
“Gift of Orzhova”, art by Johannes Voss

Data Science/ML & **Magic the Gathering**

Presentation by:
Monica Fidalgo

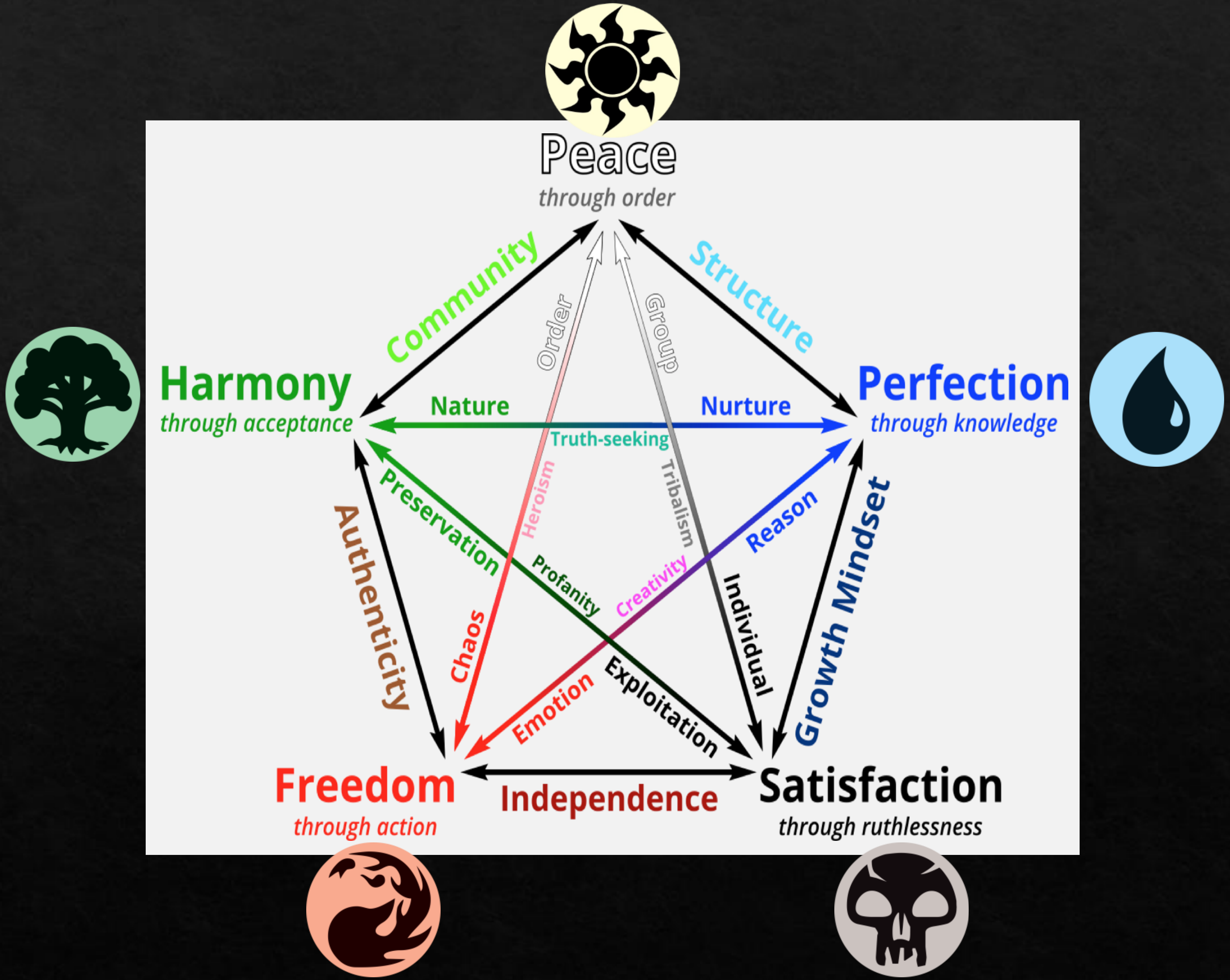
What is Magic the Gathering?

"Daze", art by Richard Wright


0:00 / 1:54
What is Magic: The Gathering?
296 125 visualizações • 13/06/2017
3,2 MIL
NÃO GOSTO
PARTILHAR
TRANSFERIR
GUARDAR
Magic: The Gathering
525 mil subscritores
SUBSCREVER

Color Identities

Colorless



Anatomy of a Card (1/2)

Mana Cost

Card name

Card Type

Abilities/
Card Text

Flavor Text (optional)

Artist, Expansion Code,
Rarity

Anatomy of a Card (2/2)

Creature Type

Keywords/ Abilities (optional)

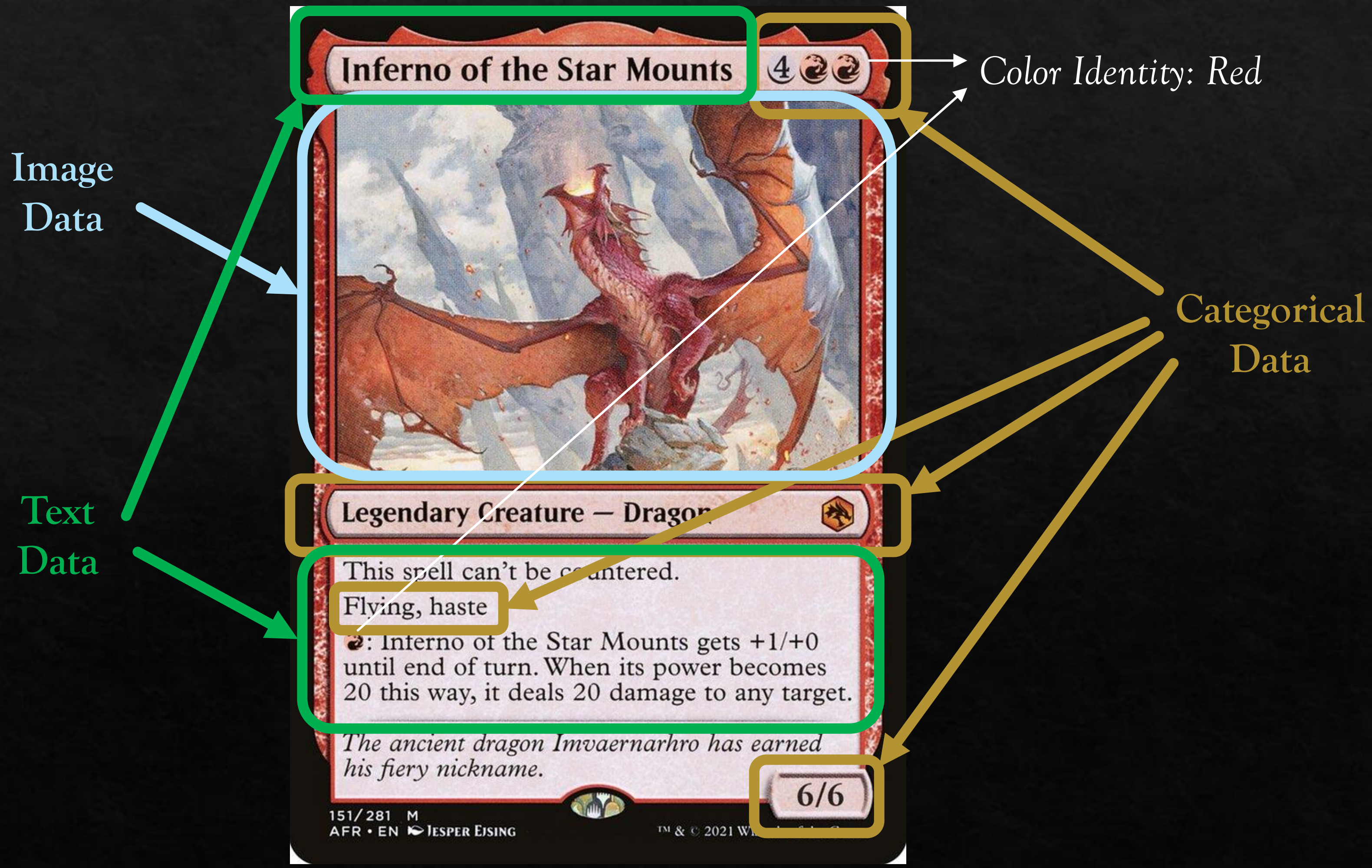
Power

Toughness

Planeswalker name

Loyalty

Classification problem: *"Given a certain card's characteristics, can we create a model that correctly predicts the card's **color identity**?"*



Data acquisition & preparation

“Solve the Equation”, art by Lie Setiawan

MTGJSON
v5.2.0+20220530

https://mtgjson.com/

AtomicCards

File containing every Card (Atomic) card.

