project 2

one stellar classification

Team 2 to Tango

Niharika, Hazel, Azlan, Peter and Enrique

purpose

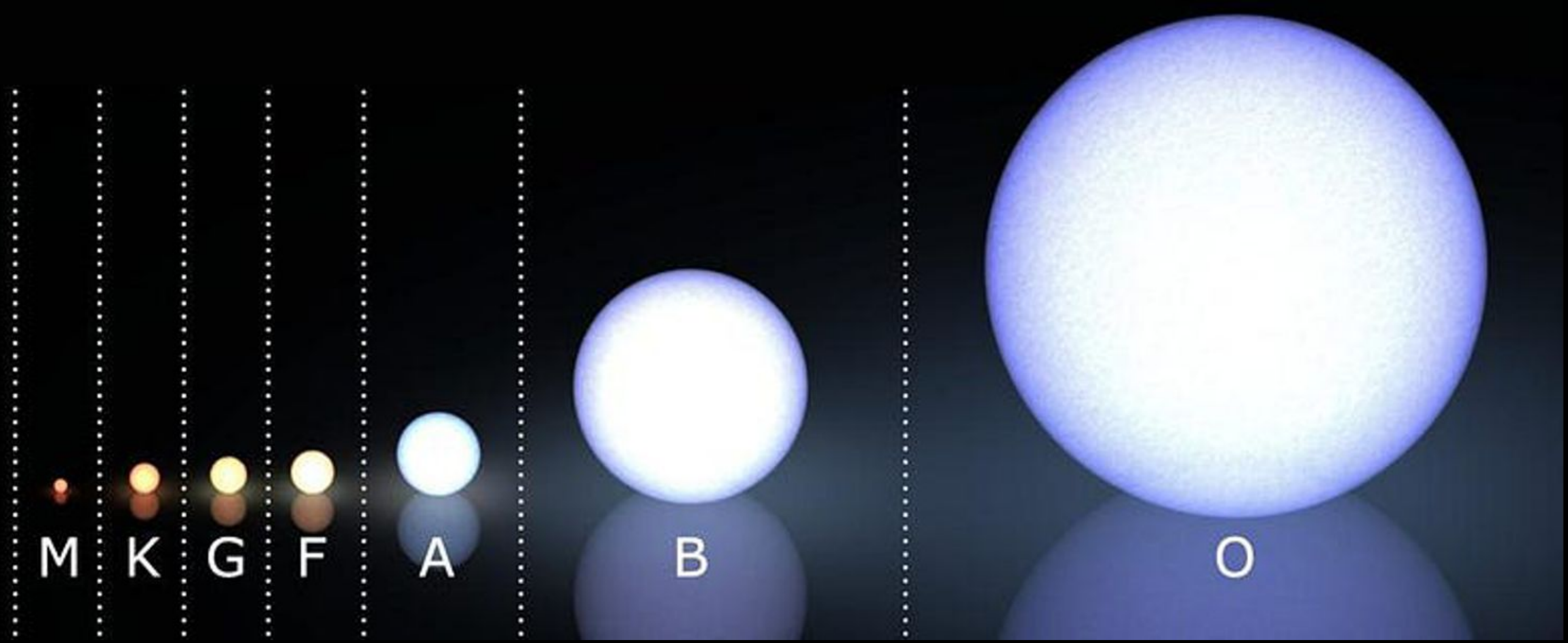
We are living in an era in which space exploration has dramatically increased. Curiosity about the universe beyond our solar system is in the minds of governments officials, private companies CEOs, and the Earth's population is exposed to frequent news about all sorts of outer space missions. We chose a stellar classification dataset to better understand the lifecycle of stars including our own sun. Machine learning can help us

predict the type of a star

based on features such as **Visual Apparent Magnitude**, **Distance Between the Star and the Earth**, **Color Index**, **and Spectral Type** the team explored a Star Dataset and implemented a ML model while building two pipelines: one to preprocess data and one to make predictions.

Additionally, team 2 utilized custom formulas to create **'new features'** and fine tuned the model for greater accuracy.

Vmag - Visual Apparent Magnitude of the Star
Plx - Distance Between the Star and the Earth
B-V color index – Hot star B-V close to 0 or negative, cool star has a B-V close to 2.0
SpType - Spectral type



Target Dwarfs [0] Giants [1]

spectral classification

dataset

Stars

NASA's Hubble Space Telescope has produced its largest mosaic image ever of the Triangulum galaxy (M33). This image spans 14,500 light-years.

