september 9, 2024

exoplanet exploration

looked around at some articles to get ideas

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **455,** 3110–3125 (2016) doi:10.1093/mnras/stv2419

The host stars of *Kepler*'s habitable exoplanets: superflares, rotation and activity

D. J. Armstrong,[1,2*] C. E. Pugh,[1] A.-M. Broomhall,[1] D. J. A. Brown,[1] M. N. Lund,[3] H. P. Osborn[1] and D. L. Pollacco[1]

[1] *University of Warwick, Department of Physics, Gibbet Hill Road, Coventry CV4 7AL, UK*
[2] *ARC, School of Mathematics & Physics, Queen's University Belfast, University Road, Belfast BT7 1NN, UK*
[3] *Stellar Astrophysics Centre (SAC), Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, DK-8000 Aarhus C, Denmark*

Accepted 2015 October 16. Received 2015 October 15; in original form 2015 September 2

ABSTRACT

We embark on a detailed study of the light curves of *Kepler*'s most Earth-like exoplanet host stars using the full length of *Kepler* data. We derive rotation periods, photometric activity indices, flaring energies, mass-loss rates, gyrochronological ages, X-ray luminosities and consider implications for the planetary magnetospheres and habitability. Furthermore, we present the detection of superflares in the light curve of Kepler-438, the exoplanet with the highest Earth Similarity Index to date. Kepler-438b orbits at a distance of 0.166 au to its host star, and hence may be susceptible to atmospheric stripping. Our sample is taken from the Habitable Exoplanet Catalogue, and consists of the stars Kepler-22, Kepler-61, Kepler-62, Kepler-174, Kepler-186, Kepler-283, Kepler-296, Kepler-298, Kepler-438, Kepler-440, Kepler-442, Kepler-443 and KOI-4427, between them hosting 15 of the most habitable transiting planets known to date from *Kepler*.

Key words: planets and satellites: general – planets and satellites: magnetic fields – stars: activity – stars: flare – stars: rotation.

1 INTRODUCTION

In recent years the pace of discovery of exoplanets has intensified, with an increasing number of small, potentially rocky planets being found. This is largely due to the *Kepler* mission (Borucki et al. 2010; Koch et al. 2010), which studies ∼150 000 stars with high-precision photometry, and has found several thousand candidate exoplanets. With more planets comes a focus on new questions, including the potential habitability of these planets in a much more diverse range of environments than known in the Solar system.

The key driver of an exoplanet's local environment is its host star. This leads to well-known properties such as the equilibrium temperature of the exoplanet, defining the habitable zone where liquid water could exist on the planet's surface (e.g. Kasting 1993). Many of the newly discovered planets orbit stars cooler than the Sun, because around these stars habitable zone planets are often easier to detect. These stars are known to have increased activity relative to Sun-like stars (Wright et al. 2011), as well as increased potential for flaring, coronal mass ejections (CMEs), X-ray and EUV flux (Mamajek & Hillenbrand 2008), and active magnetic fields (Reiners & Mohanty 2012; Vidotto et al. 2014). Such properties tend to get weaker as a host star ages and spins-down (e.g. Hempelmann et al. 1995; van Saders & Pinsonneault 2013; García et al. 2014, along with many others) and all have a potential effect on planetary habitability.

Stellar flares are associated with increased UV and charged particle flux, but this is thought not to affect planetary habitability (Segura et al. 2010). They are, however, also associated with increased likelihood of CMEs (Chen & Kunkel 2010), which can compress planetary magnetospheres (Khodachenko et al. 2007) and drive atmospheric erosion (Lammer et al. 2007). UV flux, whether from a flare or background stellar radiation, can affect atmospheric heating and chemistry, as well as changing the biomarkers which future missions might search for (France, Linsky & Parke Loyd 2014; Grenfell et al. 2014). Stellar activity is associated with increased flaring rates, also potentially impacting atmospheric biomarkers and in some strong cases destroying ozone, an important element in shielding the Earth from radiation (Grenfell et al. 2012).

Planetary magnetospheres are important for shielding planets from potential atmospheric erosion and from high-energy particles, with unshielded planets orbiting M dwarfs losing their atmospheres in 1 Gyr in some cases (Zendejas, Segura & Raga 2010). The size of a magnetosphere of a planet is strongly affected by the host star, in particular its stellar wind (See et al. 2014) and magnetic fields (Vidotto et al. 2013). Variations in the stellar wind can interact strongly with a planet's atmosphere, stripping it or depositing heat and gravity waves (Cohen et al. 2014, 2015). Considering the local

* E-mail: d.j.armstrong@warwick.ac.uk

- found lots of interesting papers on habitability
- most of them were focused around very few planets that the researchers had singled out as being special in certain ways
  - finding them would probably require machine learning algorithms as the data sets were very large.

a new dataset – habitable worlds catalogue

Simplified Catalog

These simplified tables of the Habitable Worlds Catalog (HWC) are easier to explore online. Columns can be sorted in ascending or descending order by clicking the headers. Search within these tables using the browser search function Ctrl+F (Windows, Linux, and Chrome OS) or ⌘-F (Mac).

Table 1. List of the potentially habitable exoplanets in the HWC, including the conservative and optimistic samples. They are sorted by the Earth Similarity Index (ESI).

