# A Real-Bogus Classifier in Space

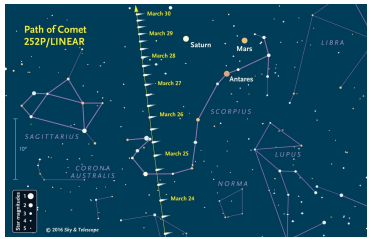
## Identifying Space Objects with Machine Learning

CS 109/209B
Spring 2018
Team #2

Michael Emanuel, Erin Williams and Anna Davydova

# Research Question:

**Can we create a classifier that will accurately distinguish between real detection of stars, asteroids, and galaxies, and bogus readings from Pan-STARRS data while keeping the False Negative rate low?**




## True Positive

Source:https://phys.org/news/2016-03-comet-252plinear-soars-predawn-view.html

## False Positive

Source:https://www.foxnews.com/science/supersonic-tic-tac-ufo-stalked-us-aircraft-carrier-for-days-pentagon-report-reveals

## False Negative

source:https://nypost.com/2016/07/31/nasa-plans-to-launch-study-of-asteroid-that-could-destroy-earth

## True Negative

Source: HBO

# Background

- Astronomical data is vast, but homogeneous and recorded in a consistent manner
- Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) telescope
  - Hawaii-based 1.8m telescope and 1.4 GP camera
  - Map approximately 41,000 square degrees 4x per month
  - Provides publically shared data on 75% of the visible nighttime sky
- Near-Earth Objects (NEOs)
  - Stars, asteroids, comets, galaxies, other celestial objects
- Over 1,000 NEO’s discovered via Pan-STARRS

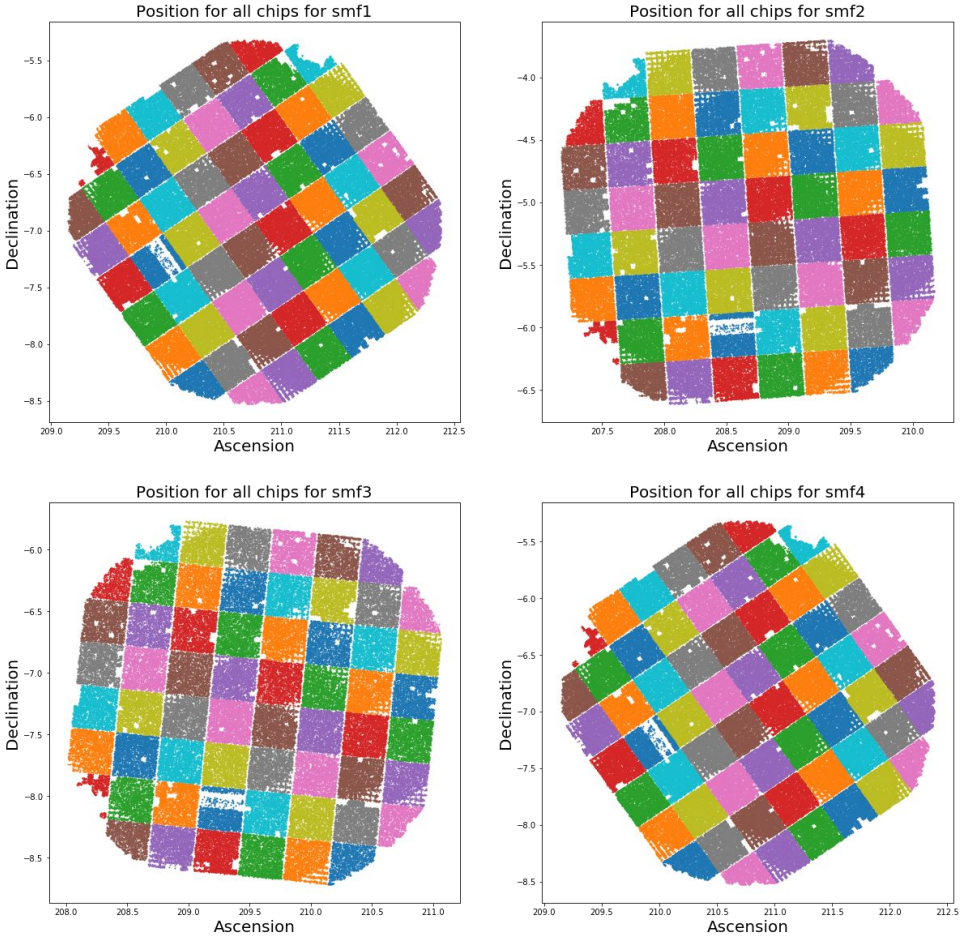
source:https://mauinow.com/2017/11/17/interstellar-object-discovered-with-maui-telescope-gets-hawaiian-name-%CA%BBoumuamua/

