Using MLB Statcast Metrics to Predict Home Runs and Extra-Base Hits

Data Source

- BaseballSavant on MLB.com
  - A site dedicated to providing player matchups, **Statcast** metrics, and advanced statistics in a simple and easy-to-view way

- Statcast:
  - A state-of-the-art tracking technology (radar and HD camera system), capable of measuring previously unquantifiable aspects of the game
  - New in 2015 to all 30 MLB ballparks
  - Statcast database contains files including measurements and metrics from each game throughout the baseball season

Data Set

- pybaseball
  - Python package for baseball data analysis; this package scrapes BaseballSavant
- Data pulled for:
  - 2015-2017 baseball seasons
  - All baseball teams
- N = 2,139,920 rows
  - Each row represents an interaction between a pitcher and a batter (i.e., a pitch thrown by the pitcher to the batter)
- 91 columns (16 useful columns used for examining data set)

Problem Statement

- Leveraging the Statcast metrics from MLB's BaseballSavant website, determine the predictive accuracy of baseball's offensive power statistics.

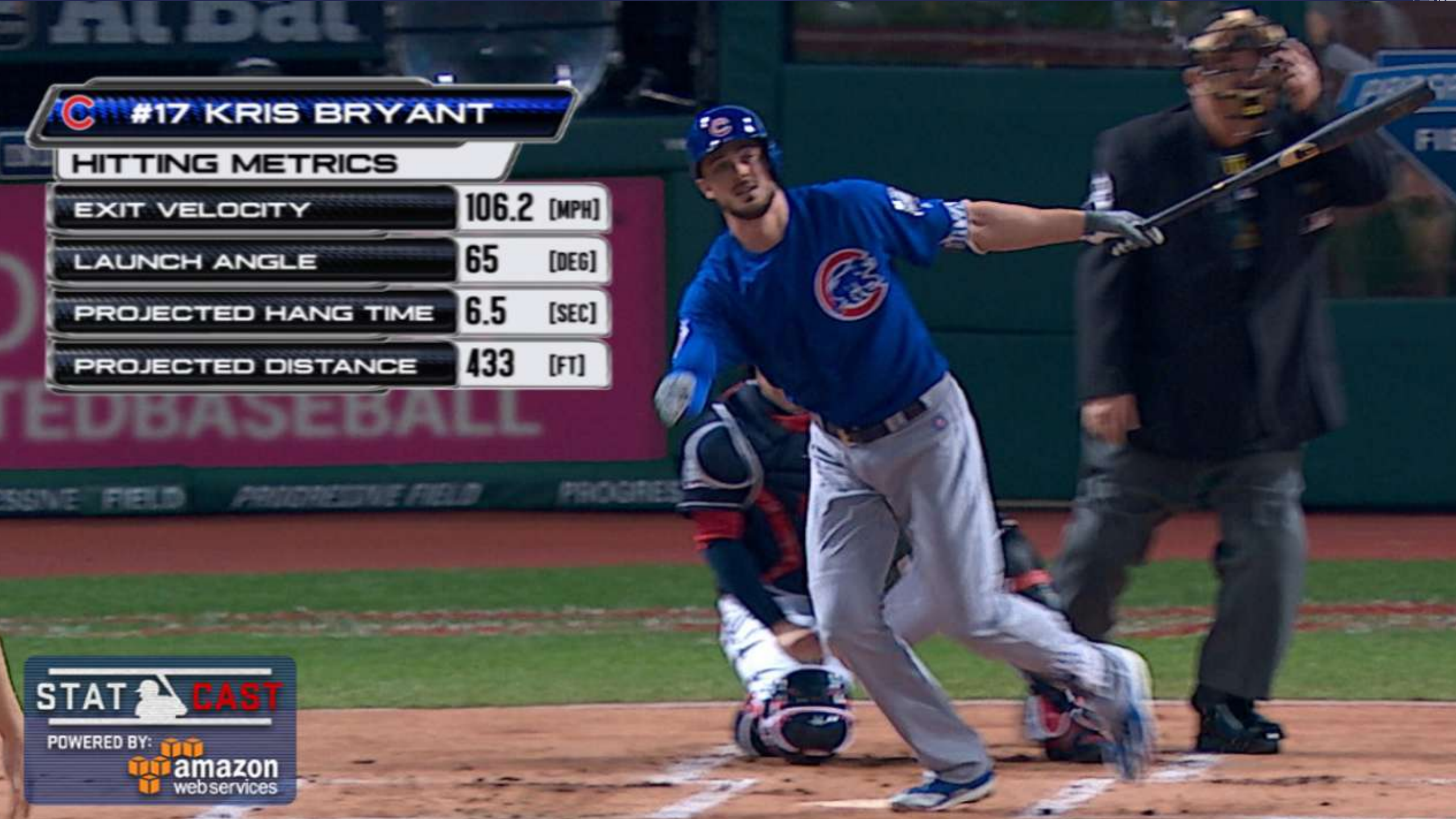
Outcomes for Supervised Learning (Predictive Modeling)

