Can Machine Learning be used to predict the survivability of a Honey Bee Hive?

FAES Spring 2019

BIOF 509

Final Project Presentation by Kevin Oakley

Recreational Honey Beekeeping

Apis mellifera, the honey bee, is an amazing insect. Humans have been fascinated by them for centuries, learning their behaviors and how tend to them recreationally. Beekeepers love their hives and share a common interest in supporting their strength to survive season after season. Unfortunately, with so many variables that can affect survival, it is not always clear if a hive will survive the winter.

The ideal objectives for this project would be to determine a model using machine learning that can help to parse through large data for variables which would indicate the survivability of an average hive.

The end goal would be to offer advice for beekeepers on how gauge the strength of a hive to enable their survival through the winter.

Some observations throughout an active honey bee season would be their level of natural defense against pests and parasites, as well as ability to store honey to sustain their colony. Indications that the colony is weak could also be the rearing of a new queen by building queen cells, or the active swarming away from their current hive in search of a new home.

Varroa Mites
Wax Moths
Swarms
Queen Cells
Honey Collected (g)
Sunshine (hr)

Common Pest and Parasites

Honey Bee Apiaries are under constant threats from invaders

1. **Bees have to contend with parasites (Varroa Mites) that evolved to feed from their abdominal hemolymph and fat deposits.**
2. **Wax moths exploit honey bee hives as their food source and ravage wax combs during their life cycle.**
3. **Humans may harvest a surplus of the hive's honey stores, thus limiting their resources to survive winter.**

(disclaimer: these are not meant to identify a comprehensive list of threats to honeybees)

Images found at this site
http://blogs.evergreen.edu/terroir-zack/pests-and-predators/

Extrinsic factors, like weather, directly affect the hives ability to forage for food. Seasonal sunlight and drought levels are clear indications of how regional plants grow and could be an indication of survivability.

Additional environmental factors, like pesticides, are huge concerns for all pollinators, however I will not attempt to address that level of prediction in this project.

https://www.unenvironment.org/resources/report/2019-visual-booklet-pollinators-and-pesticides-keeping-our-bees-safe

Risks

Pesticide risks to pollinators

Intensive agriculture threatens pollinators, pollination and wild bee diversity. This is because agricultural pesticide use can have significant negative effects on species abundance. The risk to pollinators from pesticides arises through a combination of toxicity and the level of exposure. Pesticides, particularly insecticides, have been demonstrated to have a broad range of lethal effects on pollinators. A diverse community of pollinators provides more effective and stable crop pollination than any single species.

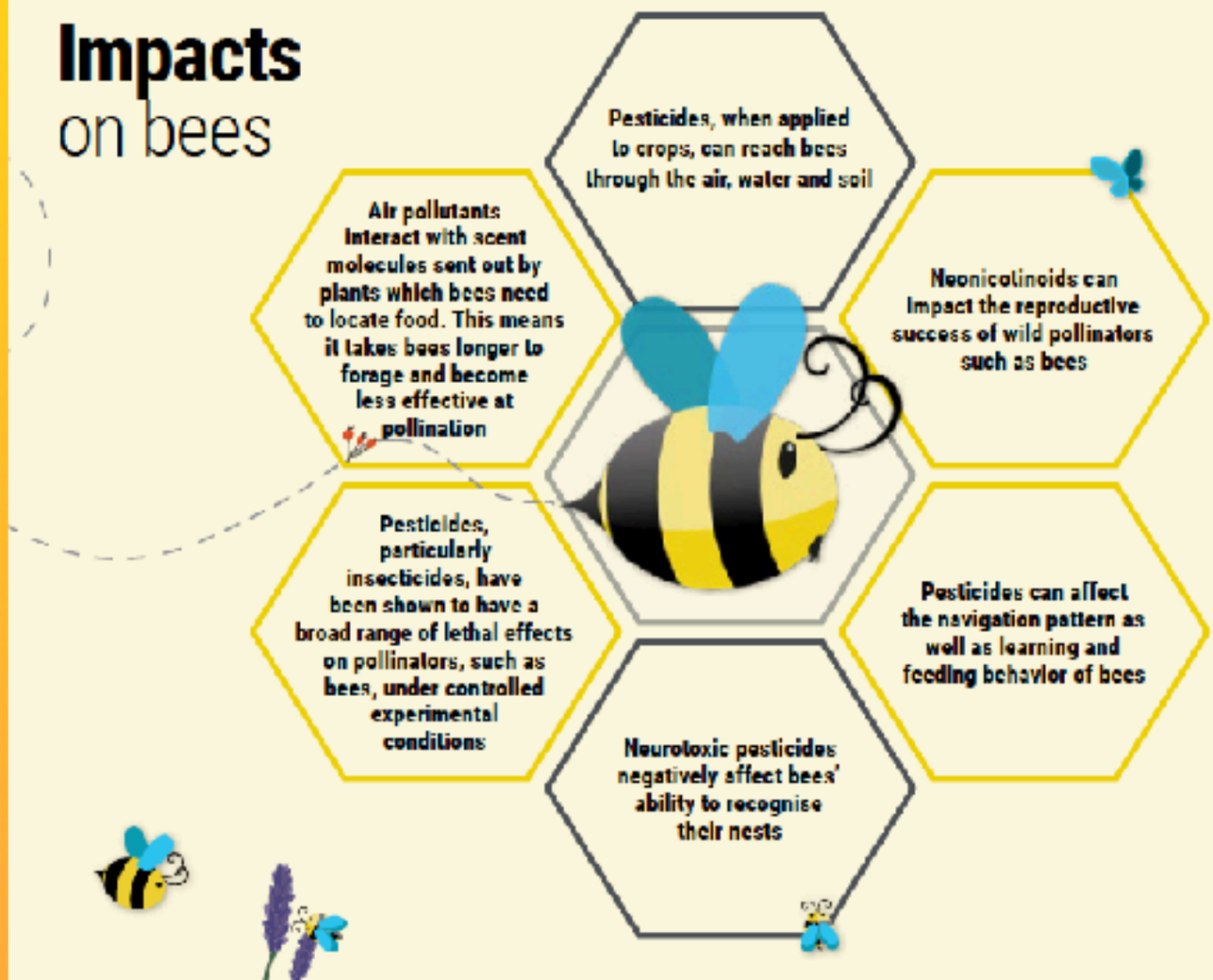
Wild bee diversity contributes to crop production even when honey bees are present in high abundance. The contribution of wild pollinators to crop production is undervalued.