	Name	Type	Detection Method	Mass (M_E)	Radius (R_E)	Flux (S_E)	T_{surf} (K)	Period (days)	Distance (ly)	Age (Gy)	ESI
1	Teegarden's Star b	M Warm Terran	Radial Velocity	≥ 1.16	~ 1.05	1.08	~ 293	4.91	12.5	> 8.00	0.97
2	TOI-700 d	M Warm Terran	Transit	~ 1.25	1.07	0.86	~ 276	37.4	101	> 1.50	0.94
3	Kepler-1649 c	M Warm Terran	Transit	~ 1.20	1.06	1.23	~ 302	19.5	300		0.93
4	TOI-700 e	M Warm Terran	Transit	~ 0.82	0.95	1.28	~ 305	27.8	101	> 1.50	0.91
5	TRAPPIST-1 d	M Warm Subterran	Transit	0.39	0.79	1.12	~ 295	4.05	40.5	> 0.50	0.91
6	LP 890-9 c	M Warm Terran	Transit	< 25.3	1.37	0.91	~ 280	8.46	105	7.20	0.89
7	K2-72 e	M Warm Terran	Transit	~ 2.21	1.29	1.30	~ 306	24.2	216		0.87
8	Proxima Cen b	M Warm Terran	Radial Velocity	≥ 1.07	~ 1.03	0.68	~ 261	11.2	4.2		0.86
9	GJ 1002 b	M Warm Terran	Radial Velocity	≥ 1.08	~ 1.03	0.67	~ 260	10.3	15.8		0.86
10	GJ 1061 d	M Warm Terran	Radial Velocity	≥ 1.64	~ 1.16	0.69	~ 246	13.0	12.0	> 7.00	0.86
11	GJ 1061 c	M Warm Terran	Radial Velocity	≥ 1.74	~ 1.18	1.45	~ 310	6.69	12.0	> 7.00	0.86
12	Ross 128 b	M Warm Terran	Radial Velocity	≥ 1.40	~ 1.11	1.48	~ 316	9.87	11.0	> 5.00	0.86
13	GJ 273 b	M Warm Terran	Radial Velocity	≥ 2.89	~ 1.51	1.06	~ 291	18.6	12.3		0.85
14	Kepler-296 e	M Warm Terran	Transit	~ 2.96	1.53	1.00	~ 281	34.1	544	4.20	0.85
15	Wolf 1069 b	M Warm Terran	Radial Velocity	≥ 1.26	~ 1.08	0.65	~ 258	15.6	31.3		0.85
16	TRAPPIST-1 e	M Warm Terran	Transit	0.69	0.92	0.65	~ 257	6.10	40.5	> 0.50	0.85
17	Kepler-442 b	K Warm Terran	Transit	~ 2.36	1.34	0.70	~ 263	112	1193	2.90	0.84
18	Kepler-62 e	K Warm Superterran	Transit	< 36.0	1.61	1.15	~ 297	122	981	7.00	0.83
19	Kepler-452 b	G Warm Superterran	Transit	~ 3.29	1.63	1.11	~ 295	384	1799	6.00	0.83
20	Kepler-1652 b	M Warm Terran	Transit	~ 3.19	1.60	0.84	~ 275	38.1	821	3.20	0.83
21	K2-3 d	M Warm Terran	Transit	< 2.20	1.46	1.45	~ 315	44.6	143	6.90	0.81
22	TOI-715 b	M Warm Terran	Transit	~ 3.02	1.55	0.71	~ 264	19.3	138	6.20	0.81
23	Wolf 1061 c	M Warm Superterran	Radial Velocity	≥ 3.41	~ 1.66	1.30	~ 306	17.9	14.0		0.80
24	Kepler-1410 b	K Warm Superterran	Transit	~ 3.82	1.78	1.07	~ 292	60.9	1196	4.07	0.80

Table 2. List of all the exoplanets in the HWC. Only those warm subterran, terran, or superterran exoplanets are considered potentially habitable. They are sorted by the Earth Similarity Index (ESI). In general, potentially habitable exoplanets have an ESI > 0.5, but this is not always true since the habitability criteria are independent of the ESI.

	Name	Type	Detection Method	Mass (M_E)	Radius (R_E)	Flux (S_E)	T_{surf} (K)	Period (days)	Distance (ly)	Age (Gy)	ESI
1	Teegarden's Star b	M Warm Terran	Radial Velocity	≥ 1.16	~ 1.05	1.08	~ 293	4.91	12.5	> 8.00	0.97
2	TOI-700 d	M Warm Terran	Transit	~ 1.25	1.07	0.86	~ 276	37.4	101	> 1.50	0.94
3	Kepler-1649 c	M Warm Terran	Transit	~ 1.20	1.06	1.23	~ 302	19.5	300		0.93
4	TOI-700 e	M Warm Terran	Transit	~ 0.82	0.95	1.28	~ 305	27.8	101	> 1.50	0.91
5	TRAPPIST-1 d	M Warm Subterran	Transit	0.39	0.79	1.12	~ 295	4.05	40.5	> 0.50	0.91
6	LP 890-9 c	M Warm Terran	Transit	< 25.3	1.37	0.91	~ 280	8.46	105	7.20	0.89

https://phl.upr.edu/hwc/data

- found this from the article i showed on the other slide
- used it since it had this system that rated exoplanets on their habitability probability:
  - 0 if inhabitable (no likelihood)
  - 1 if conservatively habitable (this is most likely)
  - 2 if optimistically habitable (this is less likely)
- not quite sure how exactly they found these values (despite looking through the about page)
- **full catalogue – all exoplanets** encompasses 5599 entries across 118 columns (features)