# Motivation

- Three years of Pan-STARRS data is 2 petabytes -- MASSIVE!
- Each Pan-STARRS scan results in a file with order 100,000 detections
- Detections can align with:
  - Known stationary objects (e.g. stars and galaxies)
  - Known moving NEOs (e.g. planets, comets, asteroids)
  - Unknown objects - either fixed or moving
  - Bogus (false) detections
- A Real-Bogus detector allows astronomers to determine which detections to examine in more detail versus which ones to ignore
- False positive vs. false negative trade-off
  - Higher cost for a false negative (missed data)
  - Goal to correctly classify 70-80% of bogus detections with a low false negative rate

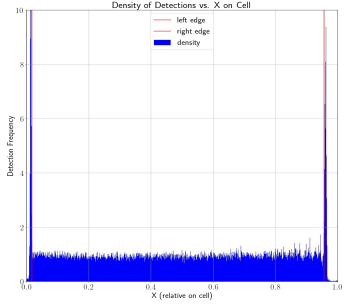
# EDA: Data Snapshot

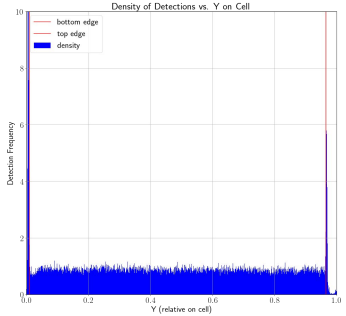
- 32 smf files with extensive information about each exposure.  20 labeled smf files.
- 60 chips for each exposure with various features.
- A visualization of 4 randomly selected  smf files (on the right) provides a snapshot of the ascension and declination parameters.
- Linear features appear to be aligned with the angle of the camera and gaps between each chip are based on their positioning,
- Gaps inside each chip occur when the light has nothing to hit.

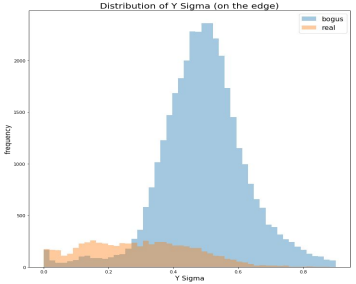


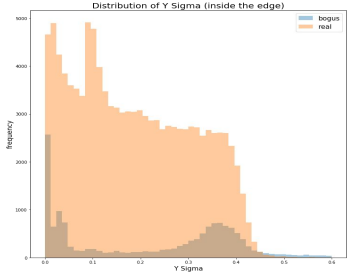
# EDA: Key Features for Classification

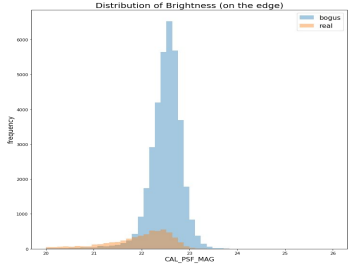
- Proximity to Edges:
  - We found a notably high concentration of detections around the chip edges.
  - Since we would expect a uniform distribution, there is a good reason to suspect that many of the edge observations are bogus.
- Sigma of Coordinates:
  - Standard deviations of X and Y coordinates tend to be higher for bogus observations (same results for X and Y coordinates), due to higher uncertainty about PSF center.
- Brightness:
  - Bogus observations tend to be dimmer, as the distributions on the right demonstrate.

