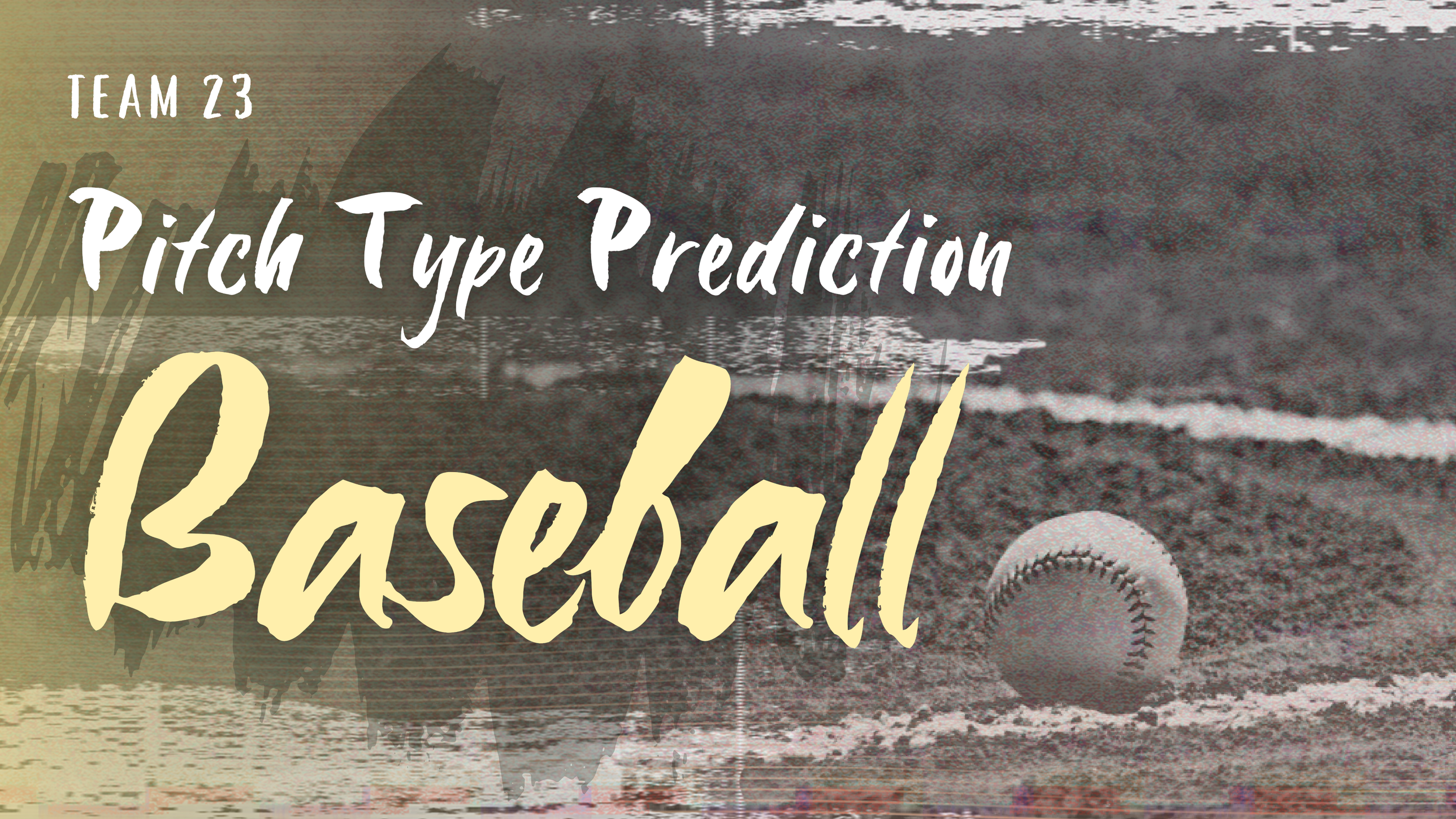
TEAM 23

Pitch Type Prediction

Baseball

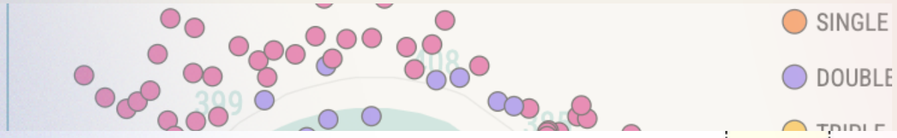


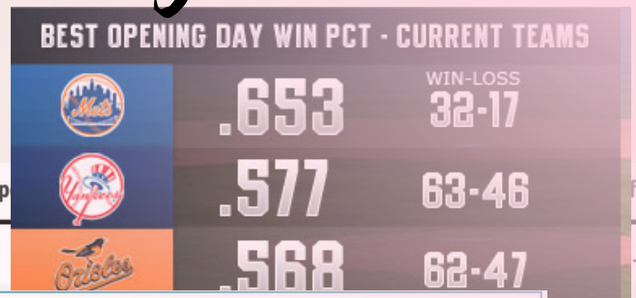

Using RNN model

Outline

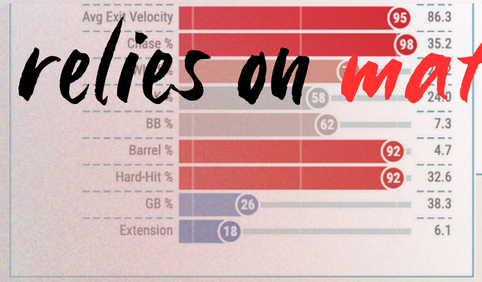
- Brief Introduction
- Question

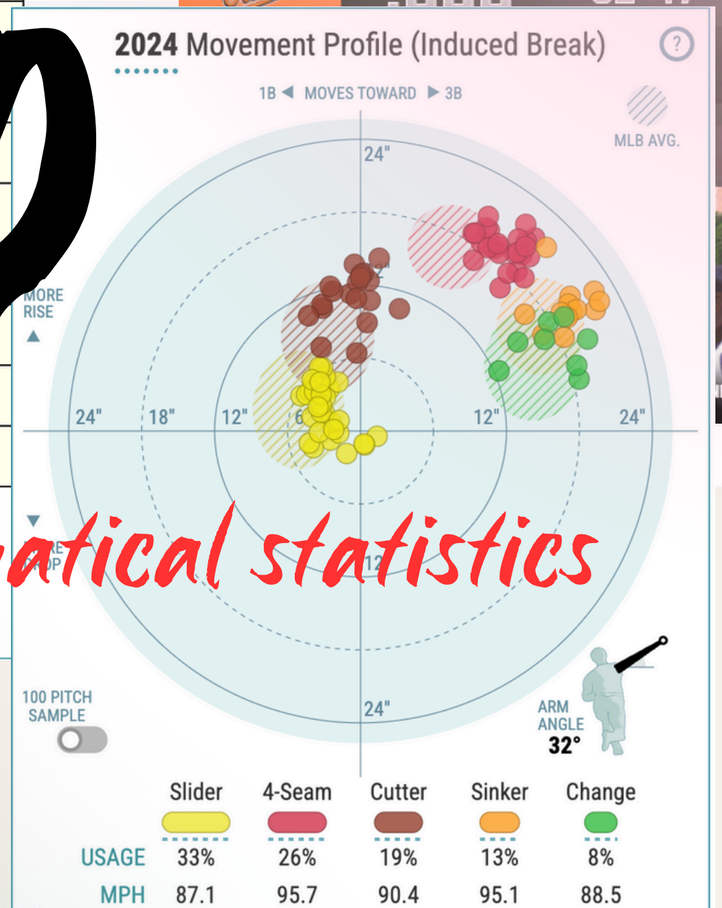
Analytics Nowadays?





Rk.	Player	Year	BF	K%	BB%	wOBA	xwOBA	LA Sweet-Sp	ff %	Swing %
1	Skenes, Paul	2024	514	33.1	6.2	.247	.248	27.8	.7	49.8
2	Glasnow, Tyler	2024	522	32.2	6.7	.258	.254		.9	48.4
3	Skubal, Tarik	2024	753	30.3	4.6	.246			.9	51.8
4	Steele, Justin	2024	555	24.3	6.7	.271			.7	51.5
5	Wheeler, Zack	2024	787	28.5	6.6	.256			.5	51.2
6	Sale, Chris	2024	702	32.1	5.6	.260			.0	49.0
7	Crochet, Garrett	2024	595	35.1	5.5	.280			.1	53.2
8	Ryan, Joe	2024	538	27.3	4.3	.282	.264		.6	50.9
9	Greene, Hunter	2024	611	27.7	9.3	.265	.271		29.7	50.6