Select a file to download

AtomicCards


```python
#we have more cards now because all the double sided cards got unfolded
json_normalize(data_dict["Rune-Tail, Kitsune Ascendant // Rune-Tail's Essence"])
```

	colorIdentity	colors	convertedManaCost	faceName	foreignData	layout	manaCost	manaValue	name
0	[W]	[W]	3.0	Rune-Tail, Kitsune Ascendant	[]	flip	{2}{W}	3.0	Rune-Tail, Kitsune Ascendant // Rune-Tail's Es...
1	[W]	[W]	3.0	Rune-Tail's Essence	[]	flip	NaN	3.0	Rune-Tail, Kitsune Ascendant // Rune-Tail's Es...

atomic_data

- Drop columns we won't need, drop all illegal cards, replace Null values, rename columns, confirm that 'Name' is a primary key

```python
df.head(15)
```

✓ 0.3s

Python

	Color	Subtype	Text	Type	Loyalty	Power	Toughness	Keyword	Name	ManaValue
0	['White']	['Plains']	({T}: Add {W}.)	['Land']	N/A	N/A	N/A	['N/A']	Plains	0.0
1	['Blue']	['Island']	({T}: Add {U}.)	['Land']	N/A	N/A	N/A	['N/A']	Island	0.0
2	['Black']	['Swamp']	({T}: Add {B}.)	['Land']	N/A	N/A	N/A	['N/A']	Swamp	0.0
3	['Red']	['Mountain']	({T}: Add {R}.)	['Land']	N/A	N/A	N/A	['N/A']	Mountain	0.0
4	['Green']	['Forest']	({T}: Add {G}.)	['Land']	N/A	N/A	N/A	['N/A']	Forest	0.0
5	['White']	['Ajani']	[+1]: Put a +1/+1 counter on up to one target ...	['Planeswalker']	4	N/A	N/A	['N/A']	Ajani, Caller of the Pride	3.0
6	['White']	['Cat', 'Soldier']	Whenever an enchantment enters the battlefield...	['Creature']	N/A	3	3	['None']	Ajani's Chosen	4.0
7	['White']	['N/A']	At the beginning of each end step, if you gain...	['Enchantment']	N/A	N/A	N/A	['N/A']	Angelic Accord	4.0
8	['White']	['Wall']	Defender (This creature can't attack.)\nFlying	['Creature']	N/A	0	4	['Defender', 'Flying']	Angelic Wall	2.0
9	['White']	['Angel']	Flying\nLifelink (Damage dealt by this creatur...	['Creature']	N/A	3	4	['Flying', 'Lifelink']	Archangel of Thune	5.0
10	['White']	['Human', 'Wizard']	When Auramancer enters the battlefield, you ma...	['Creature']	N/A	2	2	['None']	Auramancer	3.0
11	['White']	['Human', 'Cleric']	When Banisher Priest enters the battlefield, e...	['Creature']	N/A	2	2	['None']	Banisher Priest	3.0
12	['White']	['Aura']	Enchant creature\n{W}: Enchanted creature gets...	['Enchantment']	N/A	N/A	N/A	['Enchant']	Blessing	2.0
13	['White']	['Sliver']	Sliver creatures you control have double strik...	['Creature']	N/A	2	2	['None']	Bonescythe Sliver	4.0
14	['White']	['N/A']	Choose a color. White creatures you control ga...	['Instant']	N/A	N/A	N/A	['N/A']	Brave the Elements	1.0

atomic_data_clean

Label

Categorical

Text Data

Categorical*

Primary Key
(also text data)

Categorical*

	Color	Subtype	Text	Type	Loyalty	Power	Toughness	Keyword	Name	ManaValue
0	['White']	['Plains']	({T}: Add {W}.)	['Land']	N/A	N/A	N/A	['N/A']	Plains	0.0

Instead of creating dummy variables, we will treat the categorical variables as text data.

	Color	Subtype	Text	Type	Loyalty	Power	Toughness	Keyword	Name	ManaValue
0	['White']	['Plains']	({T}: Add {W}.)	['Land']	N/A	N/A	N/A	['N/A']	Plains	0.0
1	['Blue']	['Island']	({T}: Add {U}.)	['Land']	N/A	N/A	N/A	['N/A']	Island	0.0
2	['Black']	['Swamp']	({T}: Add {B}.)	['Land']	N/A	N/A	N/A	['N/A']	Swamp	0.0
3	['Red']	['Mountain']	({T}: Add {R}.)	['Land']	N/A	N/A	N/A	['N/A']	Mountain	0.0
4	['Green']	['Forest']	({T}: Add {G}.)	['Land']	N/A	N/A	N/A	['N/A']	Forest	0.0

```
df.head(5)
```

✓ 0.4s

	Color	Text	Name	Dummies
0	[White]	({T}: Add {W}.)	Plains	Plains Land N/A N/A N/A N/A 0.0
1	[Blue]	({T}: Add {U}.)	Island	Island Land N/A N/A N/A N/A 0.0
2	[Black]	({T}: Add {B}.)	Swamp	Swamp Land N/A N/A N/A N/A 0.0
3	[Red]	({T}: Add {R}.)	Mountain	Mountain Land N/A N/A N/A N/A 0.0
4	[Green]	({T}: Add {G}.)	Forest	Forest Land N/A N/A N/A N/A 0.0

"Dummies" column contains all categorical data

All "text"-type data.

Clean text data for machine learning

- Remove special characters
- Remove html tags
- Remove punctuation
- Remove non-alphabetic numbers
- Remove stop words (*e.g. 'may', 'also', 'across', 'among', 'N/A'*)
- Stemming: transform words with roughly the same semantics to one standard form (*e.g. 'enchantment', 'enchant', 'enchanted' all become 'enchant'*)

We also keep an untouched version of the PK

	Color	Name	name	text	dummies
0	White	Plains	plain	add w	plain land
1	Blue	Island	island	add u	island land
2	Black	Swamp	swamp	add b	swamp land
3	Red	Mountain	moun[illegible]	add r	mountain land
4	Green	Forest	fore[illegible]	add g	forest land
5	White	Ajani, Caller of the Pride	ajani caller [illegible]	[illegible]arget creatur gain...	ajani planeswalk
6	White	Ajani's Chosen	ajani cho[illegible]	[illegible]ld control creat...	cat soldier creatur none
7	White	Angelic Accord	angel acc[illegible]	[illegible]turn creat white ange...	enchant
8	White	Angelic Wall	angel wall	defend creatur attack fli	wall creatur defend fli

Where is the Image Data???