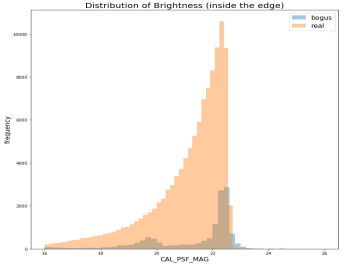


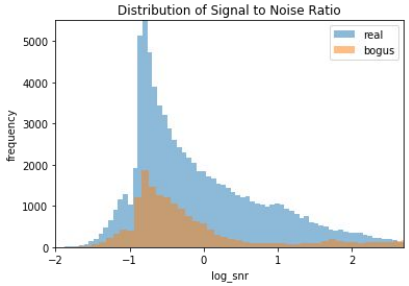



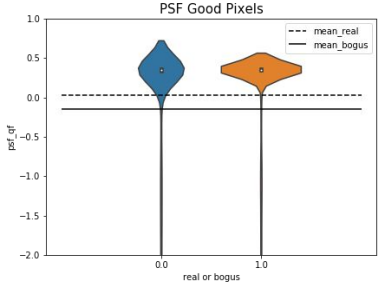



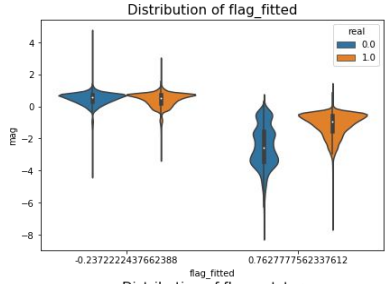



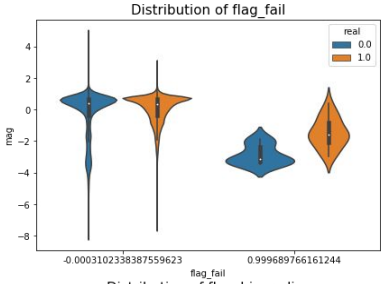
# EDA: Key Features for Classification

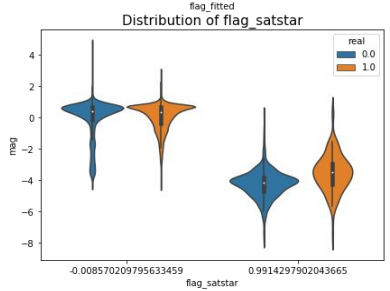
- Signal to Noise Ratio:
  - We also notice that bogus observations appear to be associated with somewhat lower signal to noise ratios.
- ‘Good’ Pixels:
  - We notice that real observations tend to have more good pixels.
- Flags:
  - Good/bag flag categories have different distributions for our key features for real and bogus observations. We provide four examples of such flags (of total 31)

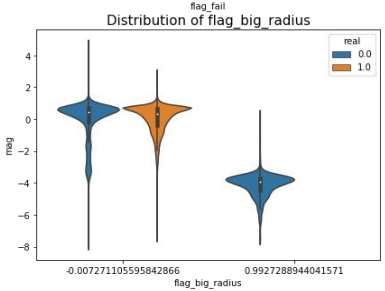


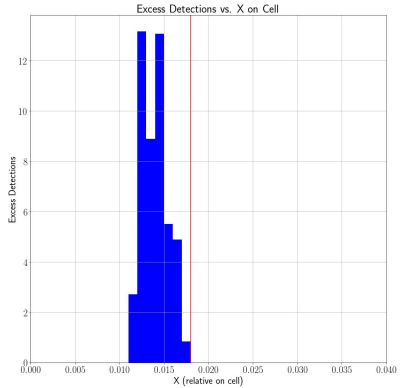



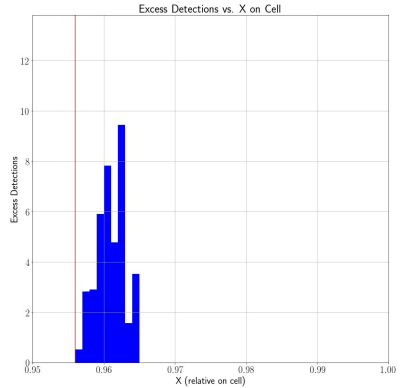
# Keeping it Old School: Logistic Regression

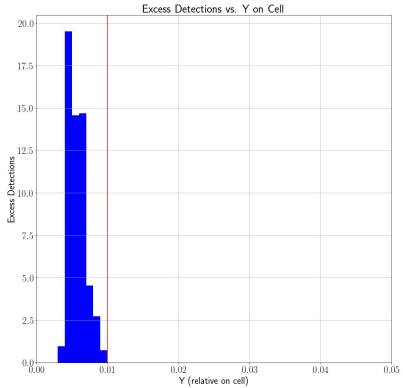
- We chose one of the oldest, simplest classifiers around: logistic regression
- This problem matches up well with the assumptions behind the LR, with a number of features making approximately additive contributions to log likelihood
- LR is computationally lightweight: runs fast with light memory footprint.
  - This makes it highly scalable, which is essential for a classifier to be useful here
- LR is also conceptually simple and easy to worth with; we can test our intuition about features quickly
  - This was a good choice because we had a tight time budget to develop this classifier
- The time saved by using a simple LR model was invested in feature engineering and feature selection
  - We spent considerable time tuning the shape of the four nonlinear edge features
  - We experimented with different groups of features, then combined them at the end