Credits: NASA, ESA, and M. Durbin, J. Dalcanton, and B.F. Williams (University of Washington) dataset

dataset/Star39552_balanced.csv

	Vmag	Plx	e_Plx	B-V	SpType	Amag	TargetClass
0	10.00	31.66	6.19	1.213	K7V	22.502556	1
1	8.26	3.21	1.00	1.130	K0III	15.792525	0
2	8.27	12.75	1.06	0.596	F9V	18.797552	1
3	6.54	5.23	0.76	1.189	K1III	15.132508	0
4	8.52	0.96	0.72	0.173	B8V	13.431356	1
...	...	...	...	...	...	...	...
39547	5.83	0.17	0.52	0.474	B7Iab	6.982245	0
39548	7.05	18.12	0.92	0.424	F5V	18.340790	1
39549	9.21	3.89	1.46	0.227	A1IV	17.159748	1
39550	9.01	2.13	1.46	1.467	M5III	15.651898	0
39551	9.12	3.82	0.79	0.480	F5V	17.030317	1

39552 rows × 7 columns

dataset

dataset/Star39552_balanced.csv

	Vmag	Plx	e_Plx	B-V	SpType	Amag	TargetClass
0	10.00	31.66	6.19	1.213	K7V	22.502556	1
1	8.26	3.21	1.00	1.130	K0III	15.792525	0
2	8.27	12.75	1.06	0.596	F9V	18.797552	1
3	6.54	5.23	0.76	1.189	K1III	15.132508	0
4	8.52	0.96	0.72	0.173	B8V	13.431356	1
...	...	...	...	...	...	...	...
39547	5.83	0.17	0.52	0.474	B7Iab	6.982245	0
39548	7.05	18.12	0.92	0.424	F5V	18.340790	1
39549	9.21	3.89	1.46	0.227	A1IV	17.159748	1
39550	9.01	2.13	1.46	1.467	M5III	15.651898	0
39551	9.12	3.82	0.79	0.480	F5V	17.030317	1

39552 rows × 7 columns

dataset/Star99999_raw.csv

	Unnamed: 0	Vmag	Plx	e_Plx	B-V	SpType
0	0	9.10	3.54	1.39	0.482	F5
1	1	9.27	21.90	3.10	0.999	K3V
2	2	6.61	2.81	0.63	-0.019	B9
3	3	8.06	7.75	0.97	0.370	F0V
4	4	8.55	2.87	1.11	0.902	G8III
...	...	...	...	...	...	...
99994	99994	8.72	3.07	0.87	0.097	B3
99995	99995	9.25			0.131	A1V
99996	99996	8.08	1.07	0.68	1.094	G5
99997	99997	6.98	2.97	0.76	-0.143	B1.5V
99998	99998	8.51	-1.18	1.34	1.568	K5/M0III

99999 rows × 6 columns

existing features from original dataset

Vmag – Visual Apparent Magnitude of the Star
Plx – Distance Between the Star and the Earth
B-V color index – Hot star B-V close to 0 or negative,
– Cool star has a B-V close to 2.0
SpType – Spectral type