Getting the image data

MTGJSON
v5.2.0+20220530

https://mtgjson.com/

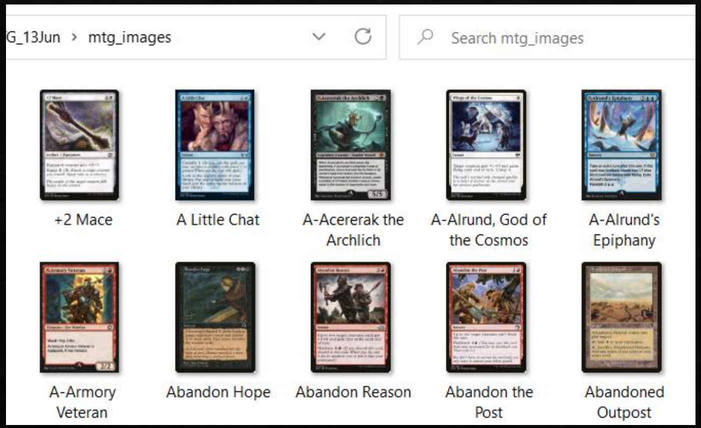
AllPrintings

all_printings

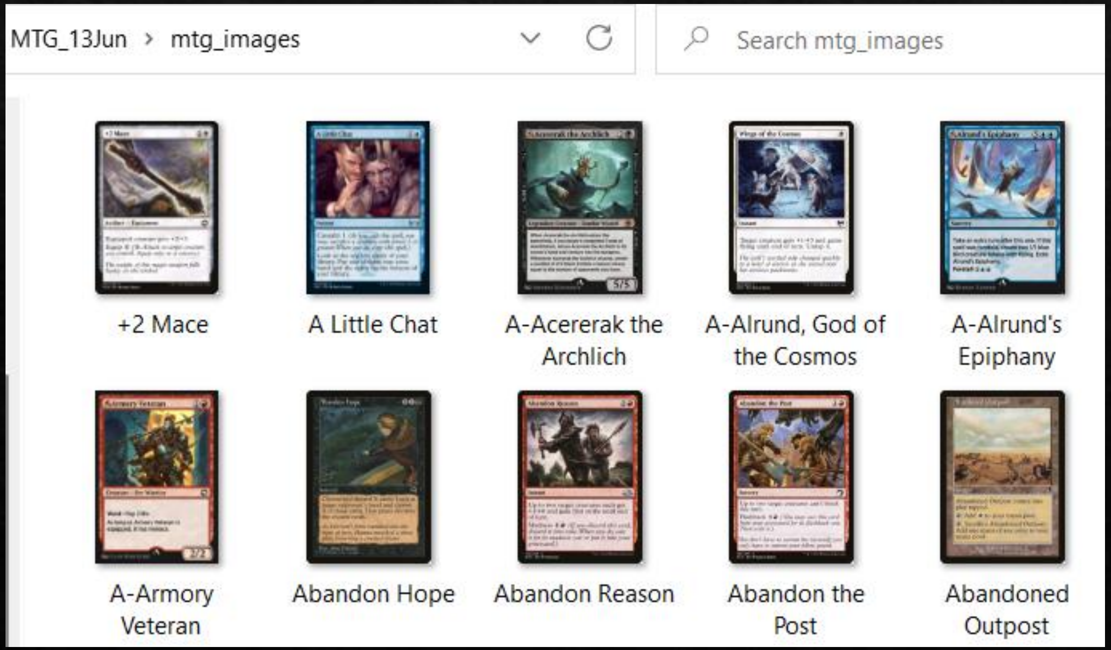
- Select only 1 card art for each card and avoid full art (so we can separate the art from the card image better).

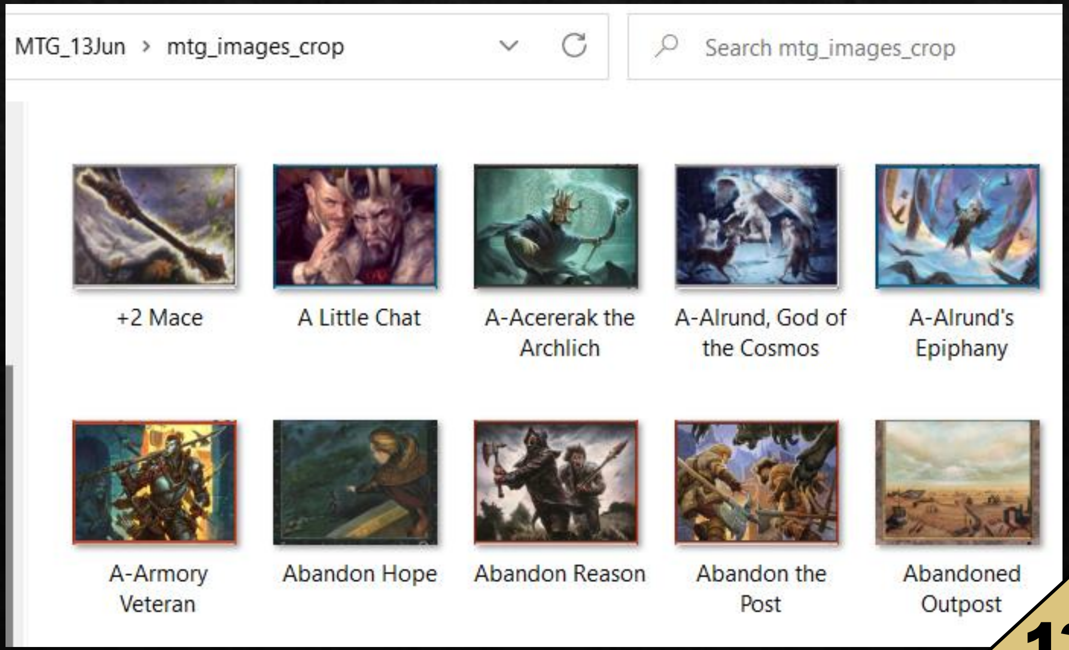
	scryfallId	Name
0	7ee52536-8cfa-482b-874e-094c0a081894	Plains
1	161aaceb-d0bb-48c5-8bd4-bff321a94b2e	Island
2	4df49e68-cadf-4196-a3f4-ae38579edaeb	Swamp
3	890037ae-c366-4769-b7f7-7185a1bebca1	Mountain
4	86dae285-a59c-426c-b6cd-3683abea75a3	Forest
...	...	...
67803	e65356e6-0ead-49fd-b069-be1ea9b1c105	Cathedral of Serra
67806	314fd1d7-4bd8-4d95-b7c2-1aa6660ab88a	Mountain Stronghold
67808	66641d88-b3f0-4bcd-8d2d-29aa2de69e30	Seafarer's Quay
67810	d43c01b7-443d-4061-a934-6863d230c9b8	Tolaria
67811	9de534ff-fb48-4692-bd0f-dd237ca28502	Unholy Citadel

MTG API
https://scryfall.com/



Crop the card art out of all the images





Save image data inside the data frame

	Color	Name	name	text	dummies
0	White	Plains	plain	add w	plain land
1	Blue	Island	island	add u	island land
2	Black	Swamp	swamp	add b	swamp land

	Color	Name	name	text	dummies	ImagePath
0	White	Plains	plain	add w	plain land	D:/Benchtime/MTG_13Jun/mtg_images_crop/Plains.jpg
1	Blue	Island	island	add u	island land	D:/Benchtime/MTG_13Jun/mtg_images_crop/Island.jpg
2	Black	Swamp	swamp	add b	swamp land	D:/Benchtime/MTG_13Jun/mtg_images_crop/Swamp.jpg