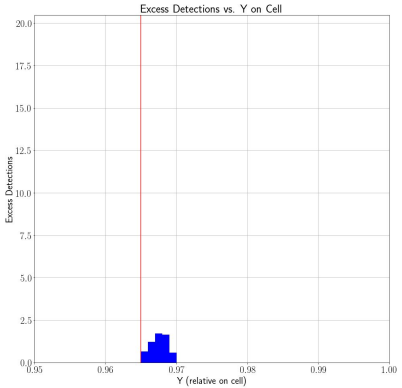
# Feature Engineering: Locating the Four Edges

- Using a simple LR model gave us more time to devote to feature engineering
- We thought we could improve on the simple hard edge indicator provided
- We defined excess detections along X and Y by normalizing the detections, subtracting 1, and flooring the result at 0
- We assembled a cumulative distribution and placed each edge (left, right, bottom, top) so as to capture 99% of the excess detections

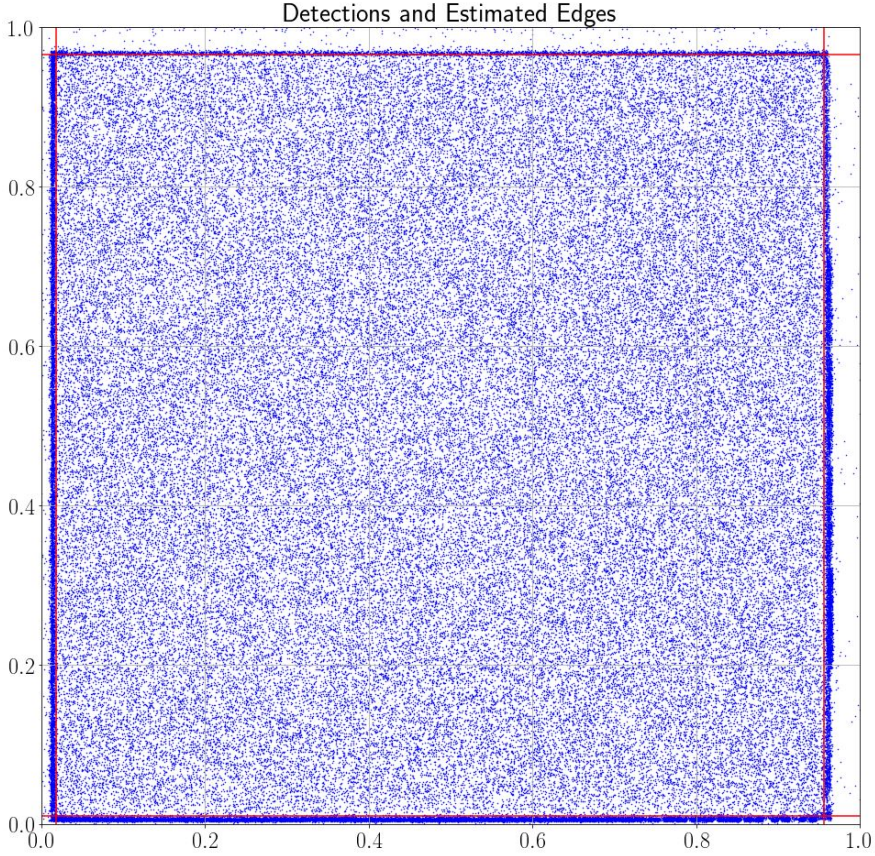


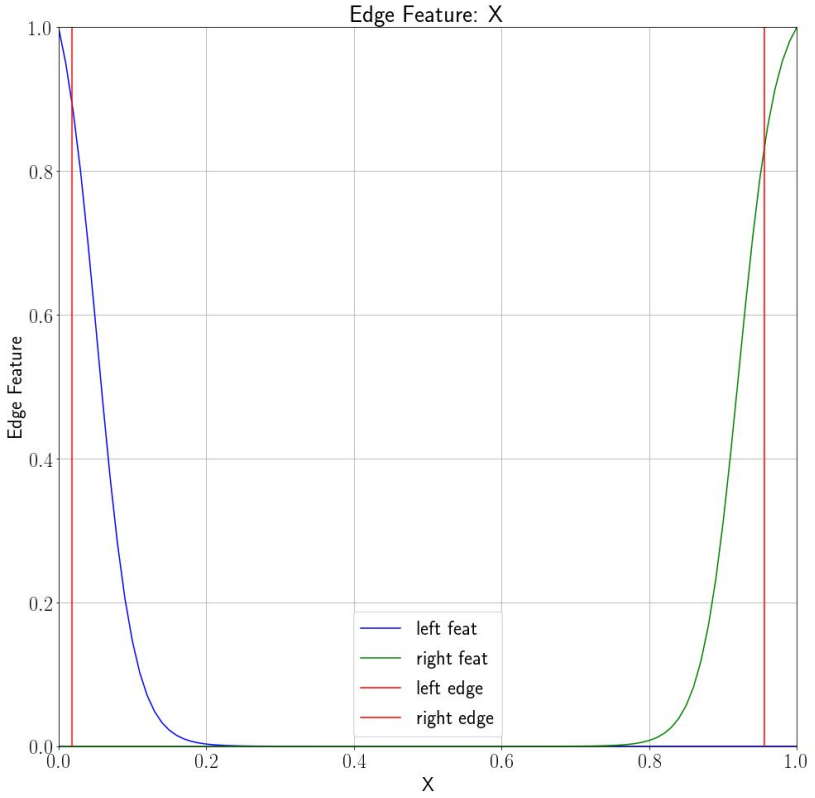



# Feature Engineering: Nonlinear Edge Features



# Scoring Our Results: Modified Confusion Matrix

- This problem has a unique structure; we predict one of 2 labels (bogus vs. real) but there are 3 ground truth categories(not in catalog, fixed, moving)
- We can describe a model's performance with a 3x2 "confusion matrix"

| **Catalog \ Predict** | **Bogus** | **Real** |
|---|---|---|
| **Unknown (30.02)** | True Negative | False Positive |
| **Fixed (69.76)** | Missed Fixed | True Fixed |
| **Moving (0.02)** | Missed Moving | True Moving |

Cost function for errors:

- False Positive: 1.0
- Missed Fixed: 2.0
- Missed Moving: 100.0

- Effective weights after boosting: unknown 15.7, fixed 72.9, moving 11.5
- Weights for moving objects reflect both rarity and importance
- Easy in sklearn; just use sample_weight parameter as a vector
- Model comparison done via weighted error rate

# Baseline Model: LR with Four Edge Features

| Feature | Coefficient |
|---|---|
| intercept | +3.151 |
| edge_left | -2.620 |
| edge_right | -3.192 |
| edge_bot | -3.570 |