machine learning – training model to identify habitable planets

- cleared any columns with ≥ 40% missing values

```
exoplanets_data = exoplanets_data.drop(['S_NAME_HD', 'S_NAME_HIP', 'P_OMEGA_ERROR_MAX', 'P_OMEGA_ERROR_MIN', 'P_ECCENTRICITY_ERROR_MIN',
                                        'P_ECCENTRICITY_ERROR_MAX', 'P_OMEGA', 'P_INCLINATION_ERROR_MAX', 'P_INCLINATION_ERROR_MIN',
                                        'S_TYPE', 'P_TEMP_SURF', 'P_MASS_ERROR_MIN', 'P_MASS_ERROR_MAX', 'P_SEMI_MAJOR_AXIS_ERROR_MIN',
                                        'P_SEMI_MAJOR_AXIS_ERROR_MAX', 'S_LOG_LUM_ERROR_MIN', 'S_LOG_LUM_ERROR_MAX'],
                                       axis = 1)
```

- if data field is categorical, used the mode

```
exoplanets_data['P_TYPE_TEMP'] = exoplanets_data['P_TYPE_TEMP'].fillna(exoplanets_data['P_TYPE_TEMP'].mode()[0])
exoplanets_data['S_TYPE_TEMP'] = exoplanets_data['S_TYPE_TEMP'].fillna(exoplanets_data['S_TYPE_TEMP'].mode()[0])
exoplanets_data['P_TYPE'] = exoplanets_data['P_TYPE'].fillna(exoplanets_data['P_TYPE'].mode()[0])
```

- imputed numerical values using mice imputer
    - looks @ all data to find reasonable missing points

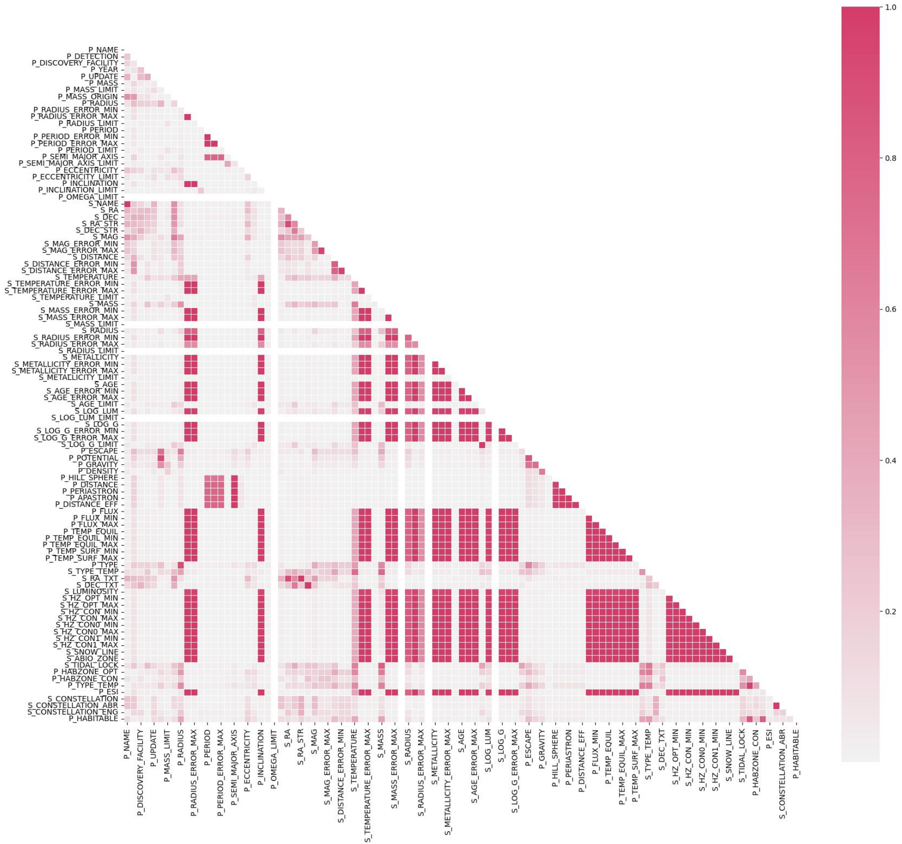
```
imputed_exoplanets_data = exoplanets_data.copy(deep = True)
mice_imputer = IterativeImputer()
imputed_exoplanets_data.iloc[:, :] = mice_imputer.fit_transform(exoplanets_data)
imputed_exoplanets_data.head(5)
```

- fixed imbalanced probability ratio with SMOTEEN over + down sampling

```
features_to_balance, labels_to_balance = imputed_exoplanets_data.drop(['P_HABITABLE'], axis = 1), imputed_exoplanets_data.P_HABITABLE

smoteenn = SMOTEENN(random_state=0)
balanced_features, balanced_labels = smoteenn.fit_resample(features_to_balance, labels_to_balance)

habitability_status_counts = Counter(balanced_labels)
for habitability_status_label, habitability_status_count in habitability_status_counts.items():
    habitability_status_percentage = (habitability_status_count / len(balanced_labels)) * 100
    print('habitability_status=%d, count=%d (%.3f%%)' % (habitability_status_label, habitability_status_count, habitability_status_percentage))
```