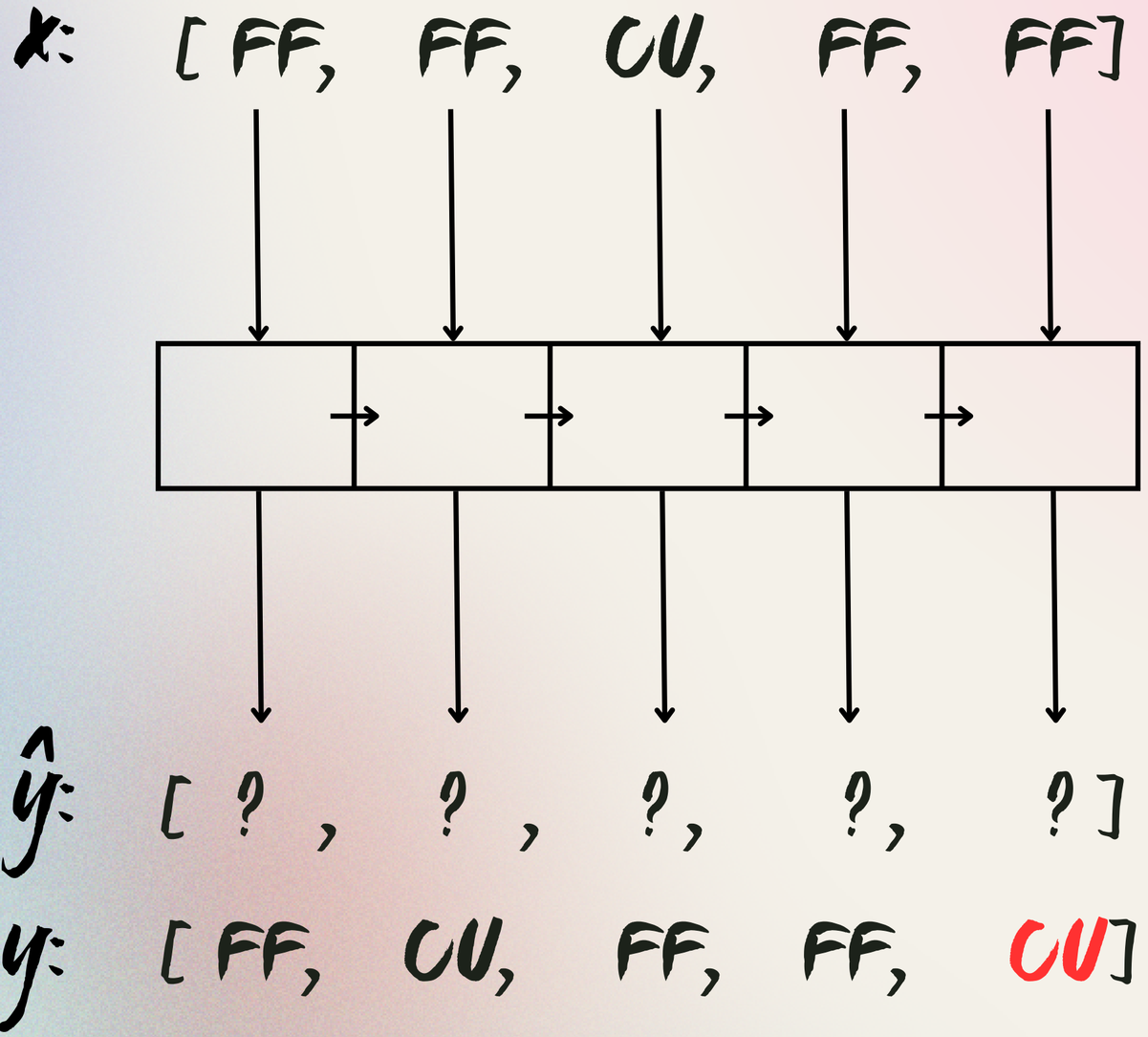


relies on mathematical statistics

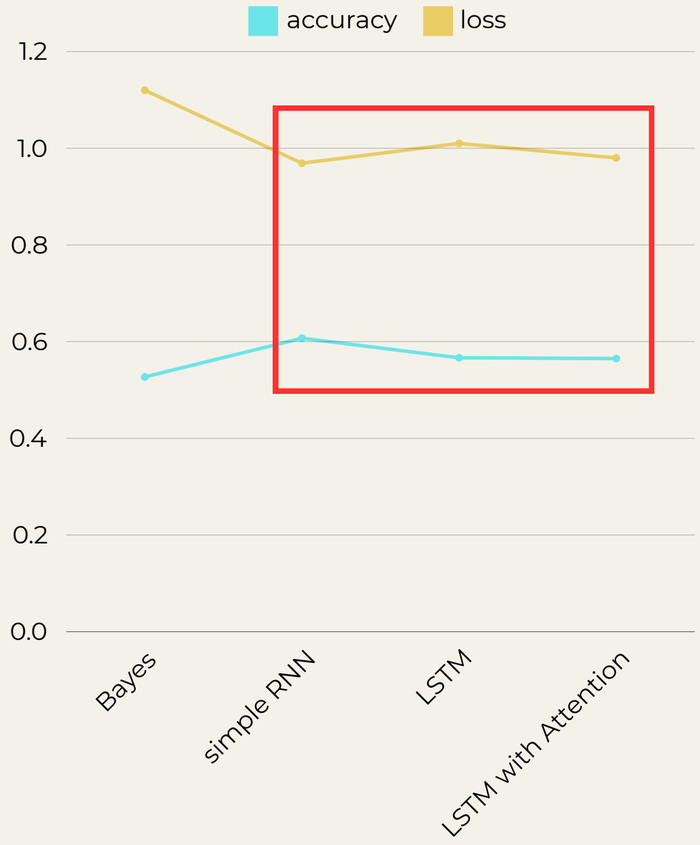
"We all think that it may overlook game context and abstract features like