- Home Run (HR)
  - Among the most important events in games; great statistic for evaluating a hitter's power
- Extra-base Hit (XBH)
  - Any hit that is not a single, meaning doubles, triples and home runs; good statistic to evaluate an offensive player's power -- and in some cases, his speed

Note: Outcomes will be predicted using Logistic Regression/ Classification because they are categorical measures.

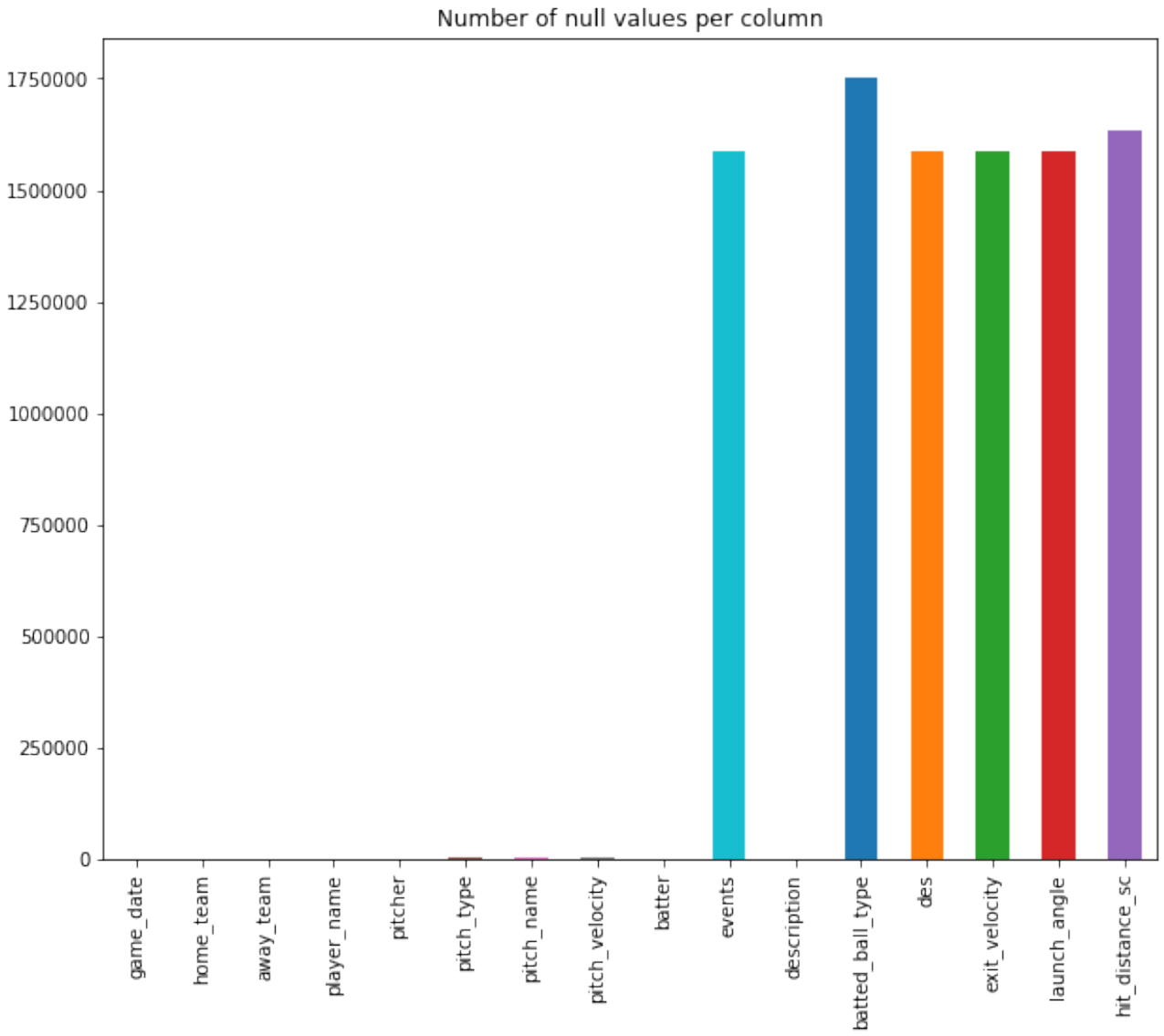
Predictors

- Exit Velocity
  - How fast, in miles per hour, a ball was hit by a batter
- Launch Angle
  - How high, in degrees, a ball was hit by a batter
- Pitch Velocity
  - How hard, in miles per hour, a pitch is thrown
- Pitch Type
  - Type of pitch thrown to defeat a batter (e.g., fastball, curveball, changeup, slider)