The use of herbicides to control weeds indirectly affects pollinators by reducing the abundance and diversity of flowering plants providing pollen and nectar.

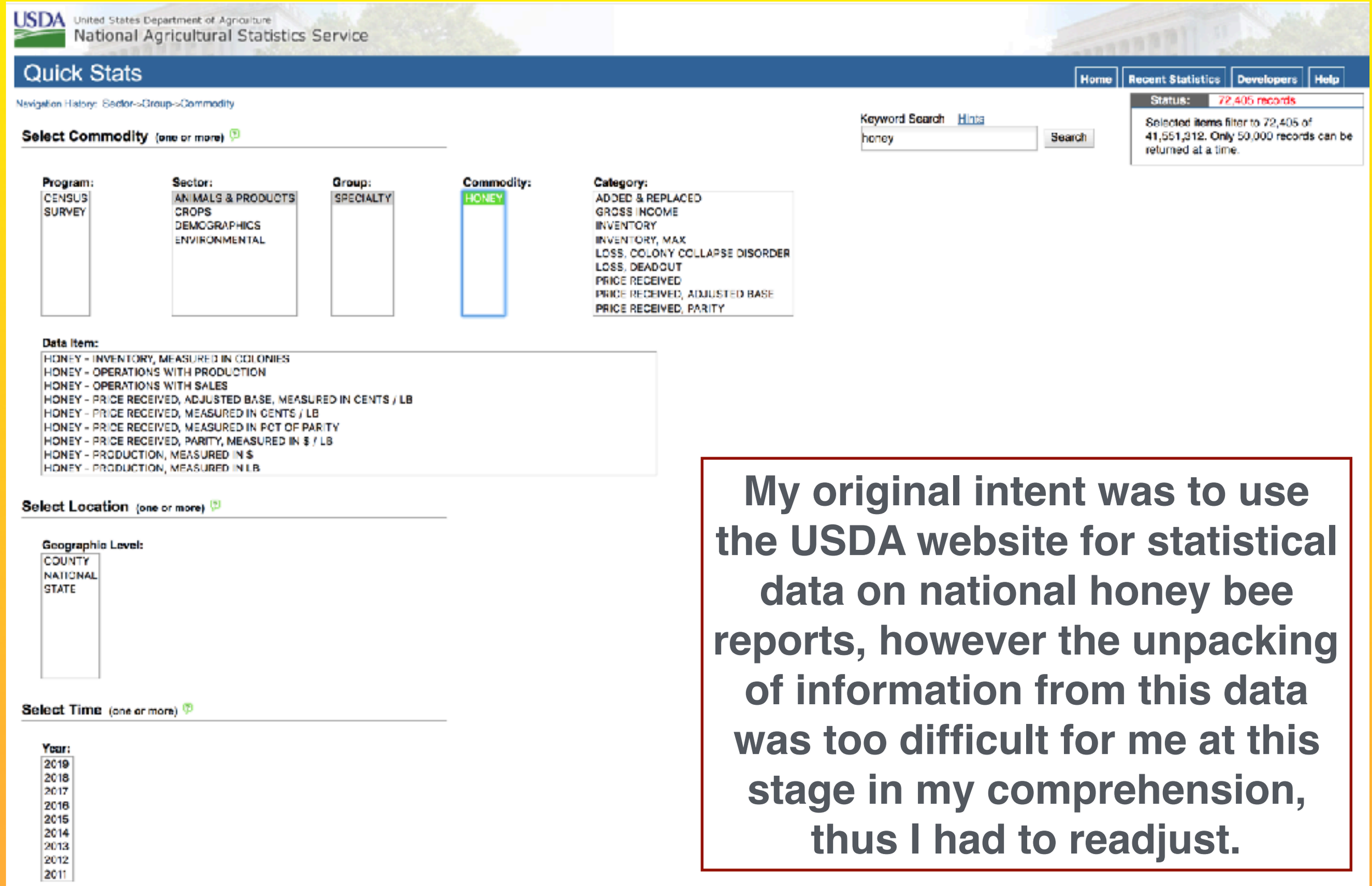
Climate Change

The ranges, abundances and seasonal activities of some wild pollinator species (e.g., bumble bees and butterflies) have changed in response to observed climate change over recent decades.