| edge_top | -2.032 |

| | Bogus | Real |
|---|---|---|
| **Unknown** | 0.1270 | 0.1733 |
| **Fixed** | 0.0284 | 0.6691 |
| **Moving** | 0.001 | 0.0023 |

- Error raw 20.17%
- Error weighted 12.27%
- Recall fixed 95.92%
- Recall moving 97.24%
- Weeded Out 42.30%

- We were surprised at how good this model was; it's about 80% accurate and achieves 97.24% recall while weeding out 42.3% of bogus detections
- The coefficients are intuitive; all four edges highly negative
  - The effect is weaker for the top edge vs. the other three

# Feature Selection: One Group at a Time

- Our approach to feature selection was to experiment with features in one logical group at a time, seeing which combinations in each group worked well
- We always kept the edge features because we knew we would use them
- We compare models using weighted error rate; baseline = **12.27%**
- The X, Y sigma features are correlated to being on the edge but still very powerful; they reduce error to 9.18%
- Magnitude is the most powerful feature; reduces error to 8.69%.  Bright observations are better!
- QF reduces error to 10.24%.  Good pixels yield better detections!
- Log SNR reduces error to 11.0%.  High SNR is better.
- Hard edge helps in isolation, but not in conjunction with the rest; skip it
- Flags reduce error to 10.73%.  Good feature; hard to interpret individual flags because of colinearity
- Can filter out 18 irrelevant flags and keep just the 13 that matter
- Chip dummy indicators are marginal but harmless
- Chi Squared was a disappointment; thought it would help but it barely moved the needle
- Other features we tried didn’t help