#17 KRIS BRYANT
HITTING METRICS
EXIT VELOCITY 106.2 [MPH]
LAUNCH ANGLE 65 [DEG]
PROJECTED HANG TIME 6.5 [SEC]
PROJECTED DISTANCE 433 [FT]
STAT CAST
POWERED BY:
amazon web services

Exploratory Data Analysis

- Goal: Keep any interaction for which Statcast could provide valid measurements for the metrics `exit_velocity` and `launch_angle`.
  - In order to do so, “non-batted ball” records were excluded.



- Handling missing values (continued)
  - Drop all missing `events` which represent interactions where the batter did not make contact with the ball or the batter made contact with the ball but it was hit foul (e.g., ball, foul, called strike, swinging strike, blocked ball, foul tip, intentional ball, foul bunt, swinging strike blocked, missed bunt, pitchout).
  - Drop all missing `batted_ball_type` which represents other remaining events where there was not a batted ball (e.g., strikeout, walk, hit by pitch, intentional walk, caught stealing, etc.).
  - Drop missing values from `pitch_name`, `pitch_velocity`, `exit_velocity`, `launch_angle` columns.
  - Drop low frequency `pitch_name` observations

Analytic Data Set

- N = 385,848 rows
- 16 columns
- Defined outcomes for modeling
  - HRs: 4.3%!
  - XBHs: 11.4%
- Predictors for modeling (features)
  - Exit Velocity
  - Launch Angle
  - Pitch Velocity
  - Pitch Type (created as dummy variables)

Data Modeling

- Build and train a logistic regression model for classification
- Split X and y into training and testing sets
  - 70/30 split
- Compute baseline accuracy
  - HRs: 0.957
  - XBHs: 0.886

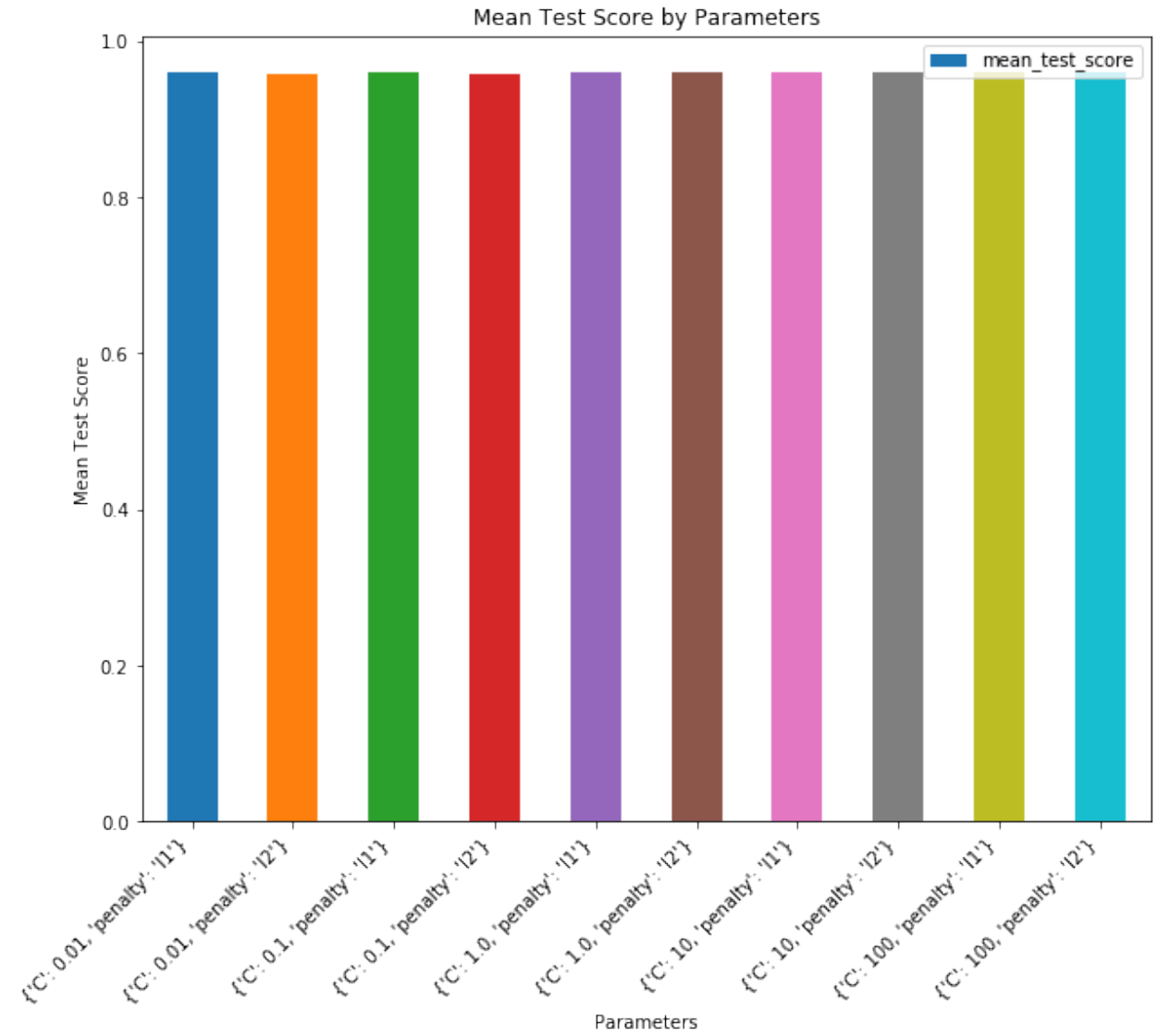
Data Modeling (continued)