Generally, the impacts of ongoing climate change on pollinators and pollination services to agriculture may not be fully apparent for several decades.



Searching for a dataset



My original intent was to use the USDA website for statistical data on national honey bee reports, however the unpacking of information from this data was too difficult for me at this stage in my comprehension, thus I had to readjust.

Searching for a dataset 

Ultimately, I could not find access to a dataset that would provide information to derive the solution to my question.
I had to make a difficult choice, but decided to fabricate a dataset based on some observations that I assume will be indicators of hive strength.

I take no pride in making this data, but in my novice exposure to data, I did not have time to discover a suitable real world alternative to meet the deadline of the project. I believe I can illustrate the approach to ML on this invented data for the purpose of the assignment.

The lack of data available may lead to a collaboration with researchers who are interested in the same work. There has been talk of building a citizen reporting app/online journal that can collect the data observations of recreational beekeepers to address regional trends of hive survivability.

Fabricated Dataset

These are the stats for each column of the combined data that I used for my dataset. I created individual spreadsheets (following slides) using the "Randbetween" Numbers (Apple) option to populate cells based on some parameters that I assumed.

The goal was to randomize most of the data with some biased values that could indicate strong vs. weak hives based on the background provided in the previous slides.

*** I had to choose a min =1 for Varroa because my logScaling was producing divisible by zero errors. In the natural world, presence of only a few mites is uncommon.**

Statistics of Random Data Generated for "Fabricated apiary data"				
Rows = 2891	Mode	Min	Max	Average
Varroa Mites Observed	3	1*	133	60.599
Hours of Sun	2676	509	3373	2149.633
Wax Moths	14	0	15	7.836
Total Honey	11728	1232	19997	10151.641
Swarm Attempted	0	0	1	0.495
Queen Cells Produced	2	0	6	1.943
Survival	0	0	1	0.486

Fabricated Dataset: Survivors

Range of Randomness	Statistics of Random Data Generated for "Survivor" Hives				
	Rows = 1050	Mode	Min	Max	Average
Varroa Mites observed rand= 1-10	Varroa Mites Observed	7	1	10	4.94
Hours of Sun rand= 2300-3374	Hours of Sun	2624	2300	3373	2835.16
Wax Moth Observed rand= 1-15	Wax Moths	11	1	15	8.07
Total Honey Collected (weight in kg) rand= 10000-20000	Total Honey	15576	10015	19997	14920.89
Swarm Attempt (Yes = 1; 0 = No)	Swarm Attempted	0	0	1	0.26
Queen Cells produced rand= 0-3	Queen Cells Produced	2	0	3	1
Survival (Yes = 1, 0 = No)	Survival	1	0	1	0.71

For the "Survivor" hives, my assumptions were that less parasites, more sunny days, more total honey harvested, less swarm attempts, less queen cells produced and higher rate of survival.

Fabricated Dataset: Collapsed

Range of Randomness
Varroa Mites observed rand= 90-133
Hours of Sun rand= 509-2800
Wax Moth Observed rand= 1-15
Total Honey Collected (weight in kg) rand= 2000-12000
Swarm Attempt (Yes = 1; 0 = No)
Queen Cells produced rand= 2-6
Survival (Yes = 1, 0 = No)

Statistics of Random Data Generated for "Collapsed" Hives				
Rows = 1050	Mode	Min	Max	Average
Varroa Mites Observed	95	90	133	111.99
Hours of Sun	2358	510	2800	1635.09
Wax Moths	1	1	15	7.86
Total Honey	5218	2004	11982	6901.35
Swarm Attempted	1	0	1	0.72
Queen Cells Produced	2	0	5	1.54
Survival	0	0	1	0.27

For the "collapsed" hives (or the ones that did not survive the season, my assumptions were that more parasites, less sunny days, less total honey harvested, more swarm attempts, more queen cells produced and lower rate of survival.

Fabricated Dataset: Random

Range of Randomness	Statistics of Random Data Generated for "Random" Hives				
	Rows = 792	Mode	Min	Max	Average
Varroa Mites observed rand= 90-133	Varroa Mites Observed	83	1	133	66.24
Hours of Sun rand=509-2800	Hours of Sun	967	509	3369	1923.45
Wax Moth Observed rand= 1-15	Wax Moths	6	0	15	7.50
Total Honey Collected (weight in kg) rand= 2000-12000	Total Honey	6991	1232	14956	8142.32
Swarm Attempt (Yes = 1; No= 0)	Swarm Attempted	1	0	1	0.50
Queen Cells produced rand= 2-6	Queen Cells Produced	6	0	6	3.09
Survival (Yes = 1, 0 = No)	Survival	0	0	1	0.48

For the "Random" hives, my assumptions were none.

I allowed the software to randomly fill the sheet with values in the ranges listed.