machine learning – training model to identify habitable planets

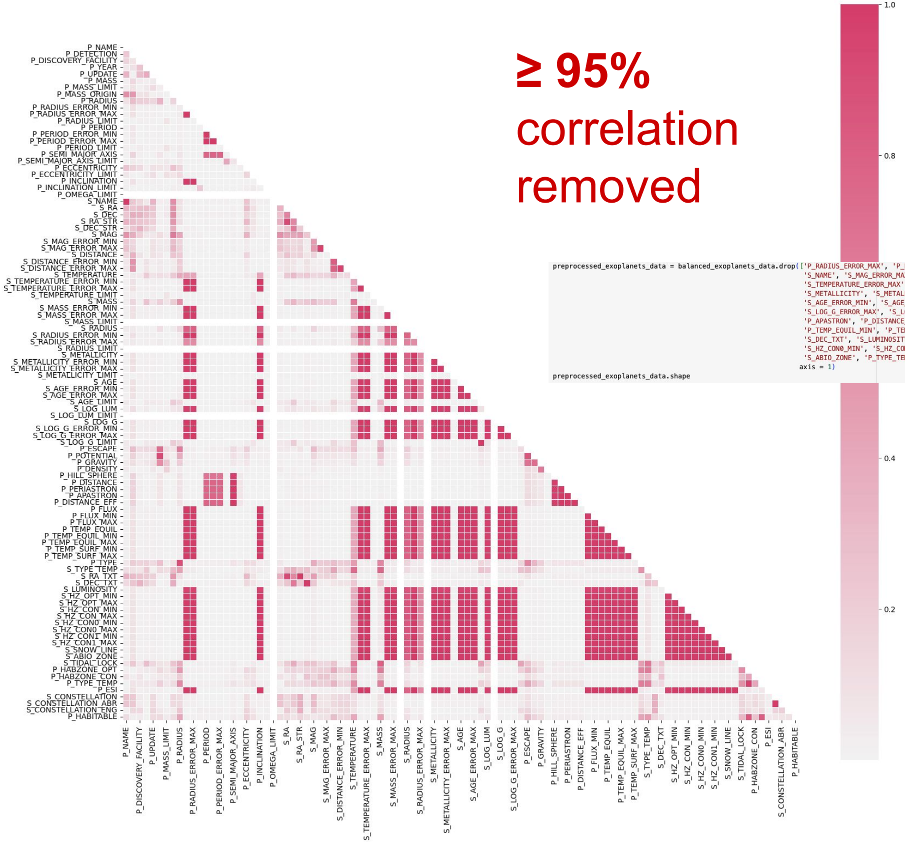
- looked @ correlation – how much do certain quantities affect each other?




P_NAME
P_DETECTION
P_DISCOVERY_FACILITY
P_YEAR
P_UPDATE
P_MASS
P_MASS_LIMIT
P_MASS_ORIGIN
P_RADIUS
P_RADIUS_ERROR_MIN
P_RADIUS_ERROR_MAX
P_RADIUS_LIMIT
P_PERIOD
P_PERIOD_ERROR_MIN
P_PERIOD_ERROR_MAX
P_PERIOD_LIMIT
P_SEMI_MAJOR_AXIS
P_SEMI_MAJOR_AXIS_LIMIT
P_ECCENTRICITY
P_ECCENTRICITY_LIMIT
P_INCLINATION
P_INCLINATION_LIMIT
P_OMEGA_LIMIT
S_NAME
S_RA
S_DEC
S_RA_STR
S_DEC_STR
S_MAG
S_MAG_ERROR_MIN
S_MAG_ERROR_MAX
S_DISTANCE
S_DISTANCE_ERROR_MIN
S_DISTANCE_ERROR_MAX
S_TEMPERATURE
S_TEMPERATURE_ERROR_MIN
S_TEMPERATURE_ERROR_MAX
S_TEMPERATURE_LIMIT
S_MASS
S_MASS_ERROR_MIN
S_MASS_ERROR_MAX
S_MASS_LIMIT
S_RADIUS
S_RADIUS_ERROR_MIN
S_RADIUS_ERROR_MAX
S_RADIUS_LIMIT
S_METALLICITY
S_METALLICITY_ERROR_MIN
S_METALLICITY_ERROR_MAX
S_METALLICITY_LIMIT
S_AGE
S_AGE_ERROR_MIN
S_AGE_ERROR_MAX
S_AGE_LIMIT
S_LOG_LUM
S_LOG_LUM_LIMIT
S_LOG_G
S_LOG_G_ERROR_MIN
S_LOG_G_ERROR_MAX
S_LOG_G_LIMIT
P_ESCAPE
P_POTENTIAL
P_GRAVITY
P_DENSITY
P_HILL_SPHERE
P_DISTANCE
P_PERIASTRON
P_APASTRON
P_DISTANCE_EFF
P_FLUX
P_FLUX_MIN
P_FLUX_MAX
P_TEMP_EQUIL
P_TEMP_EQUIL_MIN
P_TEMP_EQUIL_MAX
P_TEMP_SURF_MIN
P_TEMP_SURF_MAX
P_TYPE
S_TYPE_TEMP
S_RA_TXT
S_DEC_TXT
S_LUMINOSITY
S_HZ_OPT_MIN
S_HZ_OPT_MAX
S_HZ_CON_MIN
S_HZ_CON_MAX
S_HZ_CON0_MIN
S_HZ_CON0_MAX
S_HZ_CON1_MIN
S_HZ_CON1_MAX
S_SNOW_LINE
S_ABIO_ZONE
S_TIDAL_LOCK
P_HABZONE_OPT
P_HABZONE_CON
P_TYPE_TEMP
P_ESI
S_CONSTELLATION
S_CONSTELLATION_ABR
S_CONSTELLATION_ENG
P_HABITABLE
1.0
0.8
0.6
0.4
0.2

machine learning – training model to identify habitable planets

- looked @ correlation – how much do certain quantities affect each other?