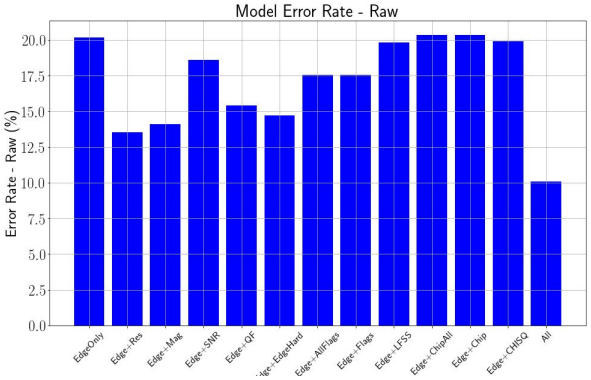
# Our Best Classifier: The Big Kahuna

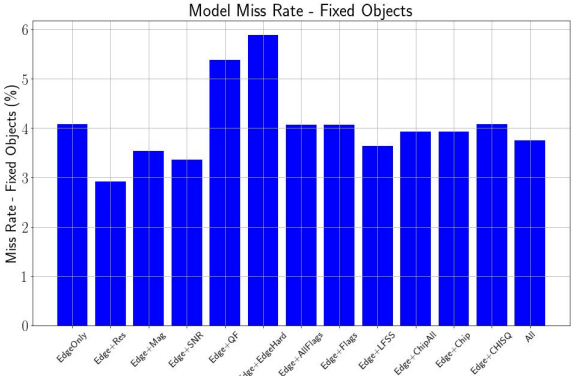
| | Bogus | Real |
|---|---|---|
| **Unknown** | 0.2255 | 0.0748 |
| **Fixed** | 0.0261 | 0.6713 |
| **Moving** | 0.0000 | 0.0023 |

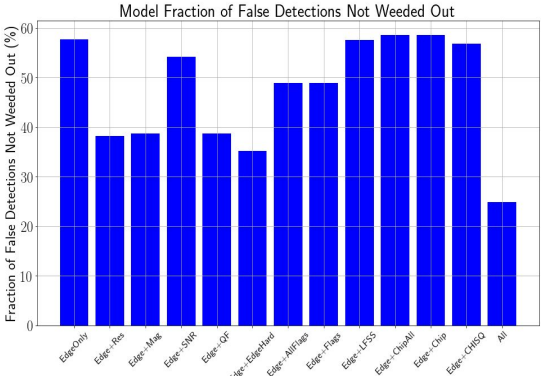
- Error raw 10.10%
- Error weighted 6.67%
- Recall fixed 96.25%
- Recall moving 99.35%
- Weeded Out 75.09%

- We used all the features that appeared to be useful in feature selection
- We did a round of testing to see if any should be dropped
- The coefficients are mostly intuitive; some flip due to colinearity
- We were very pleased with the results.
  - Better than 90% raw accuracy
  - Better than 99% recall on the most important moving objects
  - Weeding out more than 75% of the false detections

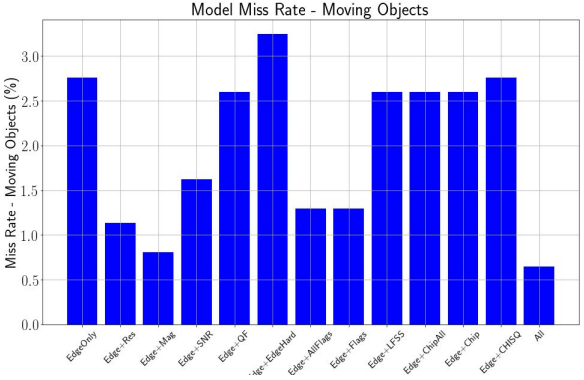
# Visualizing Performance of Different Models