My Workflow

Explore the data

↓

Scaling and Normalizing the data

↓

Principle Component Analysis (PCA)

↓

t-distributed Stochastic Neighbor Embedding (TSNE)

↓

Choose the best Classification Model

Explore the data

fabap_df = fabricated apiary data

```
In [2]: ### read in the fabricated dataset for the random apiary data
        fabap_url = 'https://raw.githubusercontent.com/ContoakleyK/BIOF509-Bee-Trends/master/Fabricated_apiary_data.csv'
        fabap_df = pd.read_csv(fabap_url)
        print(fabap_df.shape)

        fabap_df.info()
```

```
(2891, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2891 entries, 0 to 2890
Data columns (total 7 columns):
Varroa Mites observed                    2891 non-null int64
Hours of Sun                             2891 non-null int64
Wax Moth Observed                        2891 non-null int64
Total Honey Collected  (weight in g)      2891 non-null int64
Swarm Attempt (Yes = 1; 0 = No)          2891 non-null int64
Queen Cells produced                     2891 non-null int64
Survival (Yes = 1, 0 = No)               2891 non-null int64
dtypes: int64(7)
memory usage: 158.2 KB
```

```
In [3]: ### inspect the distribution of data withing the columns
        fabap_df.describe()
```

Out[3]:

	Varroa Mites observed	Hours of Sun	Wax Moth Observed	Total Honey Collected (weight in g)	Swarm Attempt (Yes = 1; 0 = No)	Queen Cells produced	Survival (Yes = 1, 0 = No)
count	2891.000000	2891.000000	2891.000000	2891.000000	2891.000000	2891.000000	2891.000000
mean	60.629886	2149.555345	7.836389	10161.841501	0.496330	1.943272	0.496367
std	50.584666	815.959618	4.362534	4877.990777	0.500065	1.606507	0.499990
min	1.000000	508.000000	0.000000	1232.000000	0.000000	0.000000	0.000000
25%	7.000000	1479.000000	4.000000	6978.500000	0.000000	1.000000	0.000000
50%	66.000000	2364.000000	8.000000	10418.000000	0.000000	2.000000	0.000000
75%	110.000000	2793.000000	12.000000	13579.500000	1.000000	3.000000	1.000000
max	135.000000	3373.000000	15.000000	19997.000000	1.000000	5.000000	1.000000

```
In [6]: ### displayed data, then repeated previous cell to show in the subsequent cell that the shuffle worked.
        fabap_df
```

Out[6]:

	Varroa Mites observed	Hours of Sun	Wax Moth Observed	Total Honey Collected (weight in g)	Swarm Attempt (Yes = 1; 0 = No)	Queen Cells produced	Survival (Yes = 1, 0 = No)
0	60	2181	10	3907	1	5	1
1	1	2481	9	18235	1	3	0
2	125	2308	9	6908	1	0	0
3	97	1056	14	8396	1	2	0
4	119	823	15	6900	1	2	0
5	3	2763	3	15706	1	0	0
6	128	563	9	6068	1	2	0
7	100	2229	7	9486	1	0	0
8	3	2823	12	14188	1	2	0
9	9	2876	12	12731	0	1	1

Preprocessing the data

In [8]:
```
### edit the column headings to remove the randomisation criteria used to generate them.
fabap_df.columns=['Varroa Mites','Sunshine (hr)', 'Wax Moths','Honey Collected (g)','Swarm','Queen Cells','Survival']
fabap_df.head()
```

Out[8]:

	Varroa Mites	Sunshine (hr)	Wax Moths	Honey Collected (g)	Swarm	Queen Cells	Survival
0	1	2966	7	13165	0	2	1
1	128	927	11	5092	1	1	0
2	131	2998	1	10769	1	0	0
3	123	935	8	6107	1	1	0
4	72	1747	5	3121	0	5	0

In [9]:
```
### convert values from the 'Sunshine (hr)' column from hours to days; and the 'Honey Collected (g)' column values to k
fabap_df['Sunshine (days)'] = fabap_df['Sunshine (hr)'].apply(lambda val: val/24)
fabap_df['Honey Collected (kg)'] = fabap_df['Honey Collected (g)'].apply(lambda val: val/1000)
fabap_df.head()
```

Out[9]:

	Varroa Mites	Sunshine (hr)	Wax Moths	Honey Collected (g)	Swarm	Queen Cells	Survival	Sunshine (days)	Honey Collected (kg)
0	1	2966	7	13165	0	2	1	123.125000	13.165
1	128	927	11	5092	1	1	0	38.625000	5.092
2	131	2998	1	10769	1	0	0	124.916667	10.769
3	123	935	8	6107	1	1	0	38.958333	6.107
4	72	1747	5	3121	0	5	0	72.791667	3.121

In [10]:
```
### drop the old columns
fabap_df = fabap_df.drop(['Sunshine (hr)','Honey Collected (g)'], axis=1)
fabap_df.head()
```

Out[10]:

	Varroa Mites	Wax Moths	Swarm	Queen Cells	Survival	Sunshine (days)	Honey Collected (kg)
0	1	7	0	2	1	123.125000	13.165
1	128	11	1	1	0	38.625000	5.092
2	131	1	1	0	0	124.916667	10.769
3	123	8	1	1	0	38.958333	6.107
4	72	5	0	5	0	72.791667	3.121

In [11]:
```
fabap_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2891 entries, 0 to 2890
Data columns (total 7 columns):
Varroa Mites            2891 non-null int64
Wax Moths               2891 non-null int64
Swarm                   2891 non-null int64
Queen Cells             2891 non-null int64
Survival                2891 non-null int64
Sunshine (days)         2891 non-null float64
Honey Collected (kg)    2891 non-null float64
dtypes: float64(2), int64(5)
memory usage: 158.2 KB
```