≥ 95%
correlation
removed

```
preprocessed_exoplanets_data = balanced_exoplanets_data.drop(['P_RADIUS_ERROR_MAX', 'P_PERIOD_ERROR_MIN', 'P_PERIOD_ERROR_MAX', 'P_INCLINATION',
                                                              'S_NAME', 'S_MAG_ERROR_MAX', 'S_DISTANCE_ERROR_MAX', 'S_TEMPERATURE_ERROR_MIN',
                                                              'S_TEMPERATURE_ERROR_MAX', 'S_MASS_ERROR_MIN', 'S_MASS_ERROR_MAX', 'S_RADIUS_ERROR_MIN',
                                                              'S_METALLICITY', 'S_METALLICITY_ERROR_MIN', 'S_METALLICITY_ERROR_MAX', 'S_AGE',
                                                              'S_AGE_ERROR_MIN', 'S_AGE_ERROR_MAX', 'S_LOG_LUM', 'S_LOG_G', 'S_LOG_G_ERROR_MIN',
                                                              'S_LOG_G_ERROR_MAX', 'S_LOG_G_LIMIT', 'P_HILL_SPHERE', 'P_DISTANCE', 'P_PERIASTRON',
                                                              'P_APASTRON', 'P_DISTANCE_EFF', 'P_FLUX', 'P_FLUX_MIN', 'P_FLUX_MAX', 'P_TEMP_EQUIL',
                                                              'P_TEMP_EQUIL_MIN', 'P_TEMP_EQUIL_MAX', 'P_TEMP_SURF_MIN', 'P_TEMP_SURF_MAX', 'S_RA_TXT',
                                                              'S_DEC_TXT', 'S_LUMINOSITY', 'S_HZ_OPT_MIN', 'S_HZ_OPT_MAX', 'S_HZ_CON_MIN', 'S_HZ_CON_MAX',
                                                              'S_HZ_CON0_MIN', 'S_HZ_CON0_MAX', 'S_HZ_CON1_MIN', 'S_HZ_CON1_MAX', 'S_SNOW_LINE',
                                                              'S_ABIO_ZONE', 'P_TYPE_TEMP', 'P_ESI', 'S_CONSTELLATION_ABR'],
                                                             axis = 1)
preprocessed_exoplanets_data.shape
```

machine learning – training model to identify habitable planets

- random Forest to find the 15 most important features:

```
features = preprocessed_exoplanets_data[['P_MASS', 'P_RADIUS', 'P_PERIOD', 'P_SEMI_MAJOR_AXIS', 'S_TEMPERATURE', 'S_MASS', 'S_RADIUS',
                                         'P_ESCAPE', 'P_POTENTIAL', 'P_DENSITY', 'P_TYPE', 'S_TIDAL_LOCK', 'P_HABZONE_OPT', 'S_TYPE_TEMP',
                                         'P_HABZONE_CON']]
```

machine learning – training model to identify habitable planets

- random Forest to find the 15 most important features:

```
features = preprocessed_exoplanets_data[['P_MASS', 'P_RADIUS', 'P_PERIOD', 'P_SEMI_MAJOR_AXIS', 'S_TEMPERATURE', 'S_MASS', 'S_RADIUS',
                                         'P_ESCAPE', 'P_POTENTIAL', 'P_DENSITY', 'P_TYPE', 'S_TIDAL_LOCK', 'P_HABZONE_OPT', 'S_TYPE_TEMP',
                                         'P_HABZONE_CON']]
```