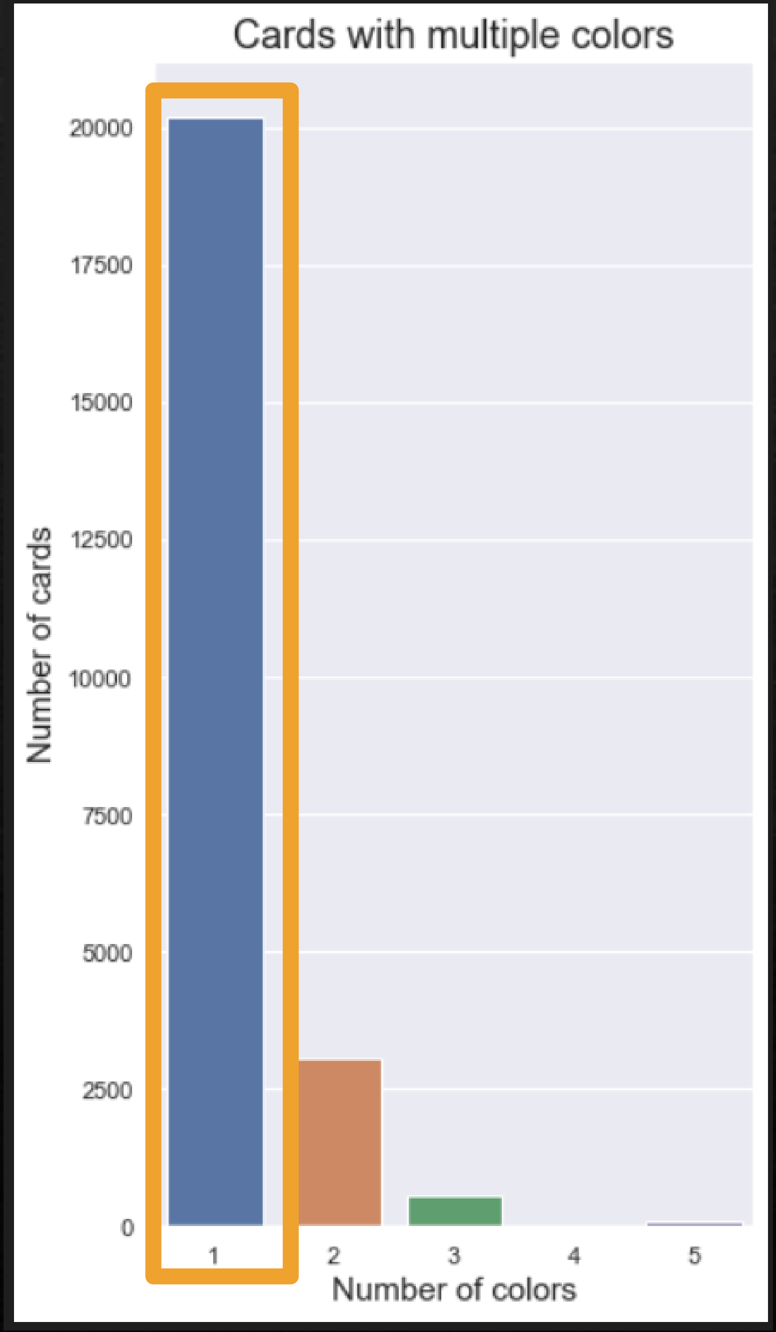
Ok now we're ready!

	Color	Name	name	text	ImageData
0	White	Plains	plain	add [illegible]	[illegible] [85, 94, 104], [101, 111, 12...
1	Blue	Island	island	add u	[illegible] 43], [illegible] 48, 31], [71, 69, 84], [2...
2	Black	Swamp	swamp	add b	[illegible] 17, 26, 32], [31, 35, 39], [77, 81, 89], [2...

Preparing data for Machine Learning

“Time Warp”, art by Dominik Mayer

Adequate sample sizes



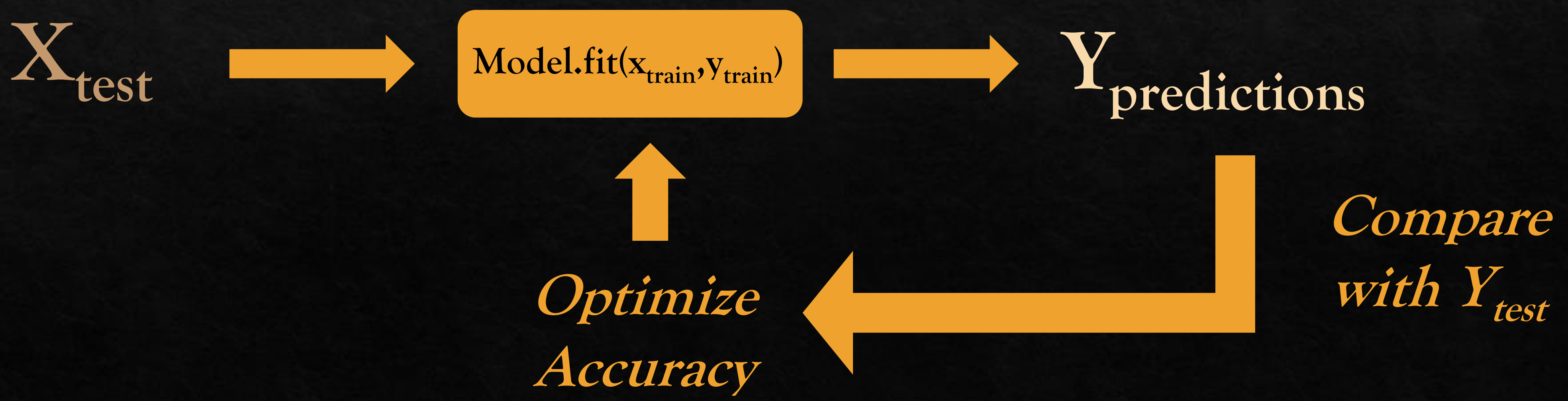

```
df['color_sum'].value_counts()

1    20202
2     3064
3      573
5       73
4       12
Name: color_sum, dtype: int64
```

Supervised Learning

Y (Correct Labels)   X (Text/Image Features)

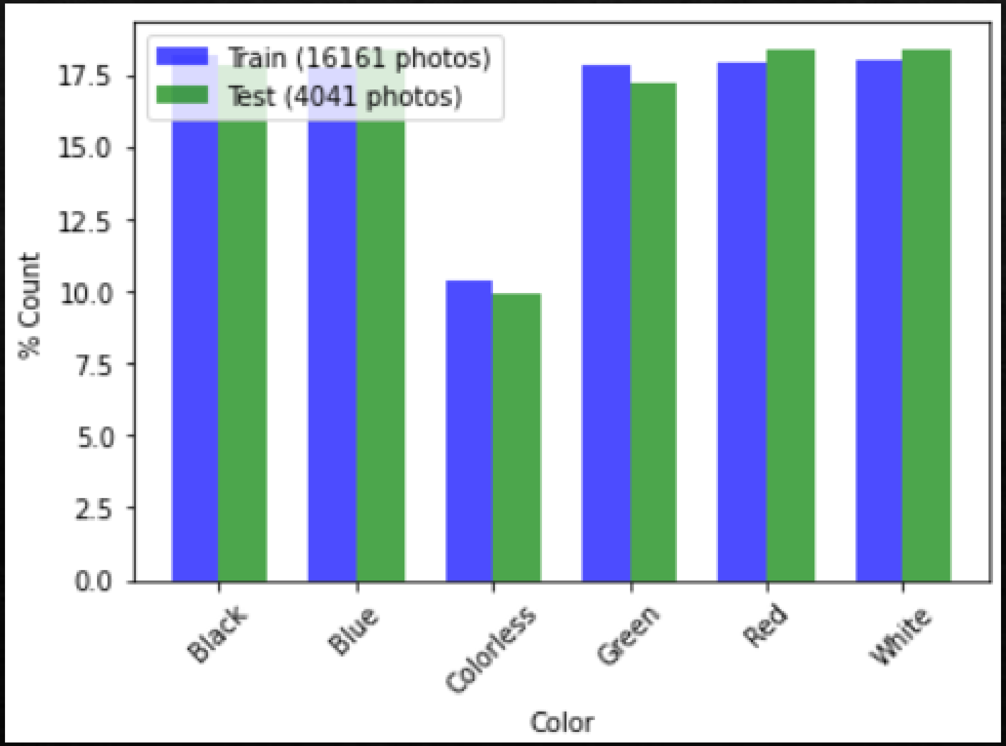
	Color	Name	name	text	dummies	ImageData
0	White	Plains	plain	add w	plain land	[[[92, 105, 112], [85, 94, 104], [101, 111, 12...
1	Blue	Island	island	add u	island land	[[[80, 54, 43], [74, 48, 31], [71, 69, 84], [2...
2	Black	Swamp	swamp	add b	swamp land	[[[17, 26, 32], [31, 35, 39], [77, 81, 89], [2...