- Use regularization to optimize model
- Use Grid Search (GridSearchCV) to perform exhaustive search over specified parameter values for the logistic estimator
  - Penalty parameter
  - Regularization strength parameter

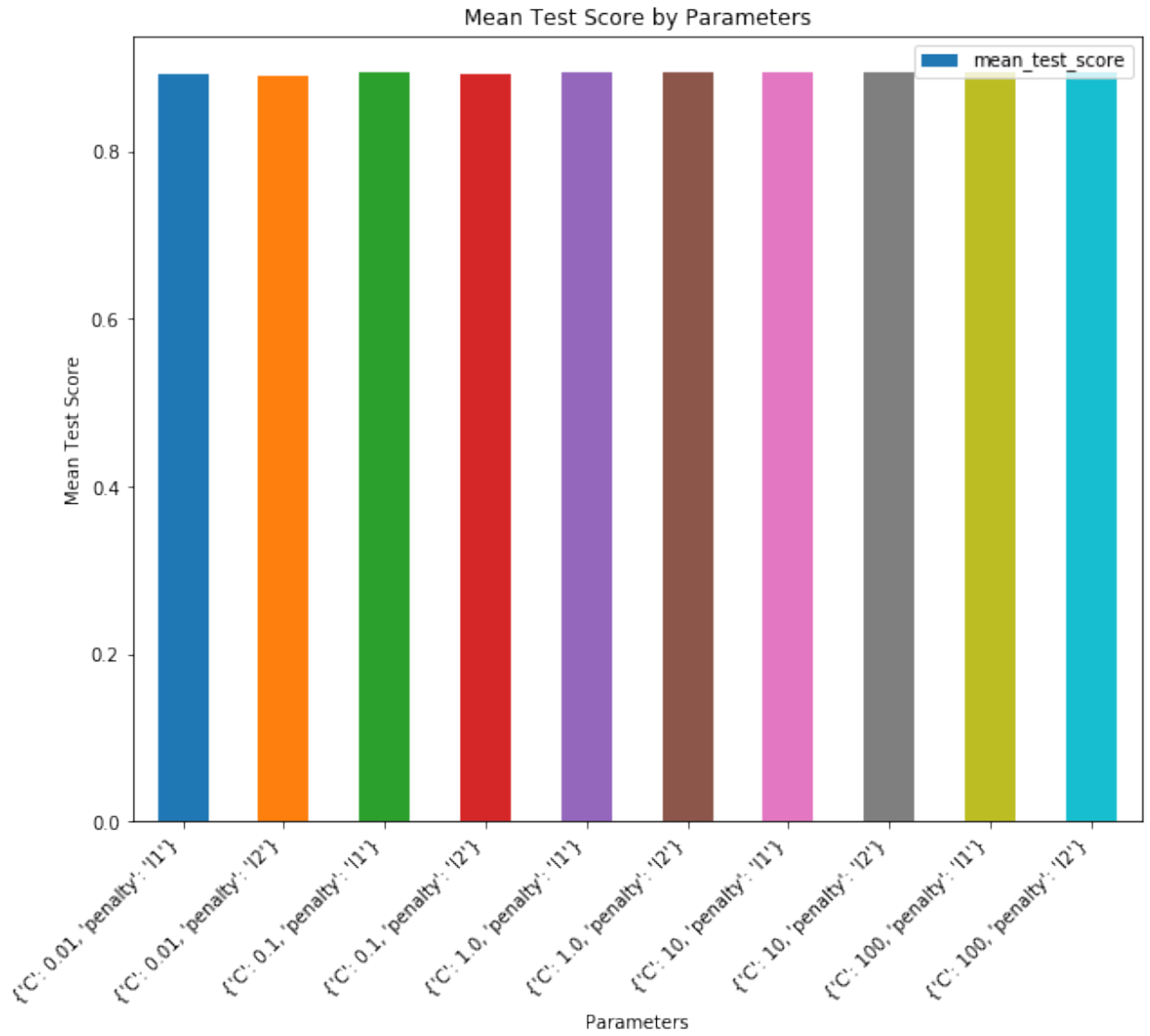
Grid Search Results

	0	1	2	3	4	5	6	7	8	9
param_C	0.01	0.01	0.1	0.1	1	1	10	10	100	100
param_penalty	l1	l2	l1	l2	l1	l2	l1	l2	l1	l2
params	{'C': 0.01, 'penalty': 'l1'}	{'C': 0.01, 'penalty': 'l2'}	{'C': 0.1, 'penalty': 'l1'}	{'C': 0.1, 'penalty': 'l2'}	{'C': 1.0, 'penalty': 'l1'}	{'C': 1.0, 'penalty': 'l2'}	{'C': 10, 'penalty': 'l1'}	{'C': 10, 'penalty': 'l2'}	{'C': 100, 'penalty': 'l1'}	{'C': 100, 'penalty': 'l2'}
mean_test_score	0.958462	0.956633	0.958896	0.958018	0.958925	0.958922	0.958914	0.958899	0.958903	0.958947
rank_test_score	8	10	7	9	2	3	4	6	5	**1**

HRs


Mean Test Score by Parameters
mean_test_score
Mean Test Score
1.0
0.8
0.6
0.4
0.2
0.0
{'C': 0.01, 'penalty': 'l1'}
{'C': 0.01, 'penalty': 'l2'}
{'C': 0.1, 'penalty': 'l1'}
{'C': 0.1, 'penalty': 'l2'}
{'C': 1.0, 'penalty': 'l1'}
{'C': 1.0, 'penalty': 'l2'}
{'C': 10, 'penalty': 'l1'}
{'C': 10, 'penalty': 'l2'}
{'C': 100, 'penalty': 'l1'}