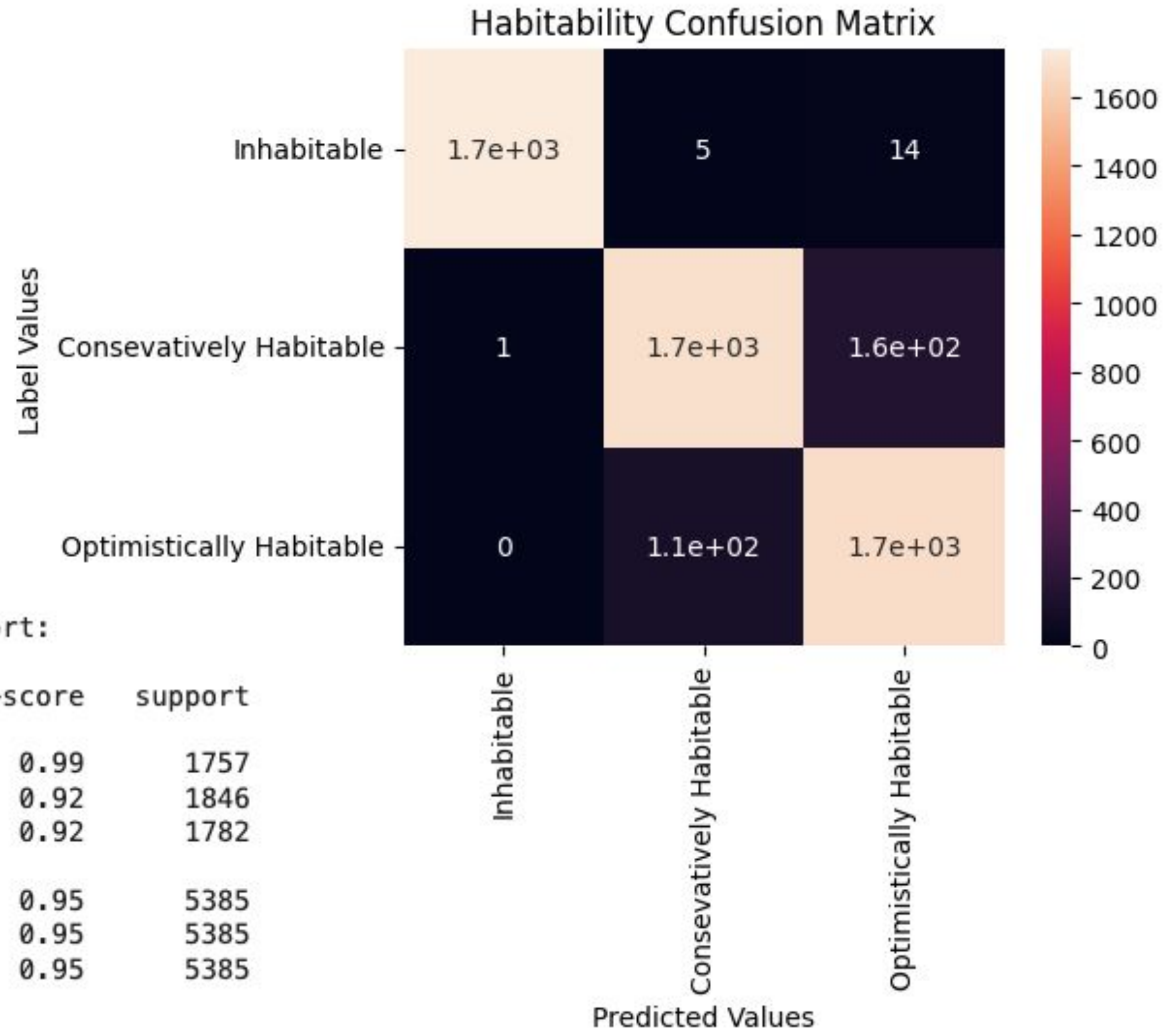
- scaled the values:

```
minMaxScaler = MinMaxScaler()
features_train = minMaxScaler.fit_transform(features_train)
features_test = minMaxScaler.fit_transform(features_test)
```

- 3 models:
    - KNN
    - decision tree
    - random forest

KNN classifier

- k = 3 (currently no adjustments to this value; could go search for it though)



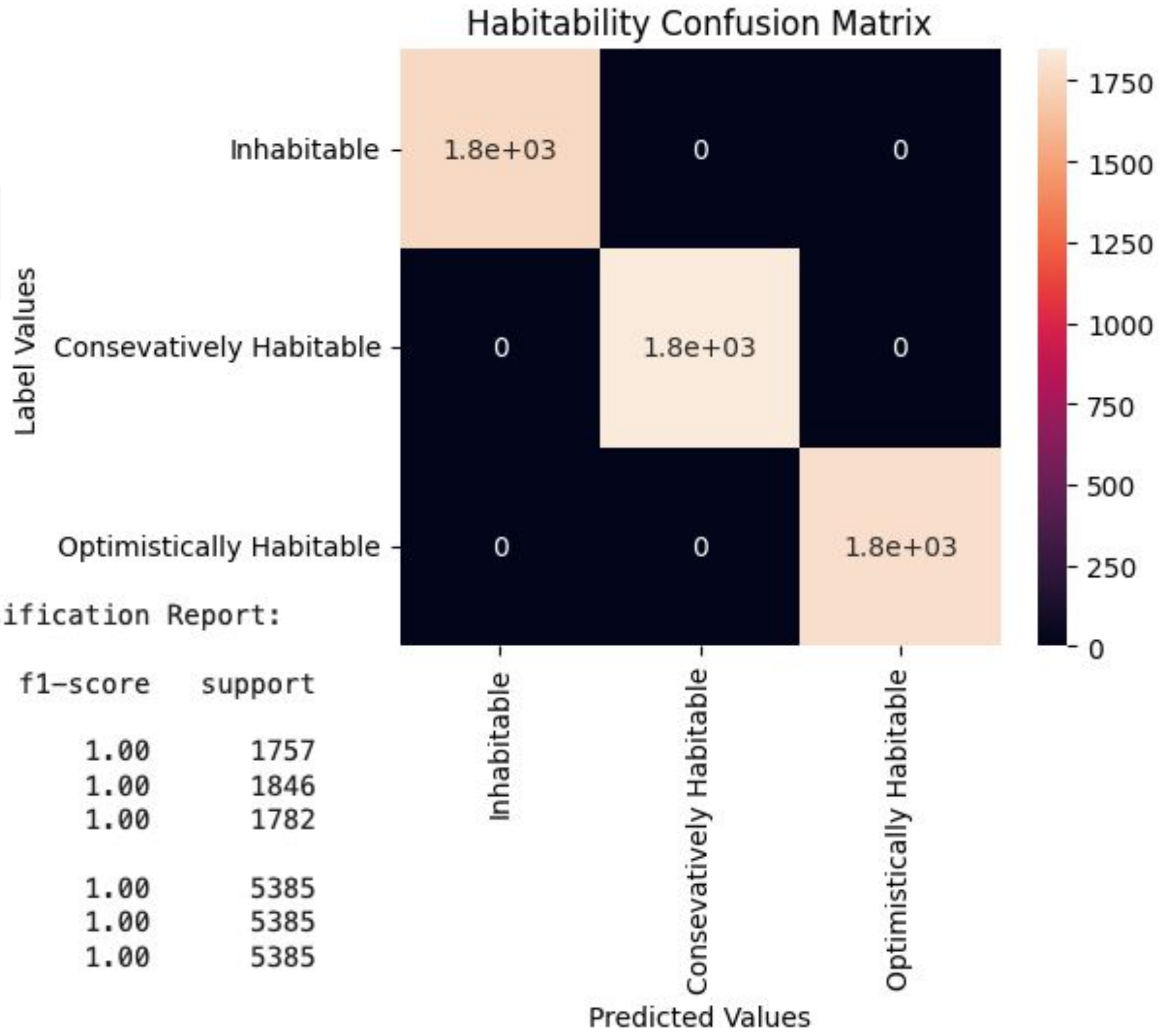
KNN Classififier - Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	1757
1	0.94	0.91	0.92	1846
2	0.91	0.94	0.92	1782
accuracy			0.95	5385
macro avg	0.95	0.95	0.95	5385
weighted avg	0.95	0.95	0.95	5385