Train/Test Split

$X, Y \rightarrow$ 80% X_{train}, Y_{train}

$X, Y \rightarrow$ 20% X_{test}, Y_{test}



2 Machine Models: 1 for text and 1 for image data

Y (Labels) | X (Text Model) | X (Image Model)

	Color	Name	name	text	dummies	ImageData
0	White	Plains	plain	add w	plain land	[[92, 105, 112], [85, 94, 104], [101, 111, 12...
1	Blue	Island	island	add u	island land	[[[80, 54, 43], [74, 48, 31], [71, 69, 84], [2...
2	Black	Swamp	swamp	add b	swamp land	[[[17, 26, 32], [31, 35, 39], [77, 81, 89], [2...

X Train
+
X Test

X Train Text
+
X Test Text

Model 1 – Text data (card name, rules and categorical variables)

X Train Image
+
X Test Image

Model 2 – Image data (card art)

Text feature extraction

X (Text Model)

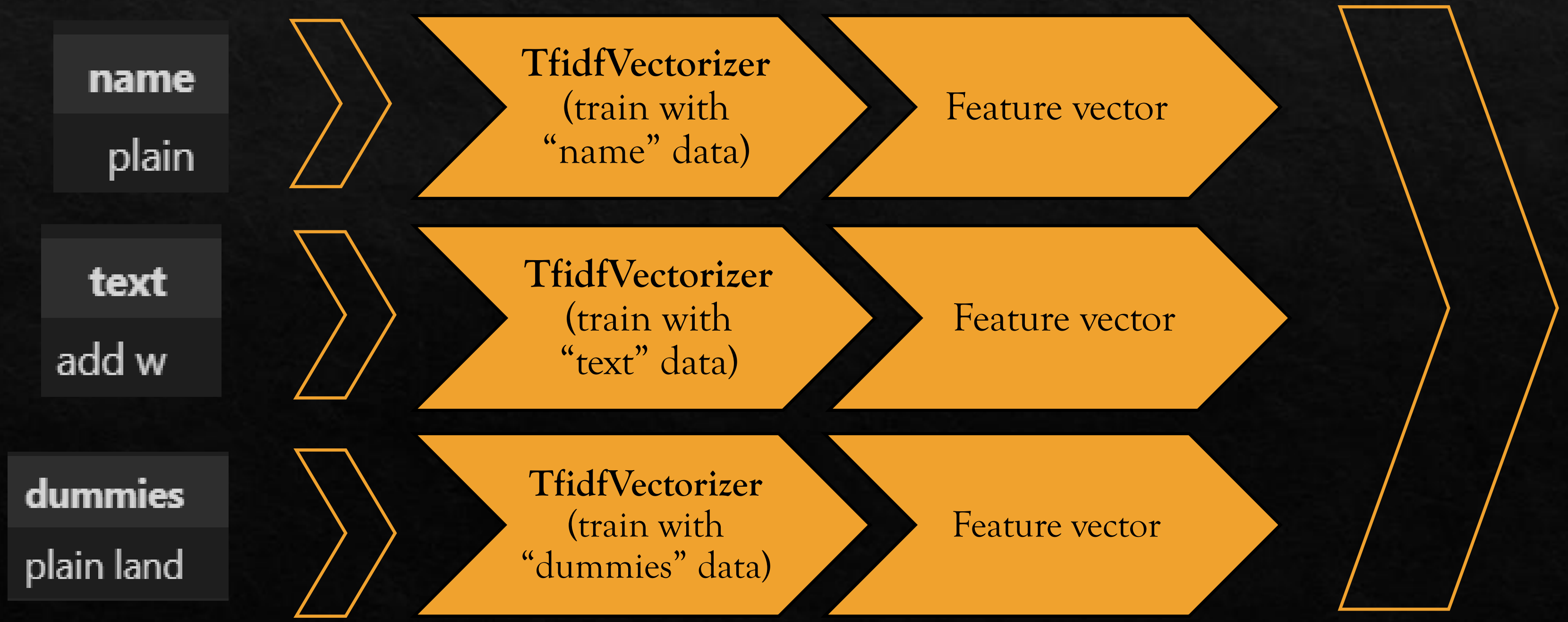
name	text	dummies
plain	add w	plain land
island	add u	island land
swamp	add b	swamp land

TF-IDF "Term Frequency – Inverse Document" method:

i) Calculates how often a given word appears within a card

ii) Downscales words that appear a lot across cards

iii) The output is a vector with frequency scores highlighting words that are more **interesting**



Stack the feature vectors inside a multi dimensional array (1 per card).

Image feature extraction

Color cv2.calcHist()

- **Color Histogram:** characterizes color distribution in an image

Texture mahotas.features.haralick()

- **Haralick Texture:** quantifies the spatial variation of grey tone values

Shape cv2.HuMoments()

- **Hu Moments:** set of 7 numbers invariant to image transformations, and calculated based on the intensity and position of the pixels in an image

Others skimage.feature.hog()

- **Histogram of oriented gradients (HOG):** counts occurrences of gradient orientation in localized portions of an image

ImageData
[[[92, 105, 112], [85, 94, 104], [101, 111, 12...

4 Image Descriptors → 4 Feature vectors → Stack the feature vectors inside a multi dimensional array (1 per image).

Model 1: Text data

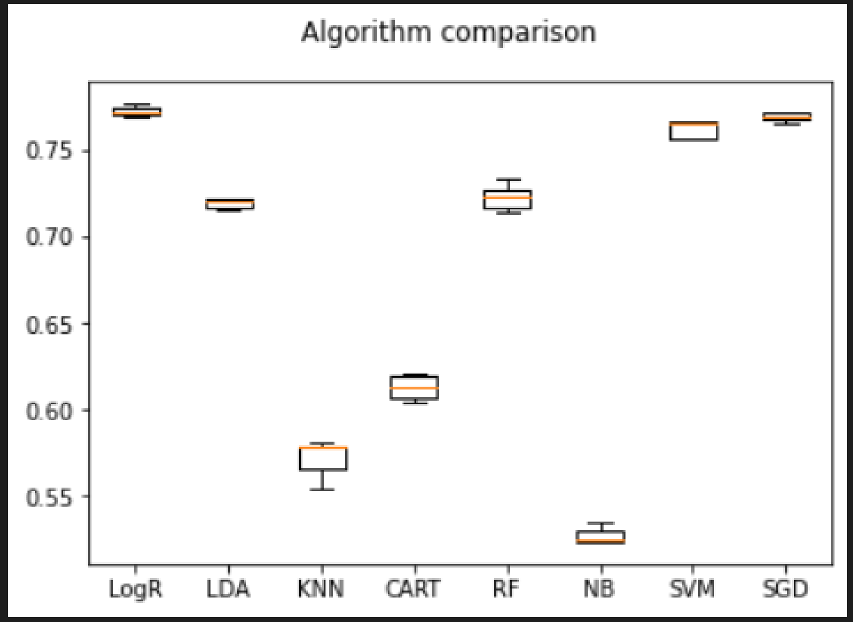
"Tolarian Scholar", art by Sara Winters

Cross-validation

To compare and select the best performing model, 10-fold (*sklearn.model_selection.KFold*) cross validation was used.