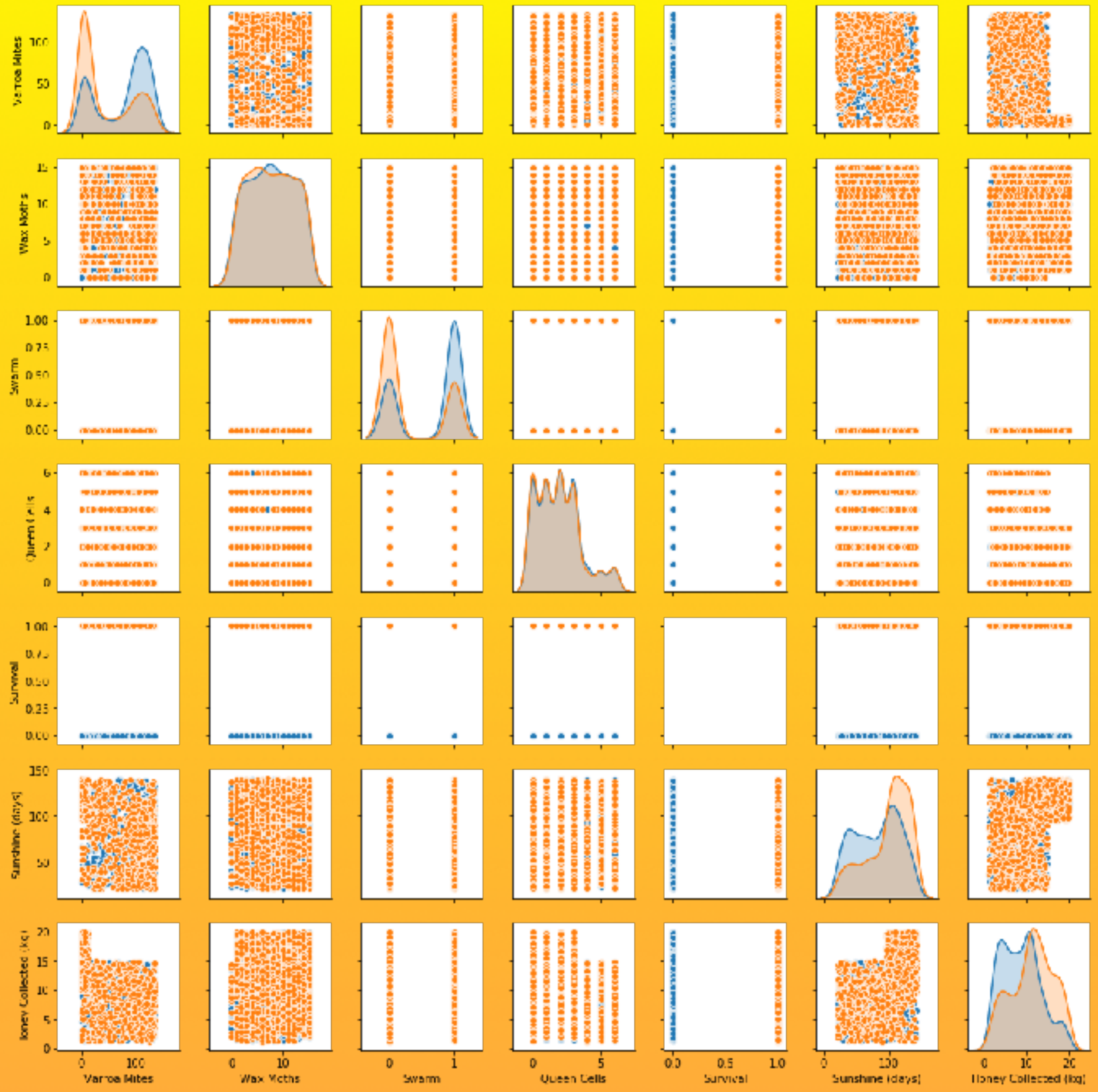
Visualize the data

Visualizing the data as it relates to the column ‘Survival’ in a pair plot does not identify any obvious feature correlation.

Standardizing the data for normalizing and scaling for feature selection and modeling is the next step

```
### Check the variance of the data
print(fabap_df.var())
```

```
Varroa Mites              2558.808297
Wax Moths                   19.031700
Swarm                        0.250065
Queen Cells                  2.580864
Survival                     0.249900
Sunshine (days)           1155.865756
Honey Collected (kg)        23.790892
dtype: float64
```



Standardizing the data

```
### Apply the log normalization function to the Varroa Mites and Sunshine (days) columns
fabap_df['Varroa_log'] = np.log(fabap_df['Varroa Mites'])
fabap_df['Sunny_log'] = np.log(fabap_df['Sunshine (days)'])


### Check the variance of the columns again
print(fabap_df['Sunny_log'].var())
print(fabap_df['Varroa_log'].var())
```

```
0.23501179305797923
2.669995615056251
```

```
# the data needs to be scaled: try Standard Scaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
Var_subset = fabap_df[["Varroa Mites"]]
Varroa_scaled = pd.DataFrame(scaler.fit_transform(Var_subset))
```

```
# create a scaled DataFrame to compare accuracy scores from original to normalized
fabap_norm = fabap_df.take([1,2,3,4,6,7,8], axis=1)
fabap_norm.head()
```