decision tree

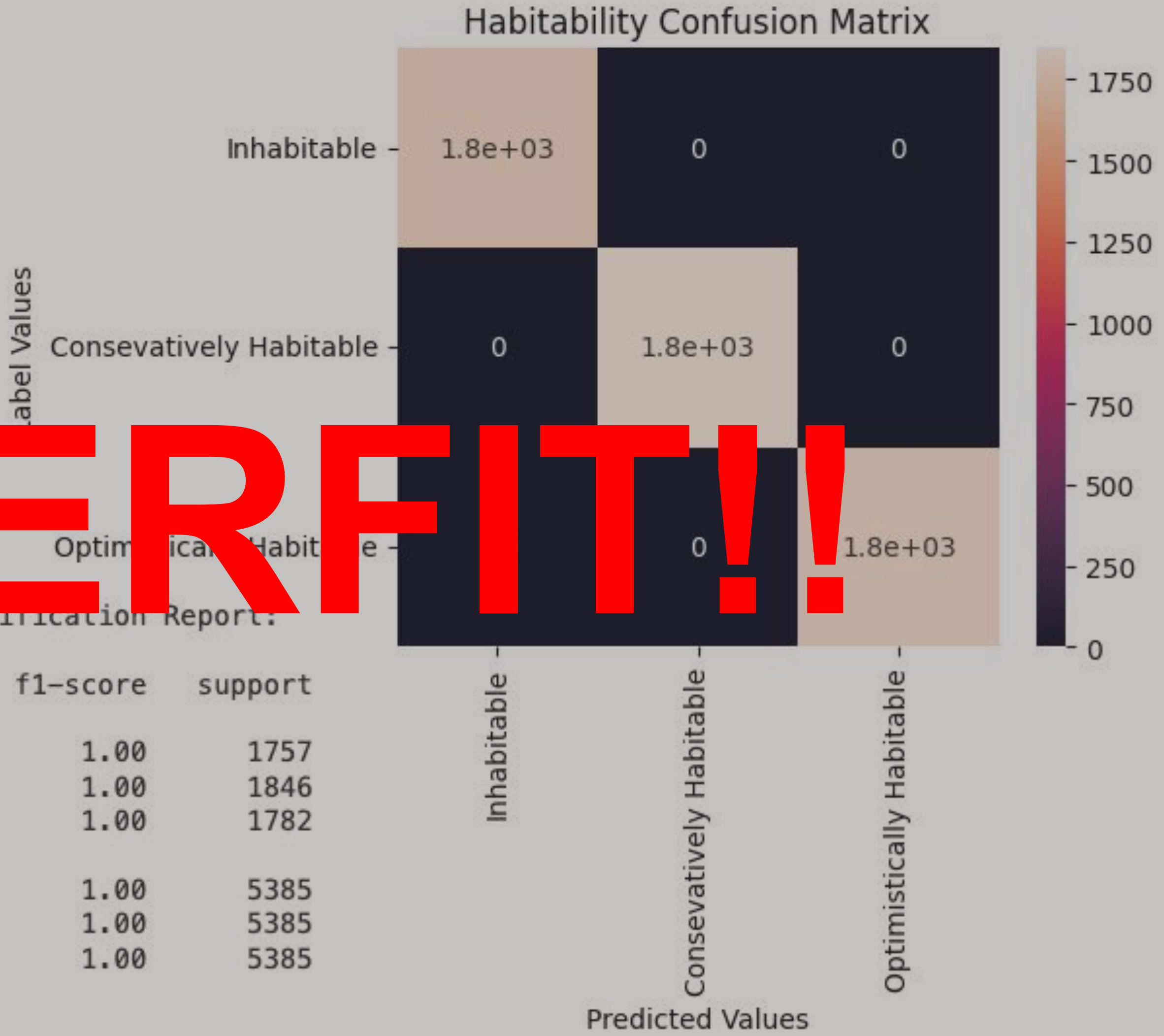
```
startTime = time.time()

decision_tree_classifier = DecisionTreeClassifier(criterion = 'gini',
                                                  splitter = 'best',
                                                  max_depth = 3,
                                                  random_state = 0,
                                                  max_leaf_nodes = 12).fit(features_train, labels_train)
```

```
Decision Tree Classifier - Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1757
           1       1.00      1.00      1.00      1846
           2       1.00      1.00      1.00      1782

    accuracy                           1.00      5385
   macro avg       1.00      1.00      1.00      5385
weighted avg       1.00      1.00      1.00      5385
```



decision tree

```
startTime = time.time()

decision_tree_classifier = DecisionTreeClassifier(criterion = 'gini',
                                                  splitter = 'best',
                                                  max_depth = 3,
                                                  random_state = 0,
                                                  max_leaf_nodes = 12).fit(features_train, labels_train)
```



OVERFIT!!