pitcher-catcher synergy or strategy shifts based on the count."

Use RNN model to learn the relationship between pitches

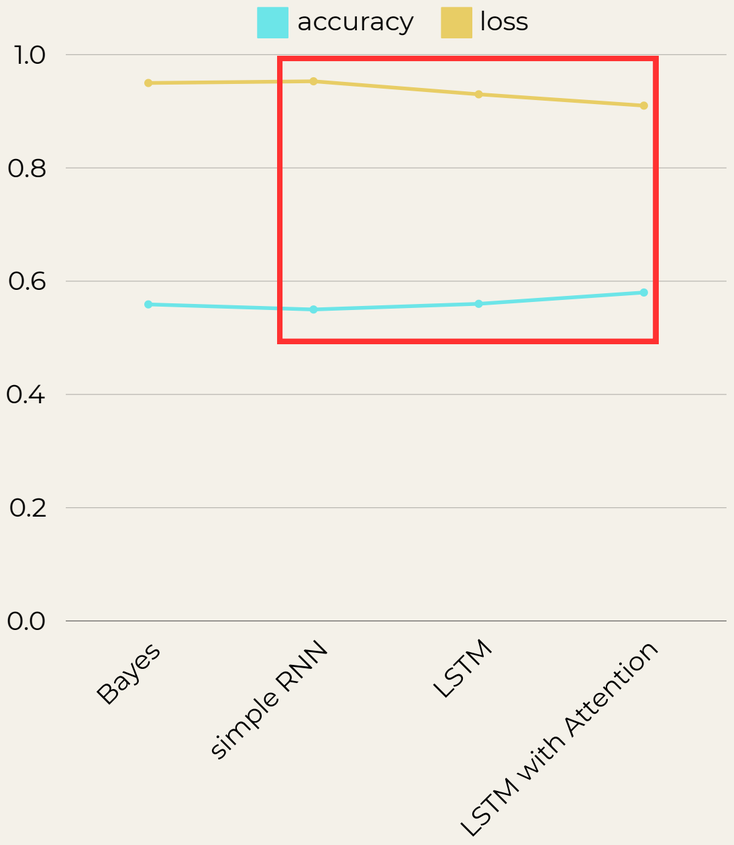
(and lots of other features...)