	Wax Moths	Swarm	Queen Cells	Survival	Honey Collected (kg)	Varroa_log	Sunny_log
0	10	1	5	1	3.907	3.912023	4.509485
1	9	1	3	0	18.235	0.000000	4.638363
2	8	1	0	0	6.908	4.828314	4.566083
3	14	1	2	0	8.398	4.574711	3.784190
4	15	1	2	0	6.900	4.779123	3.256493

```
# confirm variance of the normalized data
print(fabap_norm.var())
```

```
Wax Moths              19.031700
Swarm                   0.250065
Queen Cells             2.580864
Survival                0.249900
Honey Collected (kg)   23.790892
Varroa_log              2.669996
Sunny_log               0.235012
dtype: float64
```

Normalize the data

```
# define X for scaling purposes
X = fabap_norm.drop(['Survival'], axis=1)
```

```
# the data needs to be scaled: try Standard Scaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

X_scaled = pd.DataFrame(scaler.fit_transform(X))
```

```
print(X_scaled.describe())
```

	0	1	2	3	4	\
count	2.891000e+03	2.891000e+03	2.891000e+03	2.891000e+03	2.891000e+03	
mean	2.795719e-17	-5.844896e-17	-1.940106e-16	-1.106383e-16	5.527113e-16	
std	1.000173e+00	1.000173e+00	1.000173e+00	1.000173e+00	1.000173e+00	
min	-1.796604e+00	-9.907039e-01	-1.209835e+00	-1.829015e+00	-2.002149e+00	
25%	-8.795468e-01	-9.907039e-01	-5.872589e-01	-8.563375e-01	-8.110639e-01	
50%	3.751020e-02	-9.907039e-01	3.531737e-02	5.461811e-02	5.623241e-01	
75%	9.545672e-01	1.009383e+00	6.578937e-01	7.028986e-01	8.749987e-01	
max	1.642360e+00	1.009383e+00	2.525623e+00	2.018837e+00	9.912167e-01	

	5
count	2.891000e+03
mean	-1.618790e-15
std	1.000173e+00
min	-2.768913e+00
25%	-5.682090e-01
50%	3.993849e-01
75%	7.434377e-01
max	1.132725e+00

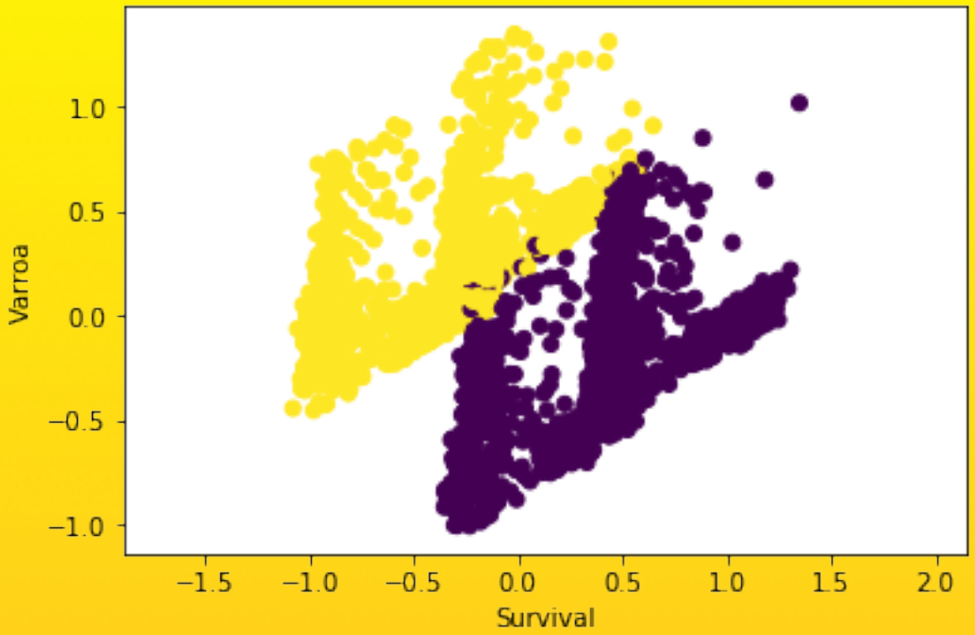
```
# the data needs to be scaled: Try MinMaxScaler as well
from sklearn.preprocessing import MinMaxScaler
scale = MinMaxScaler()

X_mM = pd.DataFrame(scale.fit_transform(X))
```