Classification models compared:

Logistic Regression >> *best performance/speed*
Linear Discriminant Analysis
K-Nearest Neighbors
Decision Tree Classifier
Random Forest Classifier
Naive Bayes
Support Vector Machines
Stochastic Gradient Descent Classifier

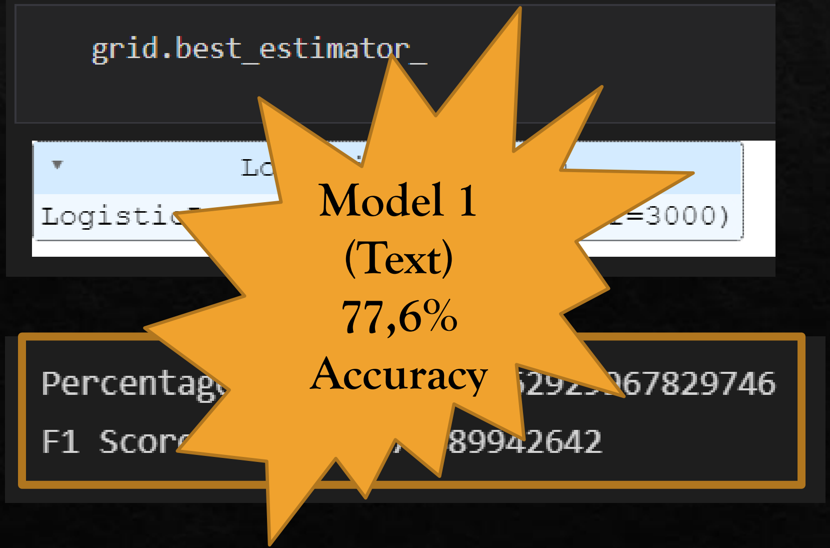


Grid-search

To tune the parameter C of the Logistic Regression model, Grid Search was used (*sklearn.model_selection.GridSearchCV*).

A high value of C tells the model to give more weight to the training data.
A lower value of C will let the model know that the training data may not be fully representative of real world data.

After testing values of C (0.1, 1, 10, 100, 1000), grid search arrives at:



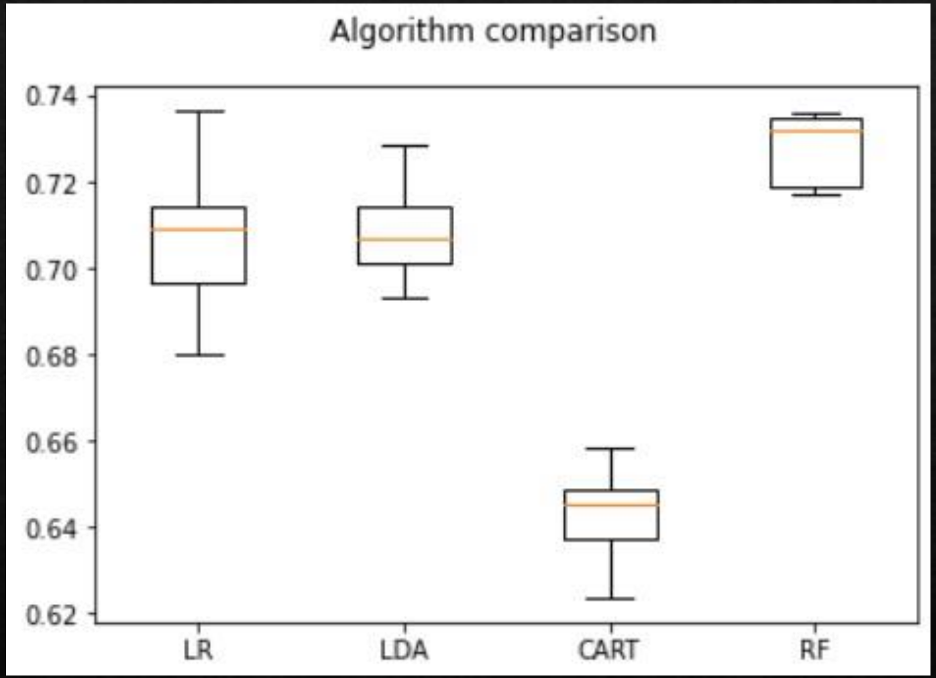
Model 2: Image data

“Sanguine Brushstroke”, art by Wayne Wu

Cross-validation

Classification methods compared:

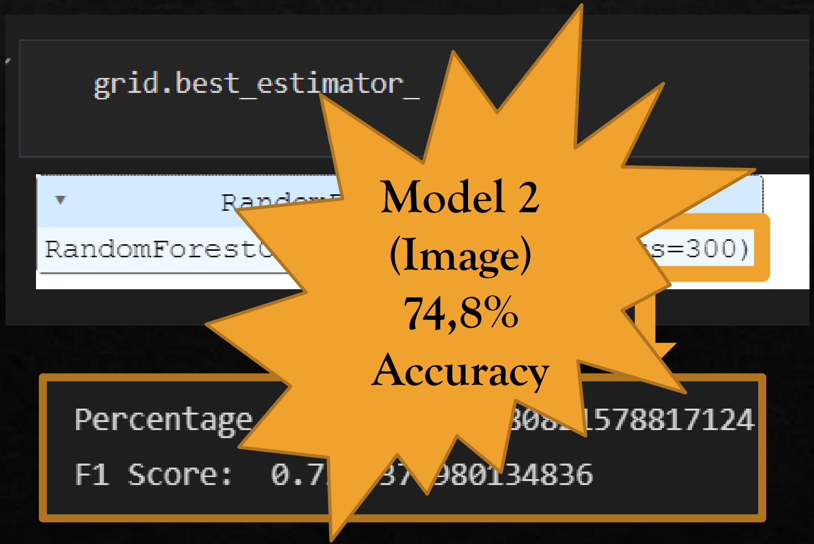
Logistic Regression
Linear Discriminant Analysis
Decision Tree Classifier
Random Forest Classifier >> *best performance/speed*



Grid-search

n_estimators is the number of trees the model should build before taking the maximum voting or averages of predictions:
A higher number will increase model performance, but will require more computational and time resources.

After testing values of *n_estimators* (100, 200, 300, 400, 500, 600), grid search arrives at:



Combined Model: Soft voting system

“Illusion of Choice”
art by John Severin Brassell

Soft voting system

Color Name name text dummies ImageData — Training set (80%)

Color Name name text dummies ImageData — Testing set (20%)

Model 1
Logistic Regression
Text data
77.629% accuracy

$$\left[p_1^{black}, p_1^{blue}, p_1^{colorless}, p_1^{green}, p_1^{red}, p_1^{white}\right]$$

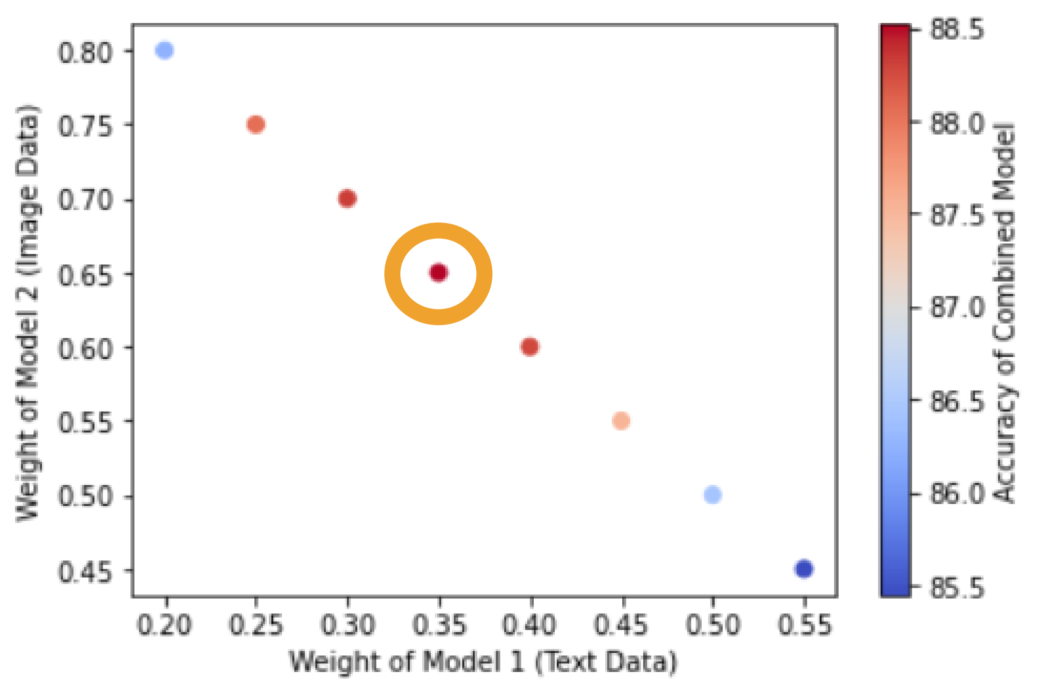
Model 2
Random Forest Classifier
Image data
74.808% accuracy

$$\left[p_2^{black}, p_2^{blue}, p_2^{colorless}, p_2^{green}, p_2^{red}, p_2^{white}\right]$$

