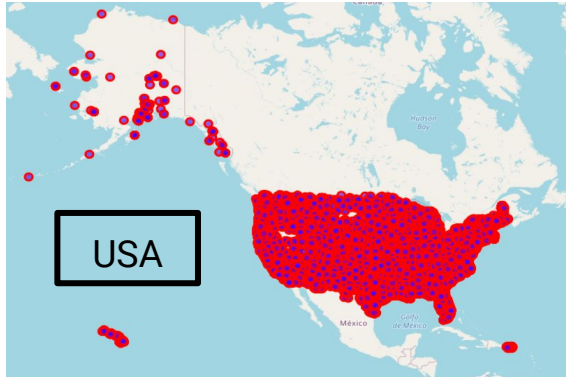
This table summarizes descriptive stats of both, numerical and categorical, data.



What does this mean?

- Word Cloud of the information within the “Comments” feature.
- The vast majority describes the event as: white, bright, light, sky, sound, object.

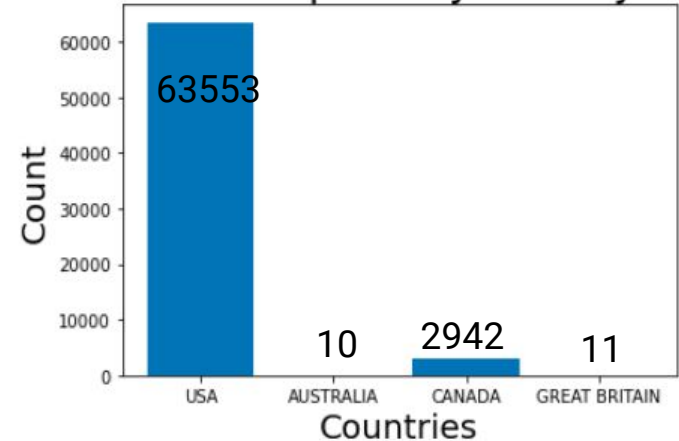
Visualization and Findings



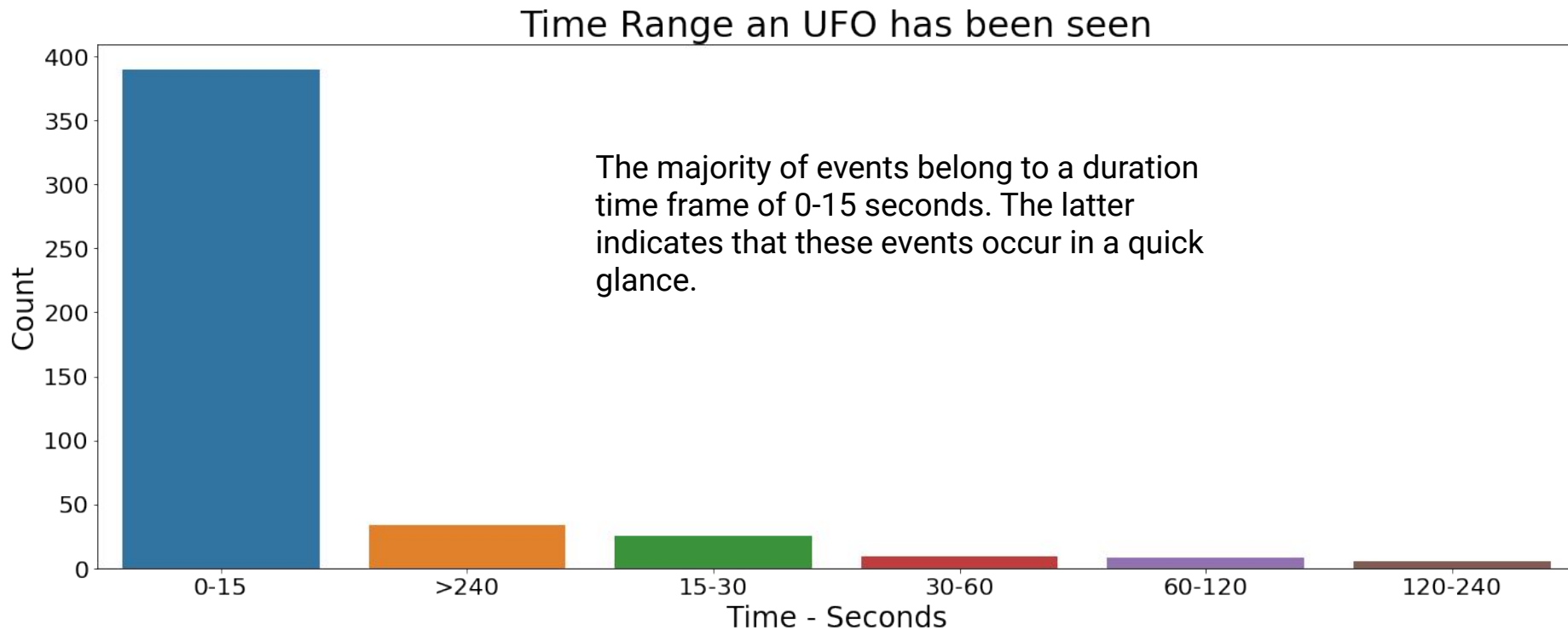
What does this mean?