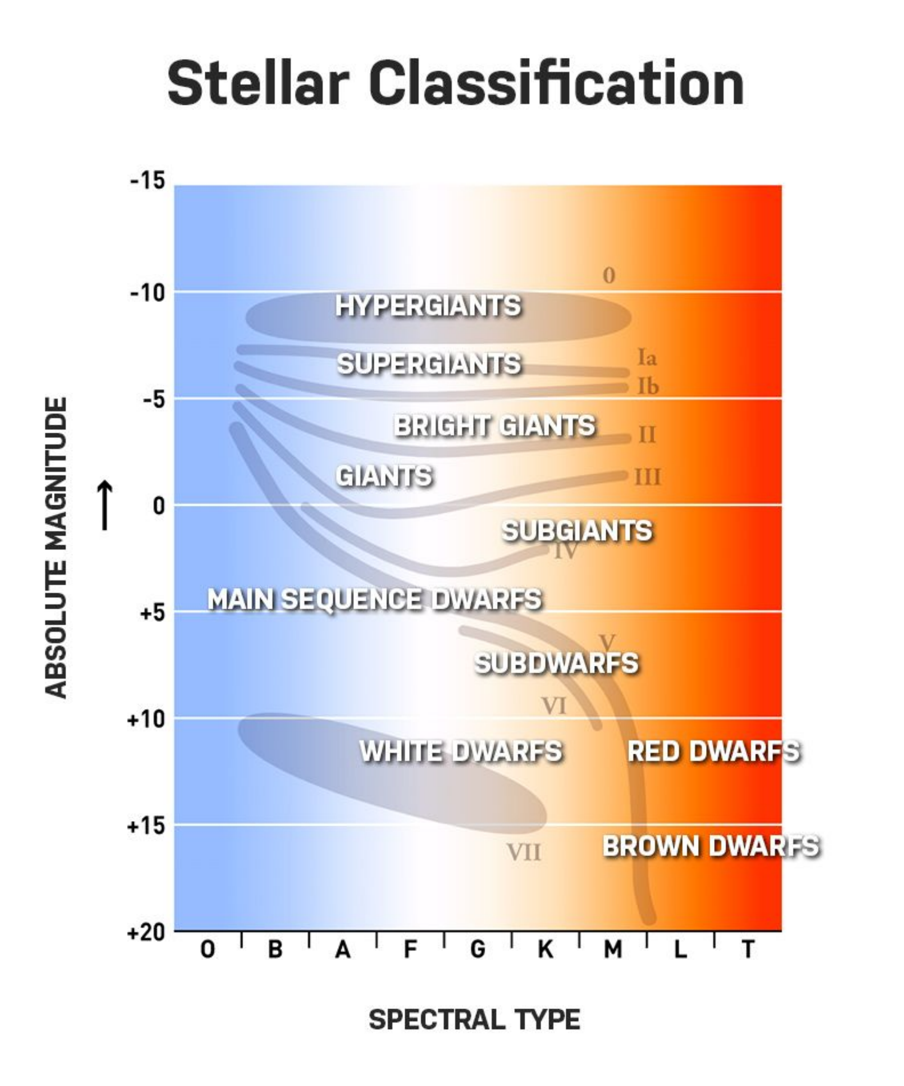
new features and calculations

Temperature
Distance (parsecs)
Distance (light years)
Amag
Luminosity (Sun=1)
Radius (Sun=1)
Plx (in arcsecs)

```
df["Plx"] = df["Plx"] / 1000
df["Distance (parsecs)"] = 1/df["Plx"]
df["Distance (light years)"] = abs(df["Distance (parsecs)"]) * 3.26156
df["Amag"] = df["Vmag"] + 5 * (np.log10(df["Plx"]) + 1)
df["Temperature (K)"] = 4600 * (1/(0.92*df["B-V"] + 1.7) + 1/(0.92*df["B-V"] + 0.62))
df["Luminosity (Sun=1)"] = 10**(0.4 * (4.85-df["Amag"]))
df["Radius (Sun=1)"] = np.sqrt(df["Luminosity (Sun=1)"]) * (5778 / df["Temperature (K)"])**2
df
```

added new features and recalculated target

new target




```python
@staticmethod
def star_type(spectral_type):
    if "VI" in spectral_type:
        return 0
    elif "IV" in spectral_type:
        return 1 # Subgiant
    elif "V" in spectral_type:
        return 0 # Main sequence
    elif ("III" in spectral_type
          or "II" in spectral_type
          or "Ib" in spectral_type
          or "Ia" in spectral_type
          or spectral_type[-1] == 0):
        return 1 # Giant
    elif "M" in spectral_type:
        return 0 # Brown dwarf
    else:
        return None
```

dropna

99999

[parsing features for numerical values]

96742 x 4 features

[add new features \ remove infinite values]

93556 x 11 features

[remove nulls after star classification]

48048 x 10 features + target

1st pipeline

from preprocess_pipeline import Preprocess

```
p = Preprocess("../Res/Star99999_raw.csv", verbose=True)
```

```
Converting numerical features to floats.
3257 rows dropped.
Calculating new features.
Replacing infinity values with NaN.
3186 rows dropped.
Classifying star type.
45508 rows dropped.
```

```
p.get_processed_df(numerical=True)
```

```
p.corr_heatmap()
```

```
p.get_df_without(["Distance (parsecs)", "Temperature (K)"])
```

```
p.get_df_without(["Distance (parsecs)", "B-V"])
```

```
p.save_csv("../Res/preporcessed.csv")
```

new features

augmented, balanced and preprocessed dataset

	Vmag	Plx	e_Plx	B-V	SpType	Distance (parsecs)	Distance (light years)	Amag	Temperature (K)	Luminosity (Sun=1)	Radius (Sun=1)	target
1	9.27	0.02190	3.10	0.999	K3V	45.662100	148.929680	5.972221	4745.140425	0.355723	0.884327	0.0
3	8.06	0.00775	0.97	0.370	F0V	129.032258	420.846452	2.506509	7044.130880	8.657582	1.979696	0.0
4	8.55	0.00287	1.11	0.902	G8III	348.432056	1136.432056	0.839409	4991.060700	40.200940	8.497428	1.0
5	12.31	0.01880	4.99	1.336	M0V:	53.191489	173.487234	8.680789	4058.107348	0.029355	0.347336	0.0
7	9.05	0.00517	1.95	1.102	M6e-M8.5e Tc	193.423598	630.862669	2.617453	4510.468347	7.816618	4.587977	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
99987	8.79	0.00089	1.28	1.194	K1III	1123.595506	3664.674157	-1.463050	4320.533599	335.135155	32.740876	1.0
99988	8.66	0.02804	2.25	1.008	M0	35.663338	116.318117	5.898890	4723.612188	0.380578	0.923057	0.0
99989	8.00	0.00041	0.92	0.854	F6Iab	2439.024390	7955.024390	-3.936081	5123.037777	3269.130719	72.730421	1.0
99992	7.69	0.00660	0.92	1.110	K2III	151.515152	494.175758	1.787720	4493.257892	16.784644	6.774675	1.0
99997	6.98	0.00297	0.76	-0.143	B1.5V	336.700337	1098.168350	-0.656218	12350.588581	159.399554	2.763270	0.0