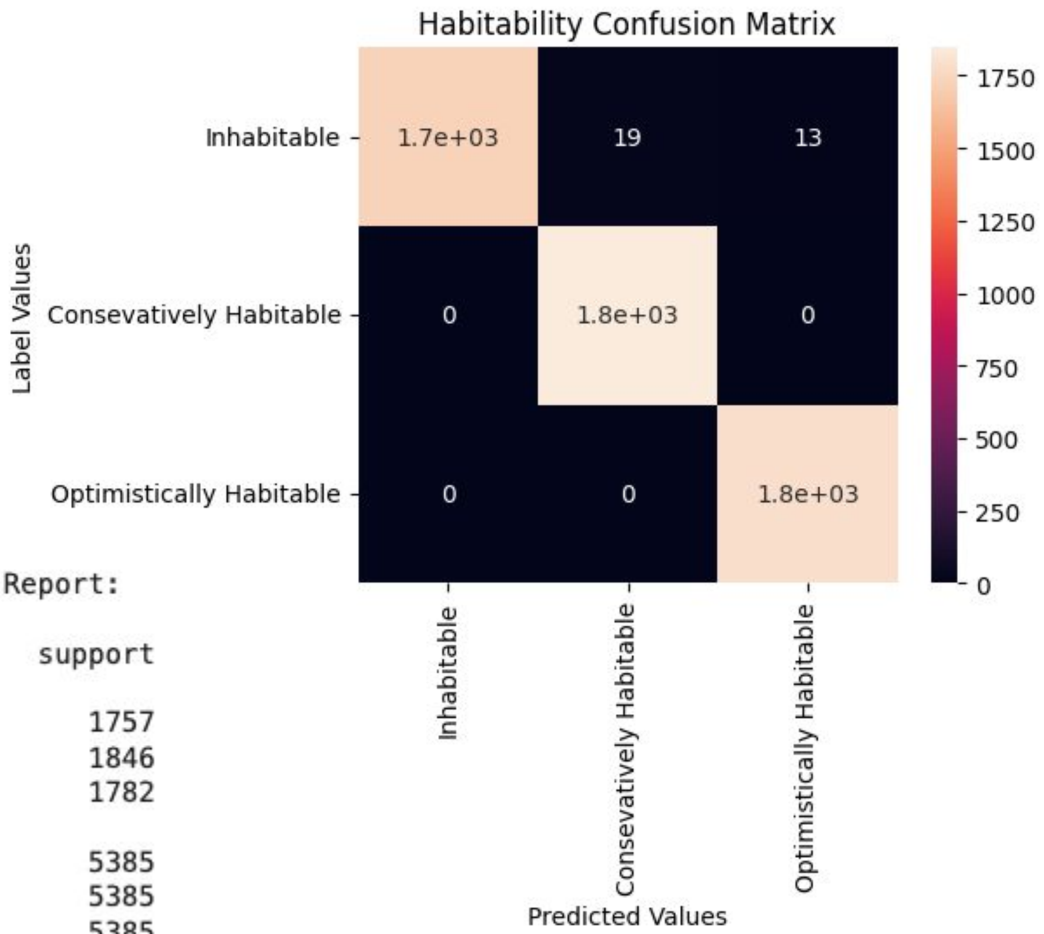
```
Decision Tree Classififier - Classification Report:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1757
           1       1.00      1.00      1.00      1846
           2       1.00      1.00      1.00      1782

    accuracy                           1.00      5385
   macro avg       1.00      1.00      1.00      5385
weighted avg       1.00      1.00      1.00      5385
```

random forest

```
param_grid = {'n_estimators': np.arange(10, 320, 50),
              'criterion': ['gini', 'entropy', 'log_loss'],
              'max_depth': np.arange(1, 15, 1),
              'max_features': ['log2', 'sqrt']}

grid_search = GridSearchCV(RandomForestClassifier(),
                           param_grid = param_grid,
                           refit = True,
                           verbose = 2)

grid_search.fit(features_train, labels_train)

print(f"Random Forest Classifier Best Parameters: {grid_search.best_params_}")
```

Show hidden output

```
startTime = time.time()

random_forest_classifier = RandomForestClassifier(n_estimators = 100,
                                                  criterion = 'gini',
                                                  max_depth = 3,
                                                  max_features = 'sqrt',
                                                  random_state = 0,
                                                  max_leaf_nodes = 12).fit(features_train, labels_train)
```



Random Forest Classifier - Classification Report:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	1757
1	0.99	1.00	0.99	1846
2	0.99	1.00	1.00	1782
accuracy			0.99	5385
macro avg	0.99	0.99	0.99	5385
weighted avg	0.99	0.99	0.99	5385

overall:

- out of the three classifiers, i’d believe the **KNN classifier** is currently the most accurate model of the data
  - decision tree model (and the random forest to a certain extent) seem overfitted (a little too perfect to be true)
- i’m also hoping to also use this model (maybe with a few tweaks) with the new dataset you mentioned coming out next january!!