- Geographical representation of the events
- The majority of the reported events are located in the US and Canada.

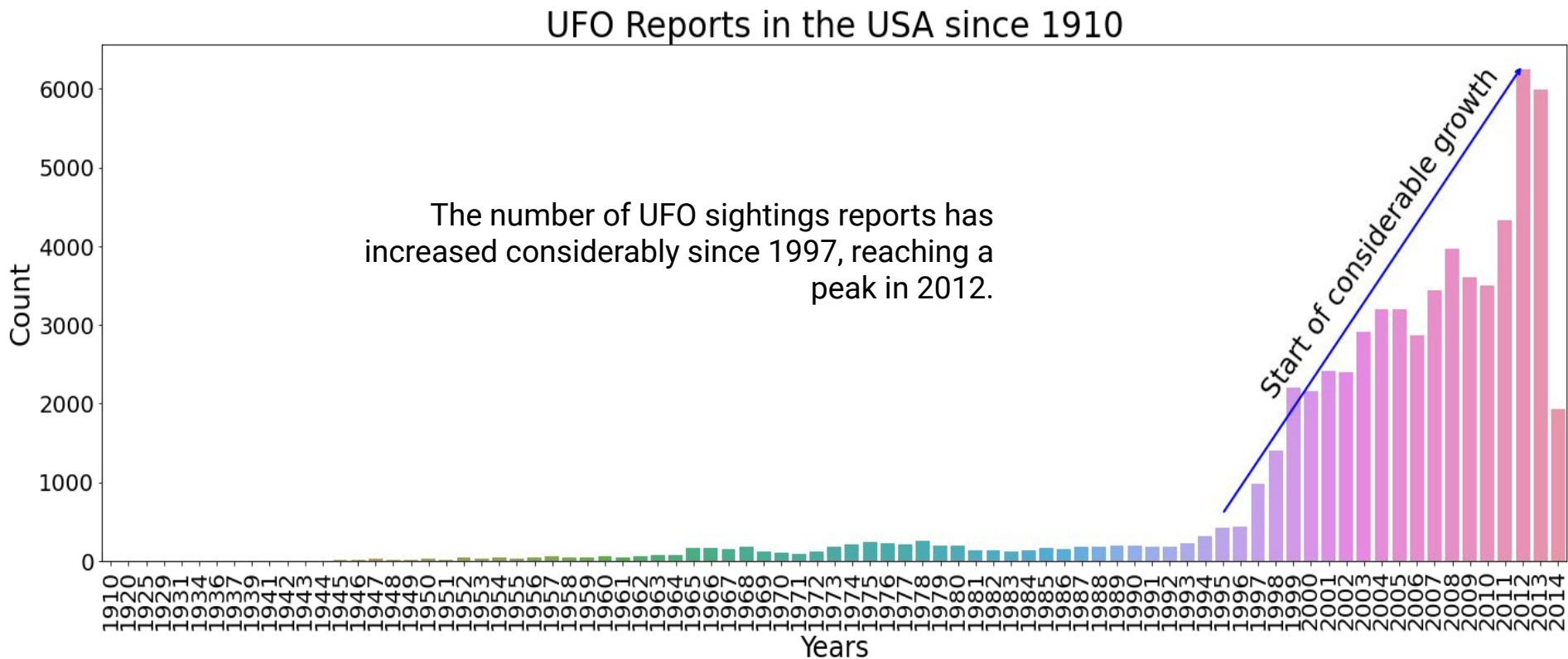
UFO Reports by country



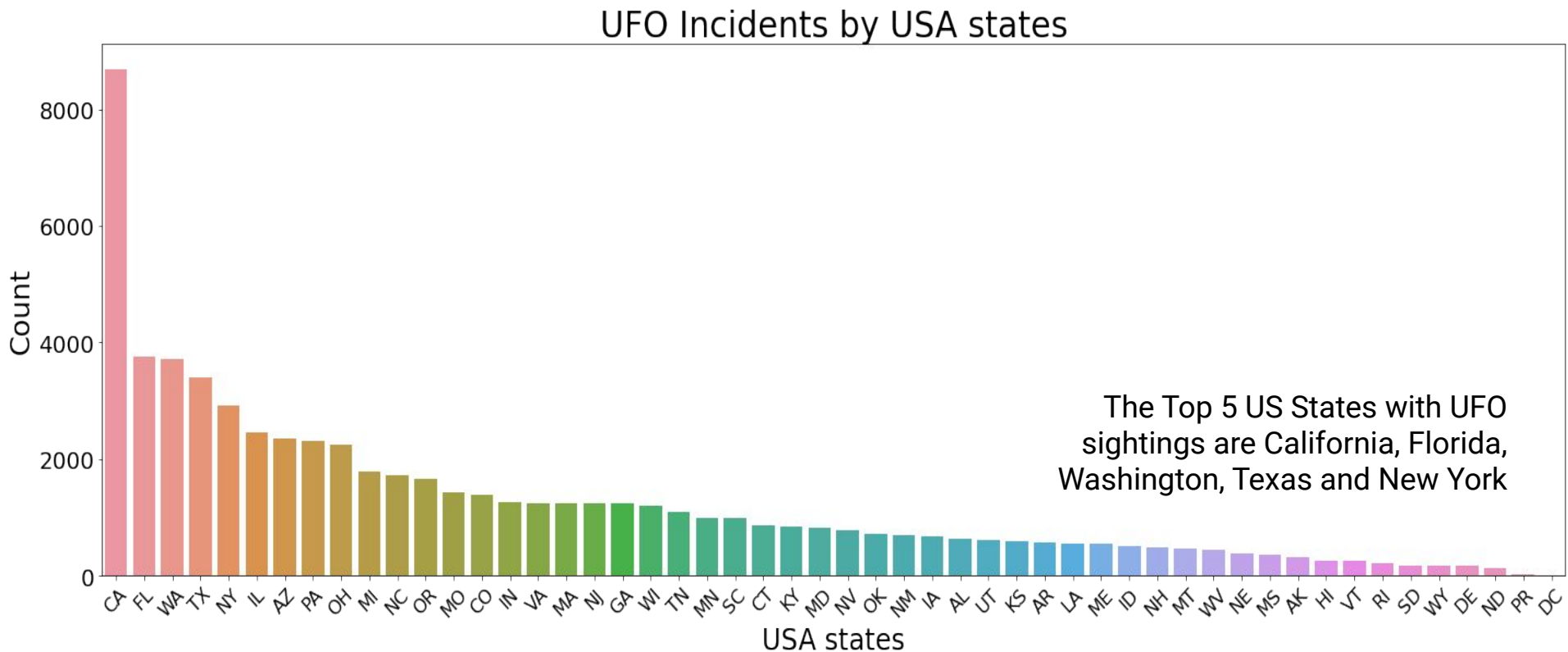
Visualization and Findings



Visualization and Findings



Visualization and Findings



Conclusion and Next Steps

- The vast majority of the events occurred within a time frame of 0-15 seconds
- The most used words to describe the UFO sightings are: white, bright, light, sky, sound, object.
- The majority of the events have been reported between 1997 and 2014, with a peak in 2012.
- The country with the most reported events is the US, where the TOP 5 states with UFO sightings are: California, Florida, Washington, Texas and New York
- Therefore, the best place to start researching/experience this event is California, US.

Next...

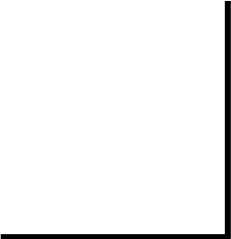
- To start gathering data exclusively from the US, with new attributes.
- Expand the comment section in: color, sound and movement description.
- Include the incident hour
- From these new events, have some of them being rejected as UFO? Which explanations have been reached? Where authorities involved? Are there any recordings?
- Centralize the analysis in the top 5 states with reported events.



The Lapidarist Problem



By Oscar Aguilar
September 2020