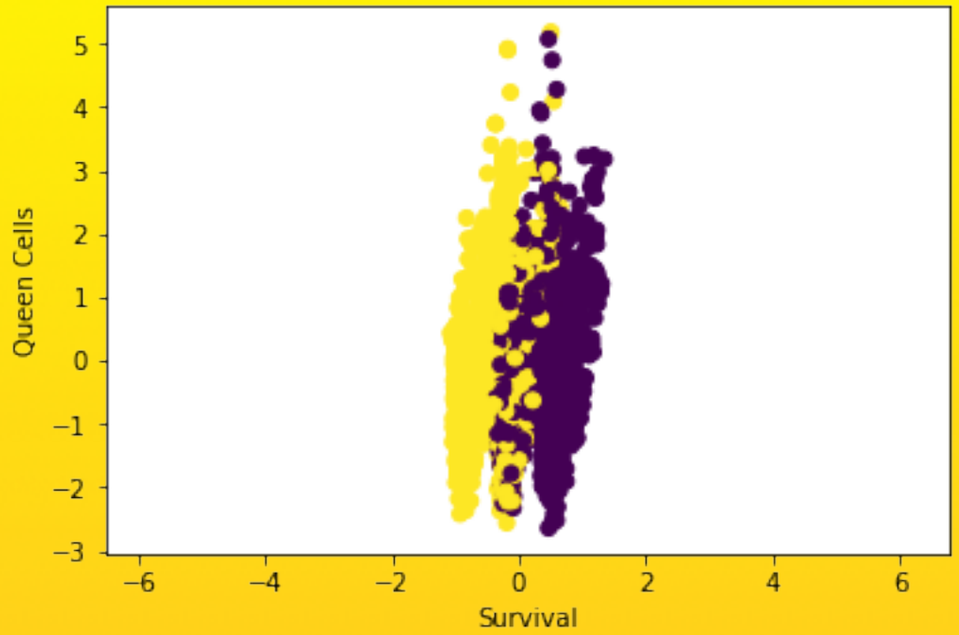
```
print(X_mM.describe())
```

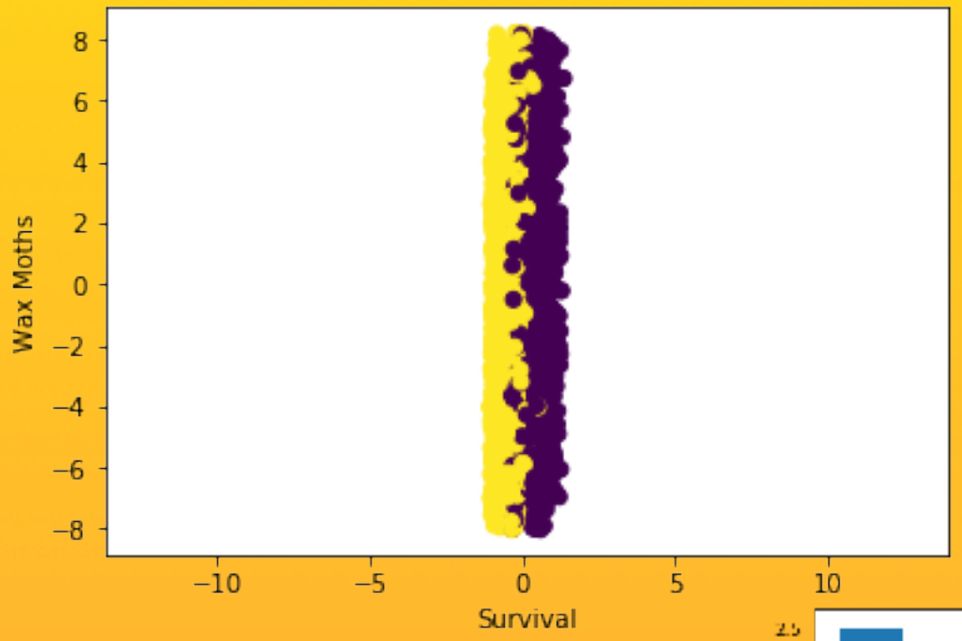
	0	1	2	3	4	\
count	2891.000000	2891.000000	2891.000000	2891.000000	2891.000000	
mean	0.522426	0.495330	0.323879	0.475334	0.668862	
std	0.290835	0.500065	0.267751	0.259930	0.334130	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.266667	0.000000	0.166667	0.252784	0.397908	
50%	0.533333	0.000000	0.333333	0.489528	0.856719	
75%	0.800000	1.000000	0.500000	0.658007	0.961175	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

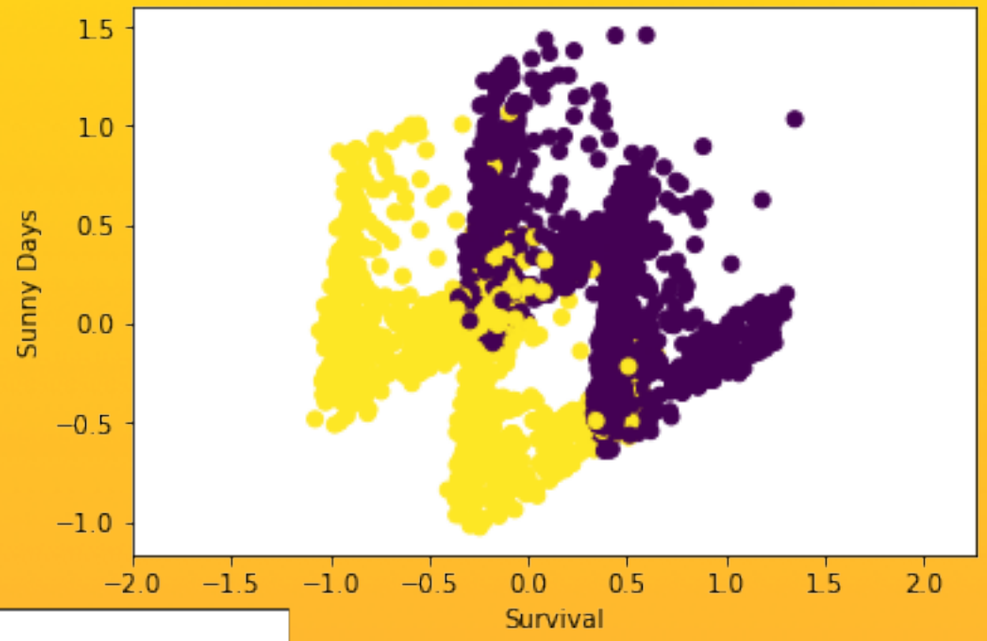
	5
count	2891.000000
mean	0.709680
std	0.256347
min	0.000000
25%	0.564046
50%	0.812043
75%	0.900225
max	1.000000