48048 rows × 12 columns

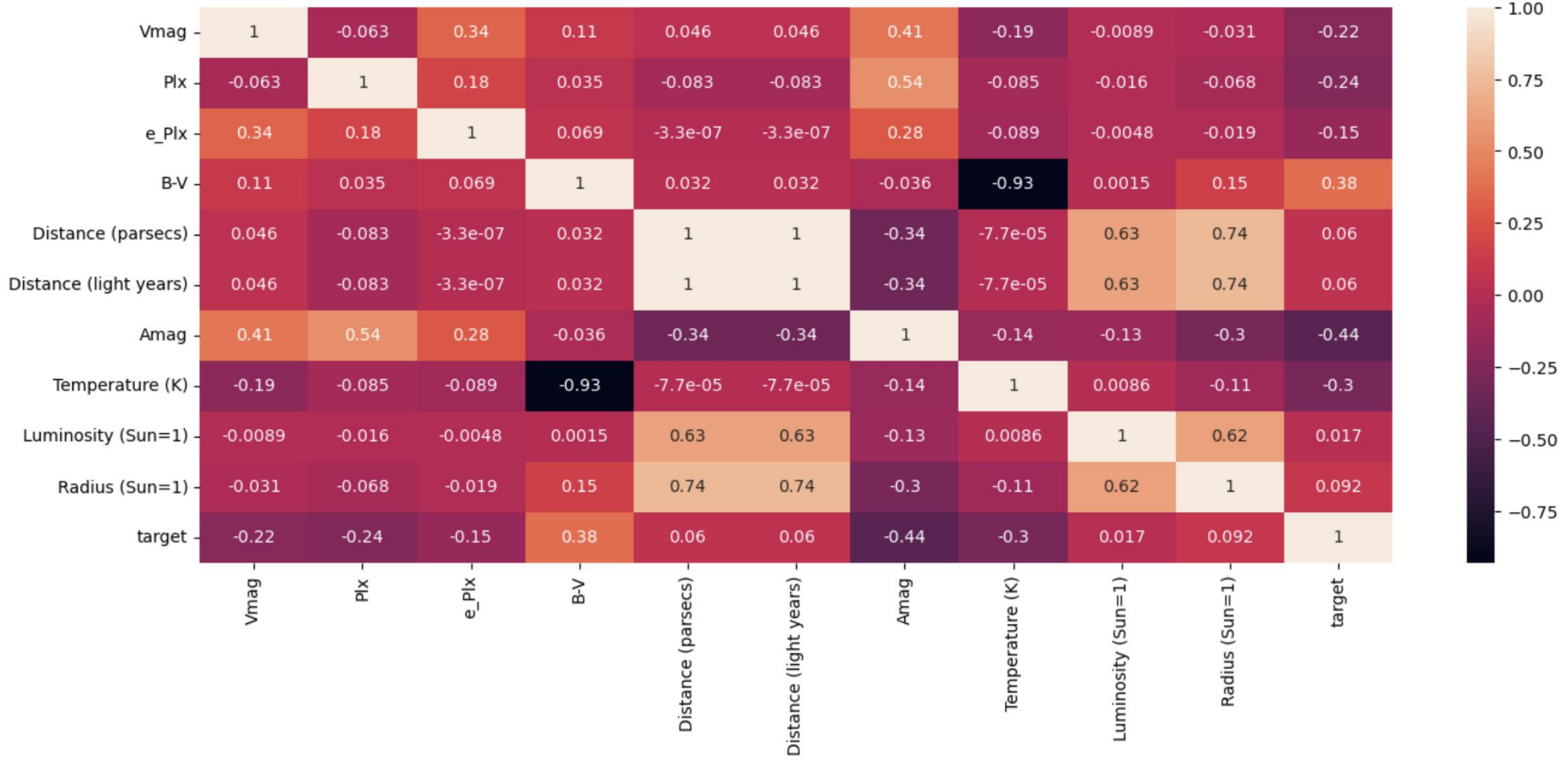
```
df["target"].value_counts()
```

```
target
1.0    25631
0.0    22417
Name: count, dtype: int64
```

```
df.to_csv("../Res/preporcessed.csv", index=False)
```

correlations

```python
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(numerical_df.corr(), annot=True)
```



data decisions

- BV and Temperature are highly negatively correlated at -0/93, we dropped BV
  - we experimented with dropping either/both, but dropping BV gives the best results