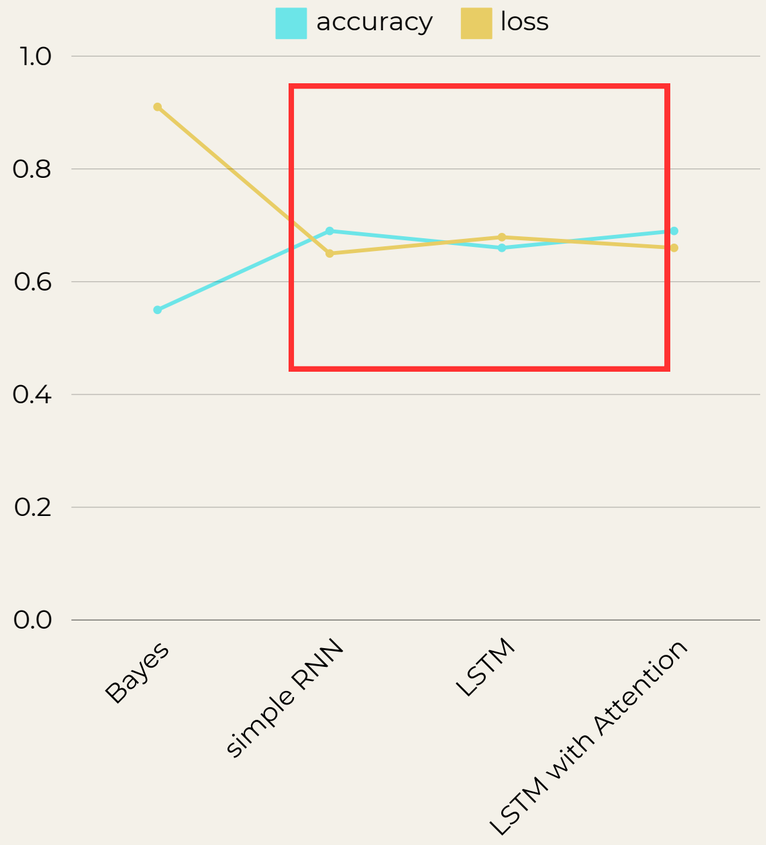
After comparing four versions of model...