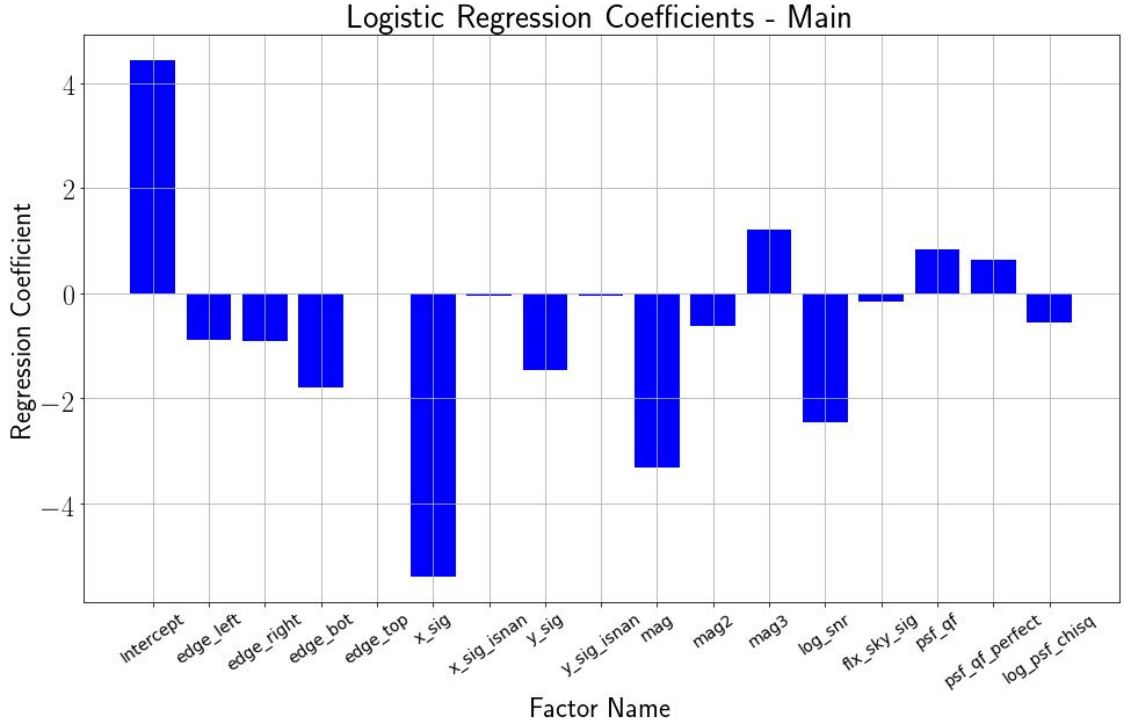



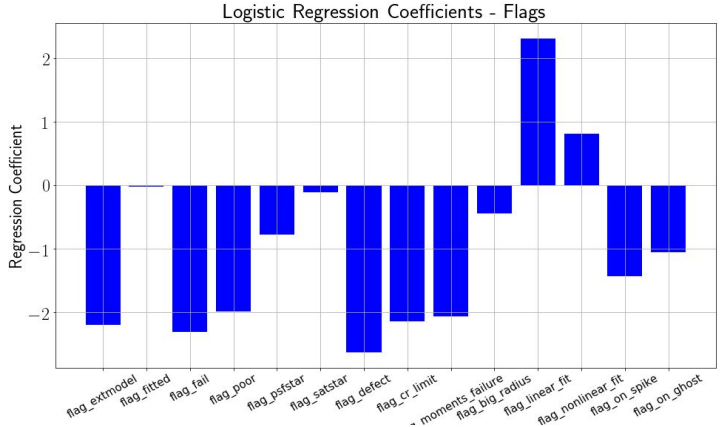





- Big Kahuna stands out
- Error rate 54% baseline
- Fixed - small improvement
- Moving - huge improvement
- Fraction weeded out - big improvement

# Visualizing Coefficients of The Big Kahuna




- Most coefficients intuitive
- x_sig, y_sig largely displace edge features
- Edge_top, xy_sig_isnan drop out
- Mag still big; mag2 and mag3 smaller now
- log_snr has flipped signs - weird
- QF factors still positive
- Chi Squared factor now negative - weird
- Flag features mostly have big, negative coefficients indicating failure conditions
- 2 good flags: linear fit (very); nonlinear (less)

# Conclusions and Future Work

- Using a simple logistic regression classifier and old fashioned feature engineering, we achieved excellent results.
  - 99.4% recall rate on moving objects while discarding 75% of bogus detections
  - Low computational resources required; could scale to full data set if desired
  - This was a success vs. the stated goal: filter out 70-80% with low false negatives (only 0.6%)
- While sophisticated ML techniques are great, sometimes they’re overkill
- Biggest room for improvement: use more SMFs (order 1000) to estimate edges for each cell individually vs. pooling all 3840 cells
- Speculative try: fit a 2 layer feed-forward neural network using same features
- Once best approach identified, tune the desired recall vs. precision, and start sorting detections in order of how promising they look.
- THANK YOU ALL FOR A GREAT SEMESTER!