- distance (parsecs) and distance(light years) are repetitive we chose distance (light years) because it is more commonly used

machine learning models

- svc
- decision tree
- knn
- log regression
- random forest

rationale:

- binary target (0,1)
- choose to train with non-linear models

- comparing model results
- processed data results
- comparison: undersampling data
- comparison: pre-processed data

2^{nd} pipeline

```python
#svc
clfr = SVC(kernel="rbf", random_state=42)
clfr.fit(train_test_res[0], train_test_res[2])
svc_pred = clfr.predict(train_test_res[1])
print("SVC")
print(classification_report(train_test_res[3], svc_pred, labels = [1, 0]))
print("------------------------------------------------------------------------")

#svc resampled underfitting
clfr_svc_und = SVC(kernel="rbf", random_state=42)
clfr_svc_und.fit(train_test_res[4], train_test_res[5])
svc_und_pred = clfr_svc_und.predict(train_test_res[1])
print("SVC undersampled")
print(classification_report(train_test_res[3], svc_und_pred, labels = [1, 0]))
print("------------------------------------------------------------------------")
```

…

#svc

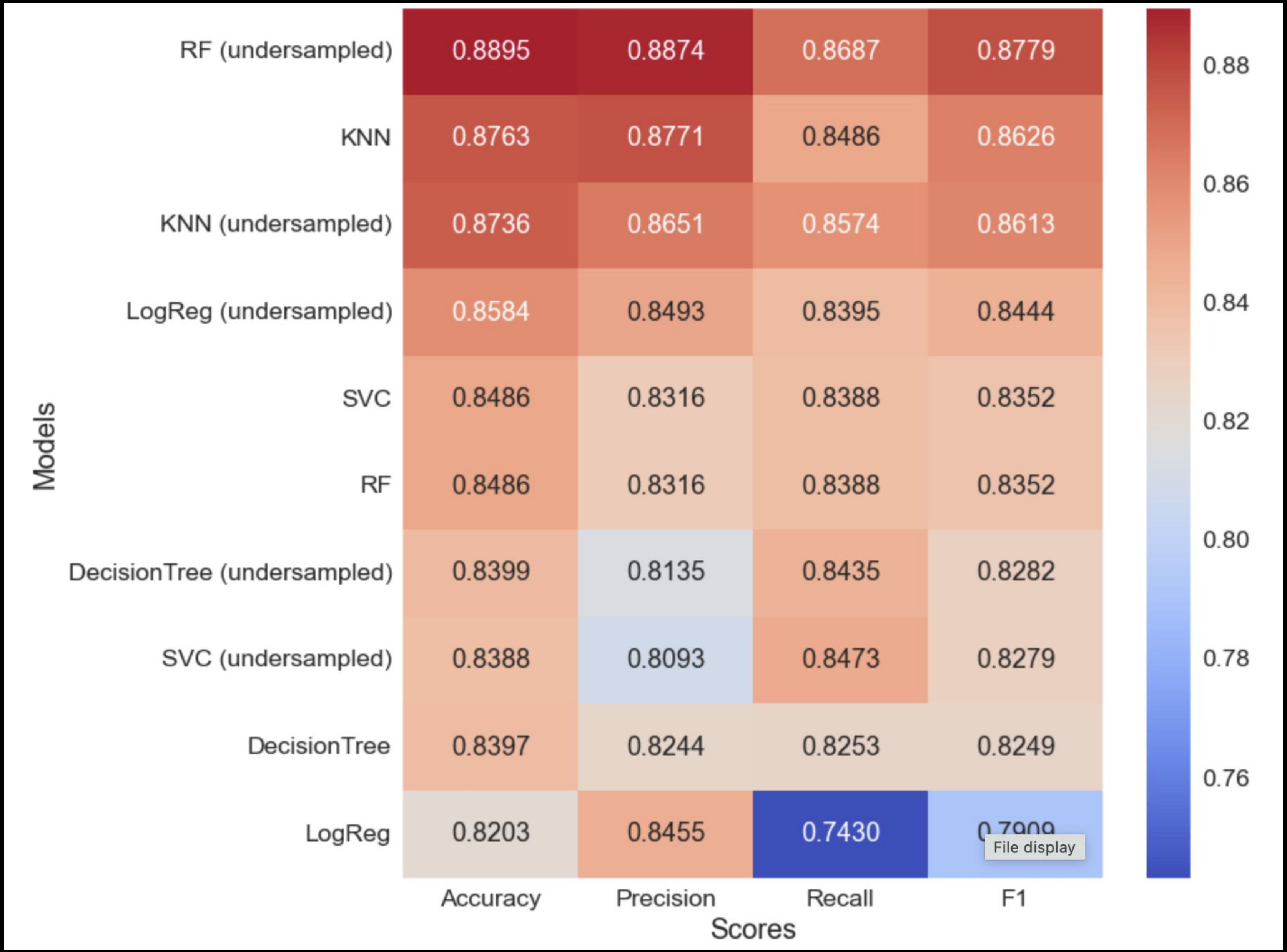
#decision tree

#knn

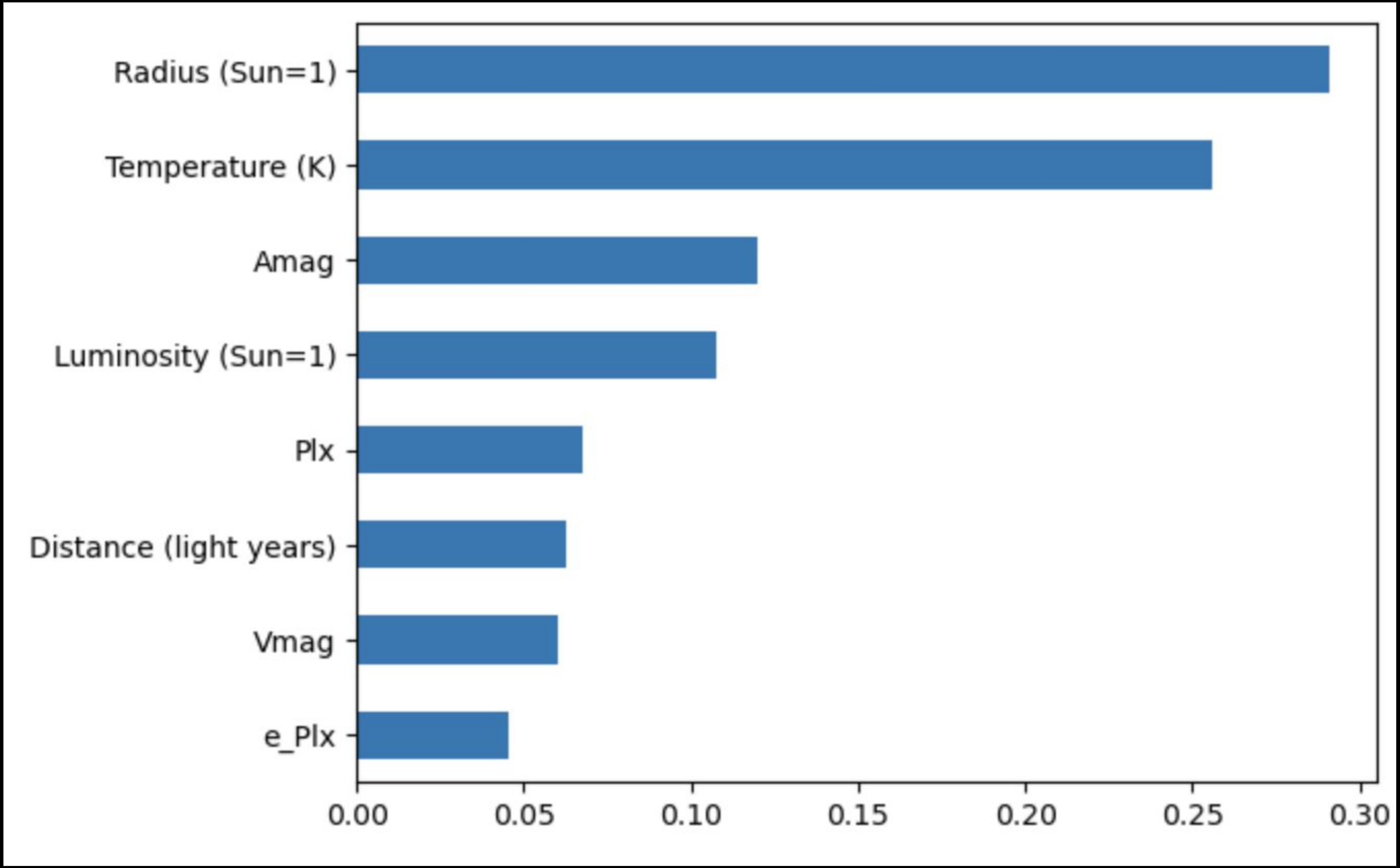
#log regression

#random forest

model performance

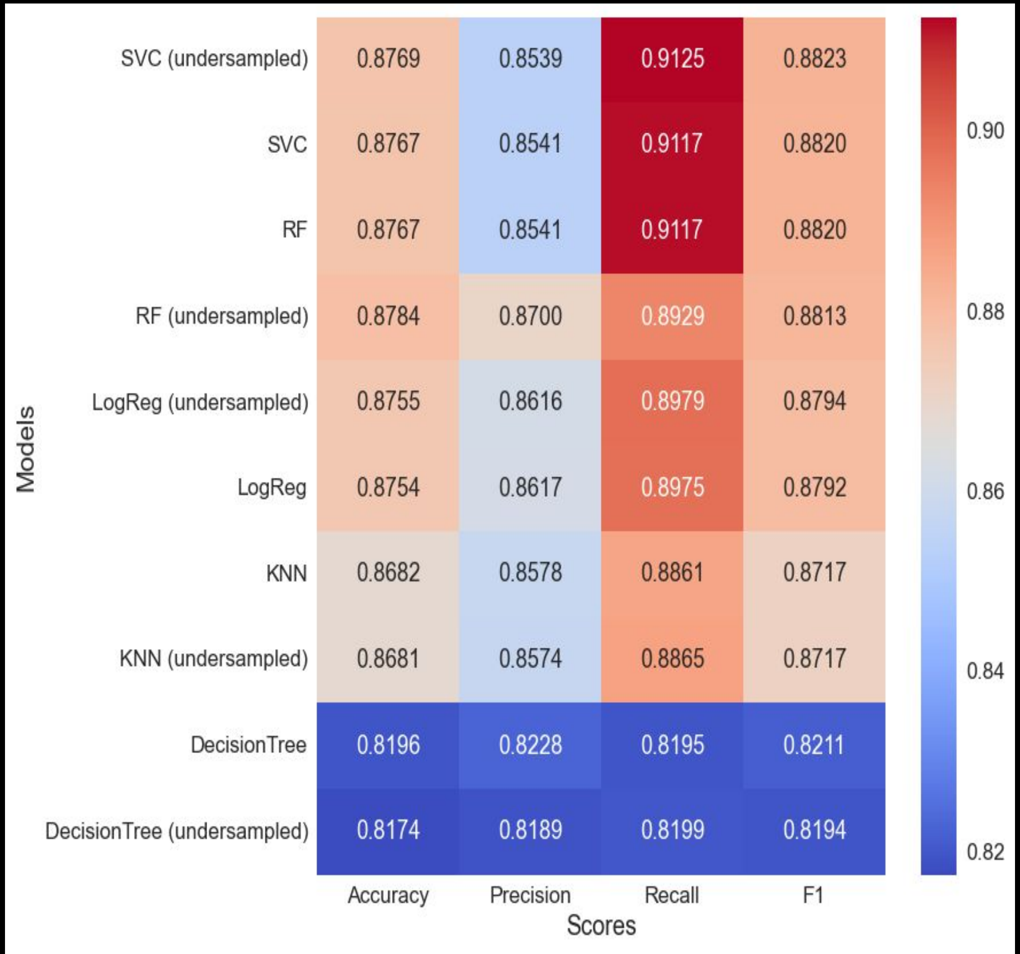


MVPs

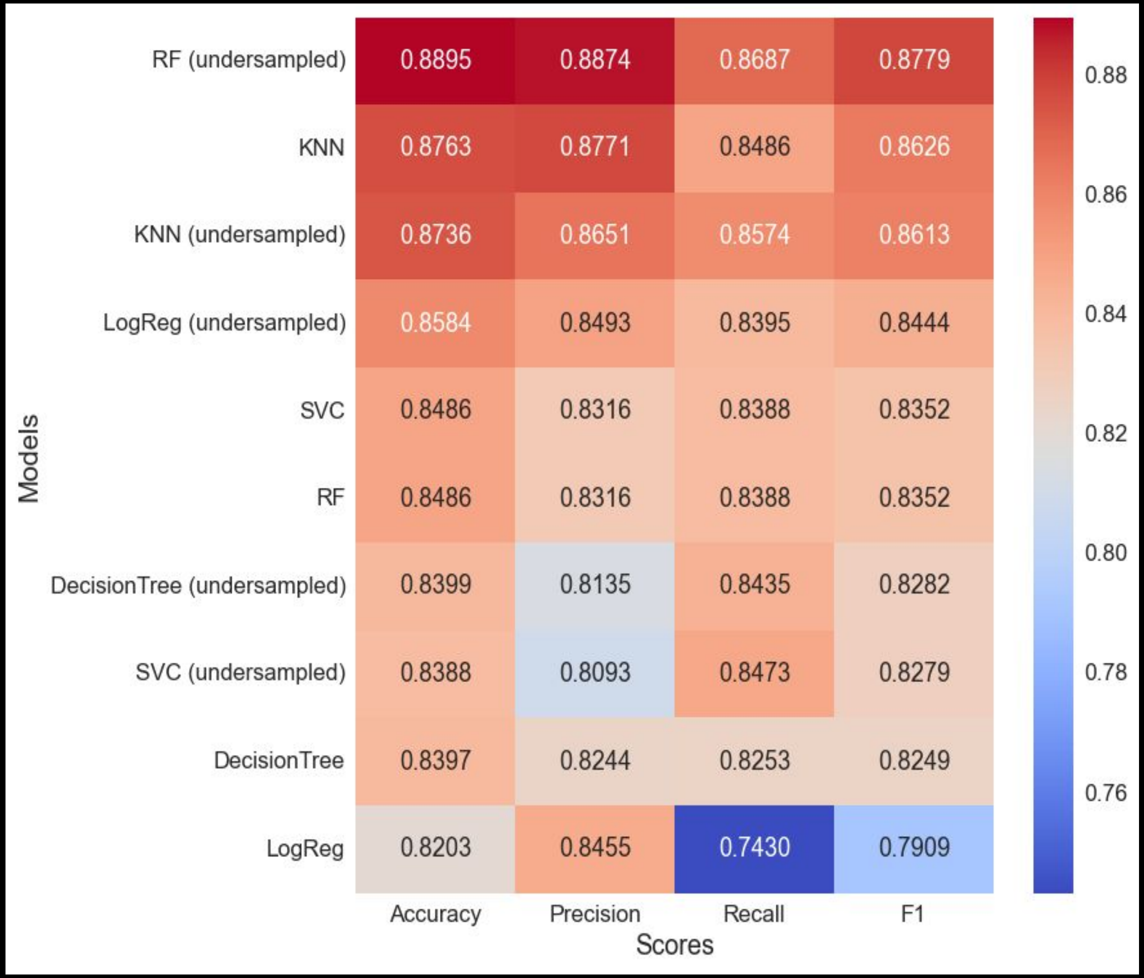

Radius (Sun=1)
Temperature (K)
Amag
Luminosity (Sun=1)
Plx
Distance (light years)
Vmag
e_Plx
0.00
0.05
0.10
0.15
0.20
0.25
0.30

comparisons

PreCleaned/ Classified Data



Our Version



conclusions

- ➔ Stellar classification is a critical step in understanding the evolutionary stage of the star - revealing crucial information about a star's life cycle
- ➔ Our Classification models was trained on available data and new features - simplified by the creation of cleansing and model variation pipelines