{'C': 100, 'penalty': 'l2'}
Parameters

XBHs


Mean Test Score by Parameters
mean_test_score
Mean Test Score
0.8
0.6
0.4
0.2
0.0
{'C': 0.01, 'penalty': 'l1'}
{'C': 0.01, 'penalty': 'l2'}
{'C': 0.1, 'penalty': 'l1'}
{'C': 0.1, 'penalty': 'l2'}
{'C': 1.0, 'penalty': 'l1'}
{'C': 1.0, 'penalty': 'l2'}
{'C': 10, 'penalty': 'l1'}
{'C': 10, 'penalty': 'l2'}
{'C': 100, 'penalty': 'l1'}
{'C': 100, 'penalty': 'l2'}
Parameters

Grid Search Results (continued)

- HRs
  - Best parameter setting:
    C: 100, penalty: l2
  - Best mean cross-validated score:
    0.959

- XBHs
  - Best parameter setting:
    C: 100, penalty: l1
  - Best mean cross-validated score:
    0.894

Testing Accuracy Score

- Uses the score defined by the best estimator from the grid search results

- HRs
  - Score:
    0.958

- XBHs
  - Score:
    0.894

Limitations

- Large class imbalance in outcomes
  - —> Unfortunately, obtaining additional data that becomes available in the future is unlikely to resolve the imbalance because the outcomes are typically rare events, especially HRs.

Next Steps

- Explore the effect of resampling on the majority class so that there is more balance within classes

Questions?