SHOTA IMANAGA

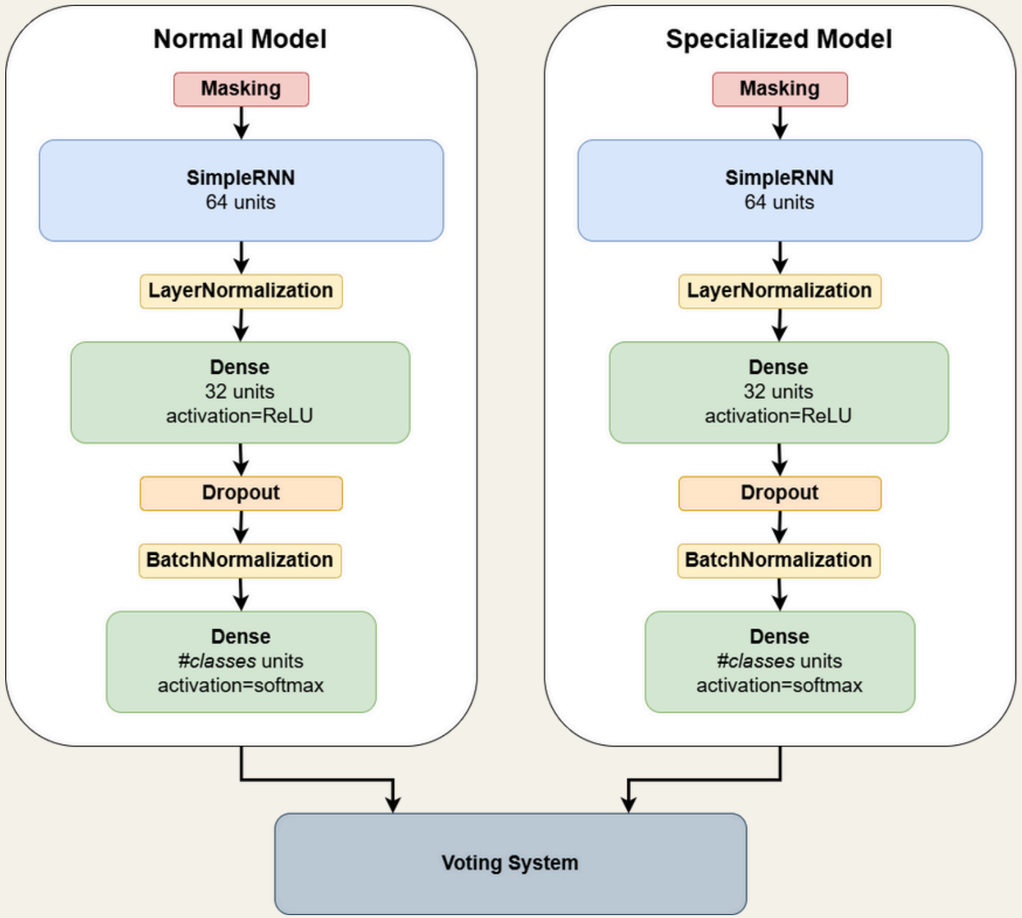


HUNTER GREENE

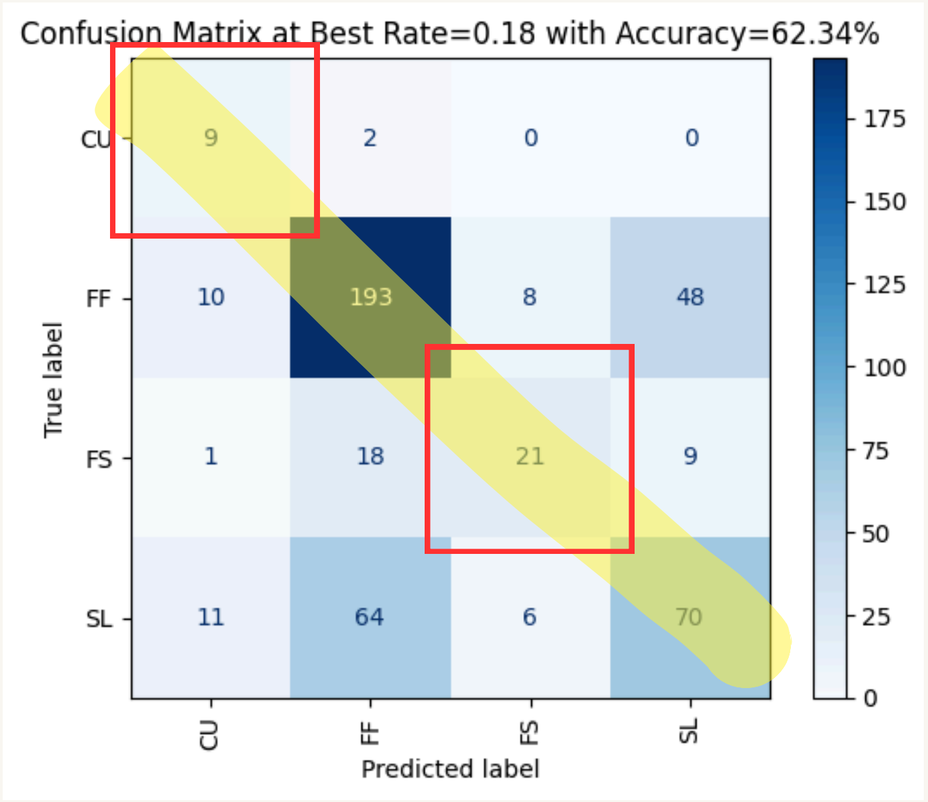


MASON MILLER

FINAL VERSION MODEL HAS THE HIGHEST ACCURACY



SHOTA IMANAGA



RELIEF PITCHER

RELIEF PITCHER

RELIEF PITCHER

MASON MILLER

Our accuracy of predicting pitch type in relief pitcher is about...

69% > 55%

ACCURACY WHICH WAY HIGHER THAN BASELINE PREDICTING BY BAYES CLASSIFIER

Practical Value of our Model:

Provides a powerful tool to analyze the pitching tendencies

Pitchers

Coaches

Catchers

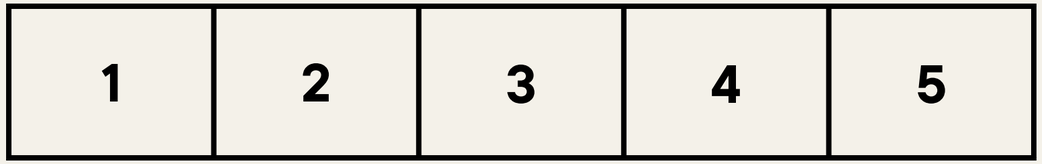

1. Data Processing

Q1. I didn't understand the Data Slicing part to add more data to the limited ones, can you explain it in a more specific way to explain the technique you used and how it increases the data size?