Index

- I. Introduction to the problem
- II. Data Inspection and descriptive statistics
- III. Dealing with categorical data and final sets
- IV. Modelling and performance evaluation
- V. Conclusion and next steps

Introduction and Problem

A lapidarist is a professional with an expertise on precious stones and the art of cutting-and-engraving them. For instance, they have considerable knowledge on the most famous precious stone: the diamond. The value of a diamond is commonly determined by a rule of thumb, named: The 4Cs: carat weight, color, cut and clarity.



In this problem, 10 diamonds have gone missing from Gringotts Wizarding Bank, and the PM urgently needs a Data Scientist to create a model to quantitatively value the pieces. Luckily, the previously mentioned bank has a database with the characteristics and price valuation of thousands of different diamonds. With this information, a model to value the missing pieces can be constructed.

Data Inspection and Descriptive Statistics

#Variable types

df.dtypes

```
carat    float64
cut       object
color     object
clarity   object
depth     float64
table     float64
price     int64
x         float64
y         float64
z         float64
dtype: object
```

A

Data types

Data shape: 53,930 rows, 10 columns

#Null Values

df.isnull().sum()

```
carat    0
cut       0
color     0
clarity   0
depth     0
table     0
price     0
x         0
y         0
z         0
dtype: int64
```

B

NaN values:
missing data

Data Inspection and Descriptive Statistics

	carat	cut	color	clarity	depth	table	price	x	y	z
count	53930.000000	53930	53930	53930	53930.000000	53930.000000	53930.000000	53930.000000	53930.000000	53930.000000
unique	NaN	5	7	8	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	Ideal	G	SI1	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	21546	11288	13065	NaN	NaN	NaN	NaN	NaN	NaN
mean	0.797976	NaN	NaN	NaN	61.749325	57.457328	3933.054942	5.731236	5.734601	3.538776
std	0.474035	NaN	NaN	NaN	1.432711	2.234578	3989.628569	1.121807	1.142184	0.705729
min	0.200000	NaN	NaN	NaN	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	NaN	NaN	NaN	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	NaN	NaN	NaN	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	NaN	NaN	NaN	62.500000	59.000000	5325.000000	6.540000	6.540000	4.040000
max	5.010000	NaN	NaN	NaN	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Dealing with categorical data and final sets

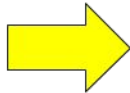
One Hot Encoding:

```
cut_dummies = pd.get_dummies(df.cut)
```

```
color_dummies = pd.get_dummies(df.color)
```

```
clarity_dummies = pd.get_dummies(df.clarity)
```

```
df=pd.concat([df, cut_dummies, color_dummies, clarity_dummies], axis=1)
```