Probability list of the Combined model

$$\left[p_1^{black}w_1 + p_2^{black}w_2, p_1^{blue}w_1 + p_2^{blue}w_2, p_1^{colorless}w_1 + p_2^{colorless}w_2, p_1^{green}w_1 + p_2^{green}w_2, p_1^{red}w_1 + p_2^{red}w_2, p_1^{white}w_1 + p_2^{white}w_2\right]$$

Choosing the best weight for each model (W1, W2)



W1	W2	Accuracy	F1 Score
0.55	0.45	85.449146	0.853941
0.50	0.50	86.439000	0.863872
0.45	0.55	87.527840	0.874745
0.40	0.60	88.270230	0.882204
0.35	0.65	88.517694	0.884682
0.30	0.70	88.319723	0.882706
0.25	0.75	88.047513	0.880014
0.20	0.80	86.241029	0.860970

What is a Confusion Matrix?

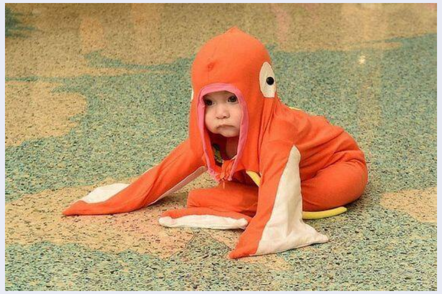
Visualizes and summarizes the performance of a classification algorithm.

Example: Is it an image of a Magikarp?