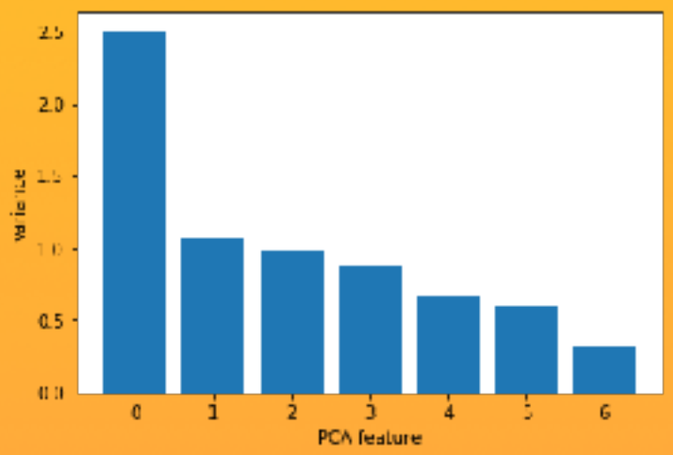
PCA of normalized data



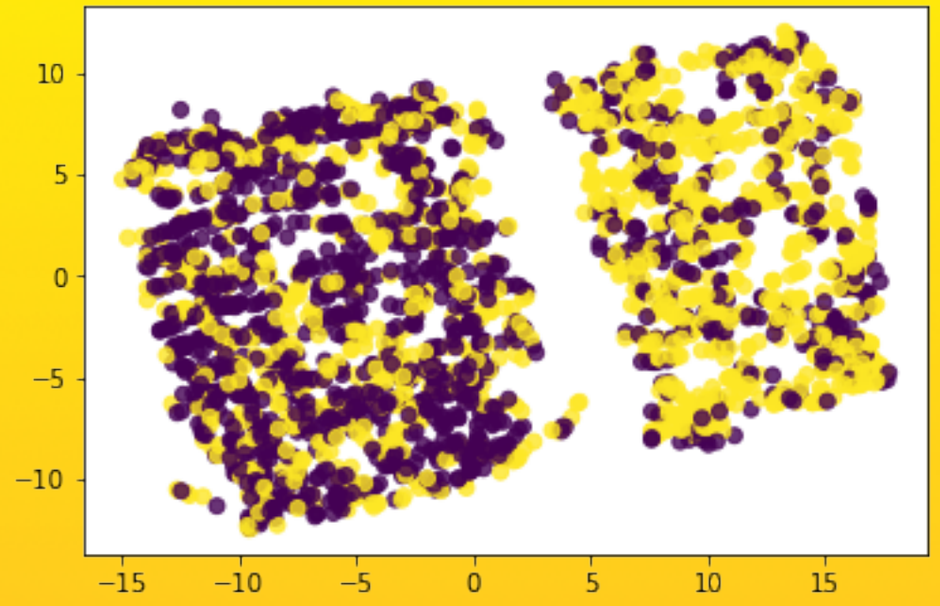








TSNE plot




```
[t-SNE] Computing 115 nearest neighbors...
[t-SNE] Indexed 2891 samples in 0.001s...
[t-SNE] Computed neighbors for 2891 samples in 0.071s...
[t-SNE] Computed conditional probabilities for sample 1000 / 2891
[t-SNE] Computed conditional probabilities for sample 2000 / 2891
[t-SNE] Computed conditional probabilities for sample 2891 / 2891
[t-SNE] Mean sigma: 1.171493
[t-SNE] KL divergence after 250 iterations with early exaggeration: 67.799423
[t-SNE] KL divergence after 500 iterations: 0.843075
```

I was able to tease the data into two main clusters with hyperparameters, but I never discovered the meaning within this arrangement

Applying Classifiers

Classifier	Score
K Nearest Neighbors	0.6542185
AdaBoost	0.6973367
SVM	0.6694329
Random Forrest	0.6583679
Logistic Regression	0.6694329

With all of the models preforming around 65%, I believe this is too coincidental.
I'm sure there is something in my code that is not correct, however I don't know how to determine the error.

Perhaps that code is fine, but the fact that the data was not field collected has more determination of the scoring for these models?

Conclusion

I cannot confidently conclude which model best performed to predict hive survival based on the dataset that was used.

My results are inconclusive and require more comprehension of machine learning to pull meaning from.

In the future, I plan to continue reviewing the concepts and logic in ML to better rationalize my approach to data. I look forward to the opportunity to solve the goal question of this project, potentially with the aid of real world citizen reported data that has more meaningful features to parse information from.

This exposure to ML has been beneficial. Even though I fell behind the learning curve, I will continue using the tools from sklearn to discovery practical approaches to data analysis.

(…and now that I'm aware of Kaggle, I'll use their examples as guides to refine how I search for and choose data for analysis)