Color			
Red			
Red			
Yellow			
Green			
Yellow			

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Dealing with categorical data and final sets

Final TRAINING Set

	carat	depth	table	x	y	z	Fair	Good	Ideal	Premium	...	J	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2	price
0	0.23	61.5	55.0	3.95	3.98	2.43	0	0	1	0	...	0	0	0	0	1	0	0	0	0	326
1	0.21	59.8	61.0	3.89	3.84	2.31	0	0	0	1	...	0	0	0	1	0	0	0	0	0	326
2	0.23	56.9	65.0	4.05	4.07	2.31	0	1	0	0	...	0	0	0	0	0	1	0	0	0	327
3	0.29	62.4	58.0	4.20	4.23	2.63	0	0	0	1	...	0	0	0	0	0	0	1	0	0	334
4	0.31	63.3	58.0	4.34	4.35	2.75	0	1	0	0	...	1	0	0	0	1	0	0	0	0	335
...
53925	0.72	60.8	57.0	5.75	5.76	3.50	0	0	1	0	...	0	0	0	1	0	0	0	0	0	2757
53926	0.72	63.1	55.0	5.69	5.75	3.61	0	1	0	0	...	0	0	0	1	0	0	0	0	0	2757
53927	0.70	62.8	60.0	5.66	5.68	3.56	0	0	0	0	...	0	0	0	1	0	0	0	0	0	2757
53928	0.86	61.0	58.0	6.15	6.12	3.74	0	0	0	1	...	0	0	0	0	1	0	0	0	0	2757
53929	0.75	62.2	55.0	5.83	5.87	3.64	0	0	1	0	...	0	0	0	0	1	0	0	0	0	2757

53930 rows x 27 columns

The columns were rearranged in order to have price (target variable) as the last feature

Dealing with categorical data and final sets

Missing Diamonds Set

	carat	cut	color	clarity	depth	table	x	y	z
0	0.71	Good	I	VVS2	63.1	58.0	5.64	5.71	3.58
1	0.83	Ideal	G	VS1	62.1	55.0	6.02	6.05	3.75
2	0.50	Ideal	E	VS2	61.5	55.0	5.11	5.16	3.16
3	0.39	Premium	J	VS1	61.6	59.0	4.67	4.71	2.89
4	0.32	Premium	G	VS1	62.1	56.0	4.43	4.40	2.74
5	0.90	Good	F	SI2	63.3	57.0	6.08	6.14	3.87
6	0.51	Ideal	D	VS1	60.9	57.0	5.20	5.17	3.16
7	1.12	Ideal	G	VVS2	62.1	54.8	6.64	6.66	4.13
8	0.40	Ideal	G	VVS2	62.4	56.0	4.72	4.74	2.95
9	0.36	Premium	I	VS2	62.7	59.0	4.54	4.58	2.86

A) One Hot Encoding for categorical variables

```
cut_dummies2 = pd.get_dummies(test.cut)
color_dummies2 = pd.get_dummies(test.color)
clarity_dummies2 =
pd.get_dummies(test.clarity)
test=pd.concat([test, cut_dummies2,
color_dummies2, clarity_dummies2], axis=1)
test.drop(['cut','color','clarity'], axis=1,
inplace=True)
```

B) Adding columns that are not included in the test set, but they are in the train set

```
test['Fair'] = 0
test['Very Good'] = 0
test['H'] = 0
test['I1'] = 0
test['IF'] = 0
test['SI1'] = 0
test['VVS1'] = 0
```