<u>True Negative</u>

Is NOT a Magikarp

<u>False Positive</u>



IS a Magikarp

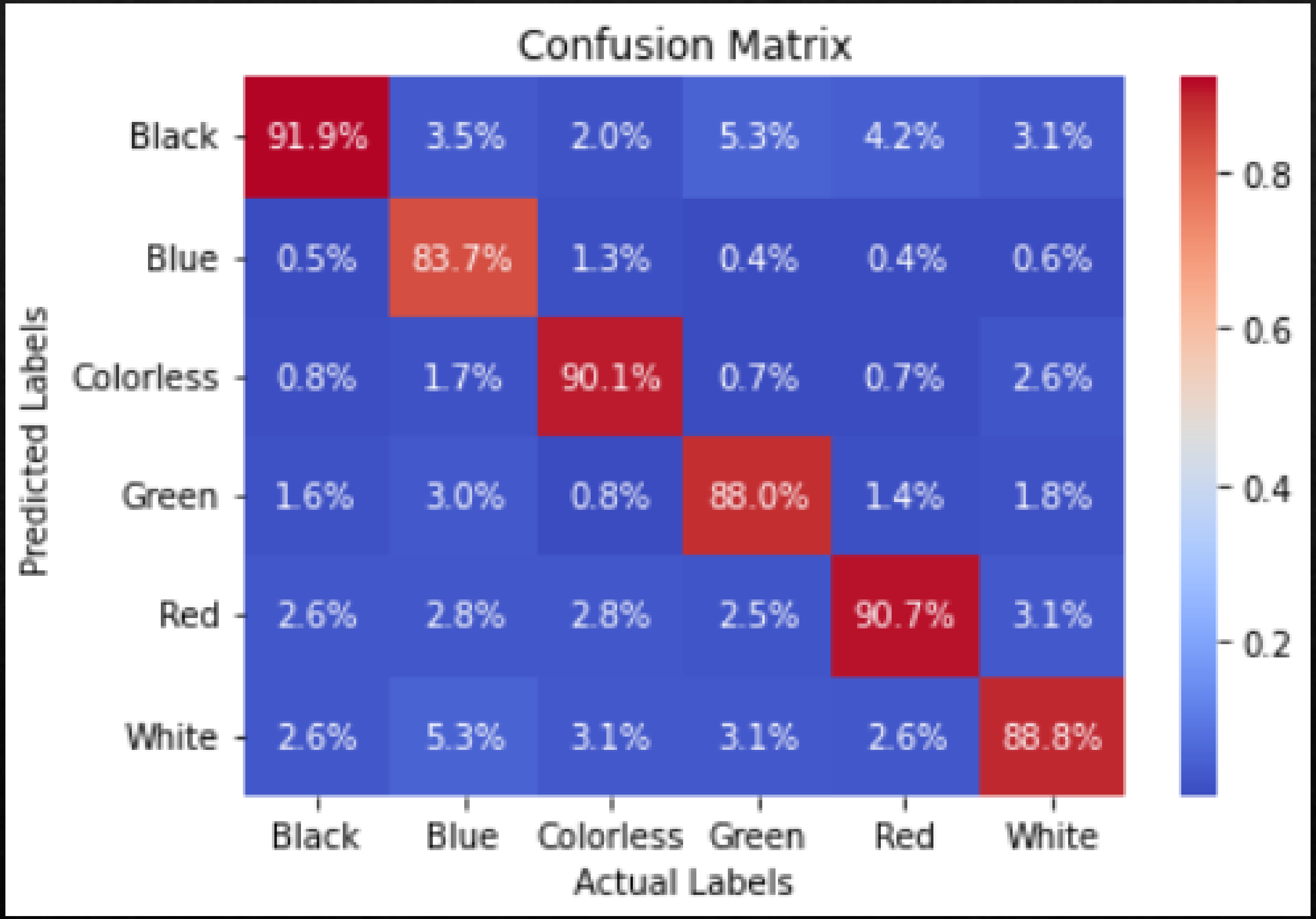
<u>False Negative</u>

Is NOT a Magikarp

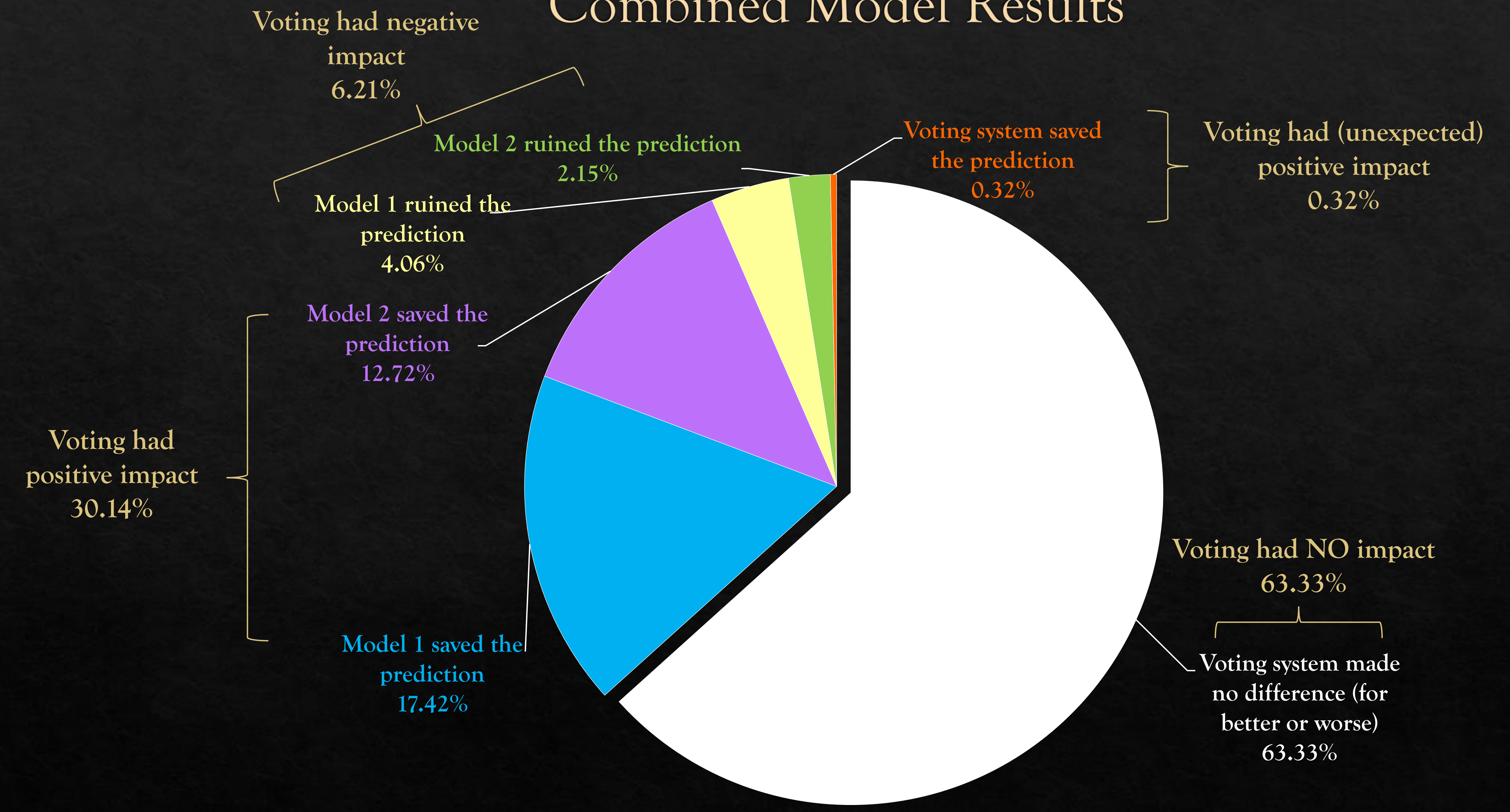
<u>True Positive</u>

IS a Magikarp

Confusion Matrix of Combined Model



Combined Model Results



Example: all models agree on correct prediction

Model 1 (Text)
0.006% Black
0.939% Blue
0.0% Colorless
0.002% Green
0.024% Red
0.029% White

Model 2 (Image)
0.051% Black
0.726% Blue
0.053% Colorless
0.077% Green
0.036% Red
0.057% White

Combined Model
0.035% Black
0.801% Blue
0.035% Colorless
0.05% Green
0.032% Red
0.047% White

Correct Answer

Example: all models guessed incorrectly

Model 1 (Text)
0.123% Black
0.001% Blue
0.861% Colorless
0.003% Green
0.0% Red
0.012% White

Model 2 (Image)
0.157% Black
0.133% Blue
0.063% Colorless
0.55% Green
0.043% Red
0.053% White

Combined Model
0.145% Black
0.087% Blue
0.343% Colorless
0.359% Green
0.028% Red
0.039% White

Correct Answer

Example: model 1 saves the prediction

Model 1 (Text)
0.006% Black
0.036% Blue
0.006% Colorless
0.939% Green
0.006% Red
0.006% White

Model 2 (Image)
0.123% Black
0.343% Blue
0.087% Colorless
0.337% Green
0.02% Red
0.09% White

Combined Model
0.082% Black
0.236% Blue
0.058% Colorless
0.548% Green
0.015% Red
0.061% White

Correct Answer

Example: model 2 saves the prediction

Model 1 (Text)
0.058% Black
0.045% Blue
0.001% Colorless
0.532% Green
0.311% Red
0.053% White

Model 2 (Image)
0.18% Black
0.13% Blue
0.063% Colorless
0.153% Green
0.37% Red
0.103% White

Combined Model
0.137% Black
0.1% Blue
0.041% Colorless
0.286% Green
0.349% Red
0.086% White

Correct Answer

Example: model 1 jeopardizes the prediction

Model 1 (Text)
0.005% Black
0.963% Blue
0.001% Colorless
0.0% Green
0.001% Red
0.03% White

Model 2 (Image)
0.077% Black
0.083% Blue
0.19% Colorless
0.072% Green
0.165% Red
0.413% White

Combined Model
0.052% Black
0.391% Blue
0.124% Colorless
0.047% Green
0.107% Red
0.279% White

Correct Answer

Example: model 2 jeopardizes the prediction

Model 1 (Text)
0.116% Black
0.142% Blue
0.536% Colorless
0.064% Green
0.07% Red
0.072% White

Model 2 (Image)
0.11% Black
0.57% Blue
0.073% Colorless
0.117% Green
0.043% Red
0.087% White

Combined Model
0.112% Black
0.42% Blue
0.235% Colorless
0.098% Green
0.053% Red
0.081% White

Correct Answer

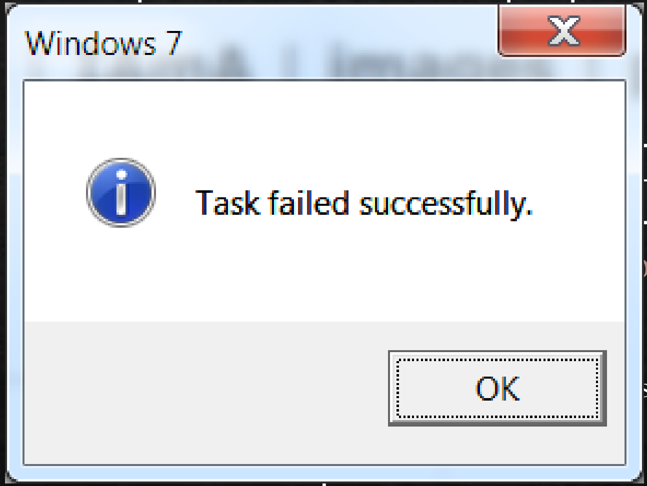
Example: model 1 and 2 are wrong, but combined model makes correct prediction

Model 1 (Text)

0.525% Black
0.376% Blue
0.0% Colorless

Model 2 (Image)

0.153% Black
0.25% Blue
0.107% Colorless
0.087% Green
0.26% Red
0.143% White



odel

s

0.111% White

Correct Answer

Q&A

Real footage of our cat helping us organize the card collection (totally not a reversed video!)

References

- https://mtgjson.com/
- https://scryfall.com/docs/api
- https://scikit-learn.org/
- https://www.datacamp.com/tutorial/wordcloud-python
- https://www.geeksforgeeks.org/python-pil-image-crop-method/
- https://www.linkedin.com/pulse/image-classification-opencv-scikit-learn-raj-ramanujam/
- https://learn.g2.com/logistic-regression
- https://gogul.dev/software/image-classification-python
- https://kapernikov.com/tutorial-image-classification-with-scikit-learn/
- https://towardsdatascience.com/visual-guide-to-the-confusion-matrix-bb63730c8eba
- https://towardsdatascience.com/a-big-of-tricks-for-image-classification-fec41eb28e01
- http://rasbt.github.io/mlxtend/user_guide/classifier/EnsembleVoteClassifier/
- https://towardsdatascience.com/combine-your-machine-learning-models-with-voting-fa1b42790d84
- https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/
- https://www.geeksforgeeks.org/an-introduction-to-multilabel-classification/
- https://www.kaggle.com/code/sociopath00/random-forest-using-gridsearchcv/notebook