- ➔ Random Forest Classifier run on undersampled data , performed the best compared to logistic regression or some of the other models indicating likely non-linear relationships between the target and features
- ➔ The Radius of the star and Temperature are the most important features in classifying a star and contribute the most to predictivity

Citations

https://www.kaggle.com/datasets/vinesmsuic/star-categorization-giants-and-dwarfs/discussion/287630

FERNANDO JOSÉ SILVA LIMA FILHO · POSTED 3 YEARS AGO 3

The "Amag" column values are wrong !!

The unit of the "Plx" column is ** Milliarcsecond **, but it has been placed as arcsecond in the absolute magnitude formula (This can be verified by making an HR diagram).

https://itu.physics.uiowa.edu/glossary/stellar-parallax

The parallax of an object can be used to approximate the distance to an object using the formula:

$$D = \frac{1}{p}$$

https://physicsfeed.com/post/how-do-we-measure-distance-nearby-star-earth/

[1 pc ~ 3.26156 light-years ~ 63241.1 AU]

Citations

https://sites.uni.edu/morgans/astro/course/Notes/section3/math11.html

Magnitude - Distance Formula - used to give the relationship between the apparent magnitude, the absolute magnitude and the distance of objects.
Formula: **m - M = -5 + 5 Log (d)** where:

- m = apparent magnitude
- M = absolute magnitude
- d = distance measured in parsecs (pc)

https://sarahspolaor.faculty.wvu.edu/files/d/2cac9872-170f-4a59-893e-f69800e0d284/04_notes.pdf

$$T = 4600\ \mathrm{K}\left(\frac{1}{0.92(B-V)+1.7} + \frac{1}{0.92(B-V)+0.62}\right)$$

http://csep10.phys.utk.edu/OJTA2dev/ojta/c2c/ordinary_stars/magnitudes/absolute_tl.html

$$L / L_{\mathrm{sun}} = 10^{0.4(4.85\ -\ M)}$$

https://www.teachastronomy.com/textbook/Properties-of-Stars/Stefan-Boltzmann-Law/

$$R_*/R_\odot = \sqrt{}\ (L_*/L_\odot)\ /\ (T_*/T_\odot)^2$$