Dealing with categorical data and final sets

Final TESTING Set

	carat	depth	table	x	y	z	Fair	Good	Ideal	Premium	...	I	J	I1	IF	SI1	SI2	VS1	VS2	VVS1	VVS2
0	0.71	63.1	58.0	5.64	5.71	3.58	0	1	0	0	...	1	0	0	0	0	0	0	0	0	1
1	0.83	62.1	55.0	6.02	6.05	3.75	0	0	1	0	...	0	0	0	0	0	0	1	0	0	0
2	0.50	61.5	55.0	5.11	5.16	3.16	0	0	1	0	...	0	0	0	0	0	0	0	1	0	0
3	0.39	61.6	59.0	4.67	4.71	2.89	0	0	0	1	...	0	1	0	0	0	0	1	0	0	0
4	0.32	62.1	56.0	4.43	4.40	2.74	0	0	0	1	...	0	0	0	0	0	0	1	0	0	0
5	0.90	63.3	57.0	6.08	6.14	3.87	0	1	0	0	...	0	0	0	0	0	1	0	0	0	0
6	0.51	60.9	57.0	5.20	5.17	3.16	0	0	1	0	...	0	0	0	0	0	0	1	0	0	0
7	1.12	62.1	54.8	6.64	6.66	4.13	0	0	1	0	...	0	0	0	0	0	0	0	0	0	1
8	0.40	62.4	56.0	4.72	4.74	2.95	0	0	1	0	...	0	0	0	0	0	0	0	0	0	1
9	0.36	62.7	59.0	4.54	4.58	2.86	0	0	0	1	...	1	0	0	0	0	0	0	1	0	0

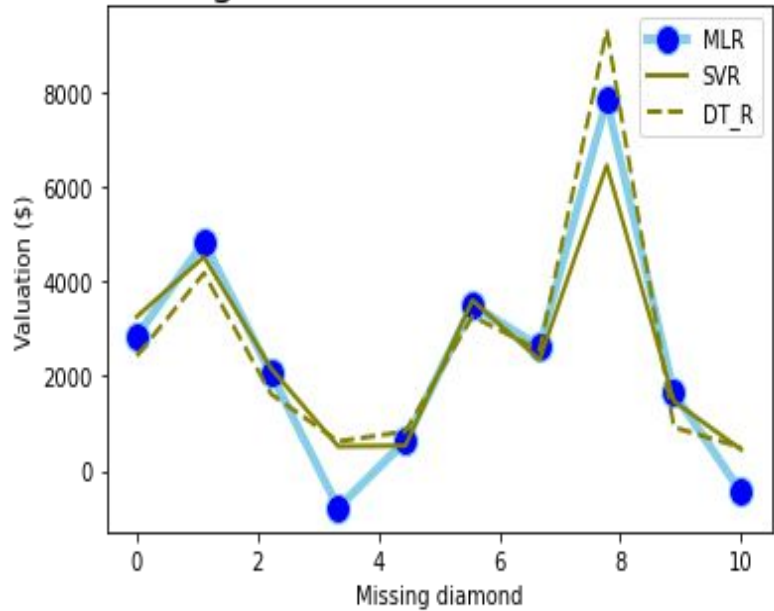
10 rows x 26 columns

Same number of features(27)-1, which is the target variable: price.

Modelling and performance evaluation

—

Missing values estimated valuation



Model/Metric	RMSE	R^2	Predictions
Multiple Linear Regression (MLR)	1251039.33	0.92	2853.01, 4838.81 , 2055.06 , -788.78, 623.36, 3503.57, 2615.85, 7839.25, 1648.97 , -458.50
Support Vector Regressor (SVR)	3237845.68	0.81	3260.43, 4524.86 , 2161.63, 512.22 , 527.65, 3605.21, 2345.57, 6465.56, 1470.73, 444.87
Decision Tree Regressor (DTR)	3237845.68	0.80	2428, 4181, 1624, 616, 828, 3267, 2550, 9333, 917, 491.

Conclusion and Next Steps

- The model based on MLR outperformed the models that used DTR and SVR.
- However, it can be improved due to the fact that it predicted 2 negative prices.
- Although there was a model with better evaluation metrics, SVR and DTR performed accurately as well.

Next...

- Attempt more regression models, such as polynomial or logistic regression.
- Use cross validation sampling to improve model evaluation
- Use a deep-learning approach such as convolutional neural networks (CNN)