Data Slicing: Take an example

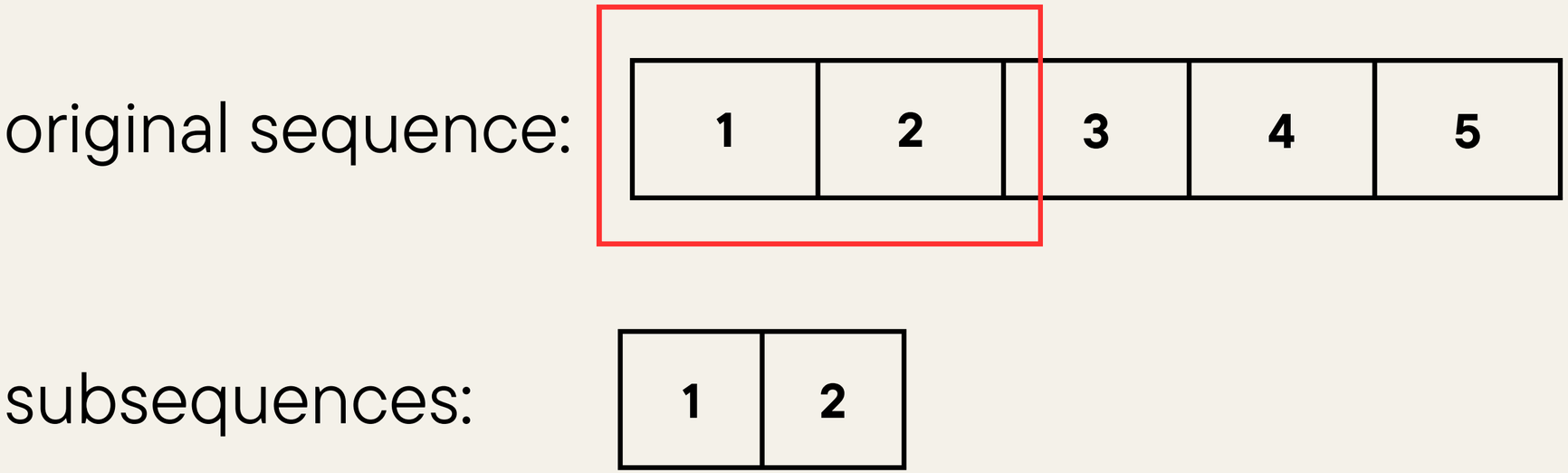
input: min = 2, max = 3

original sequence:



Data Slicing

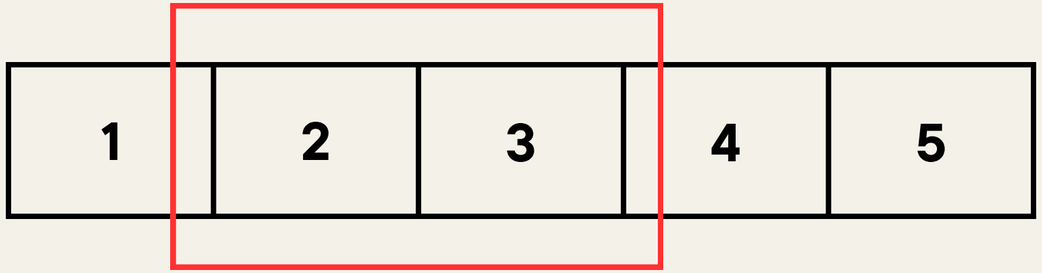
length = 2



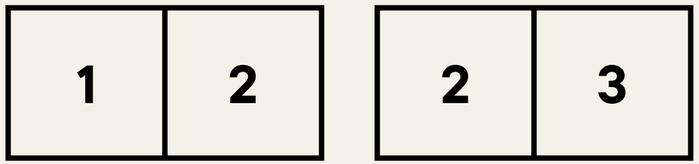
Data Slicing

length = 2

original sequence:



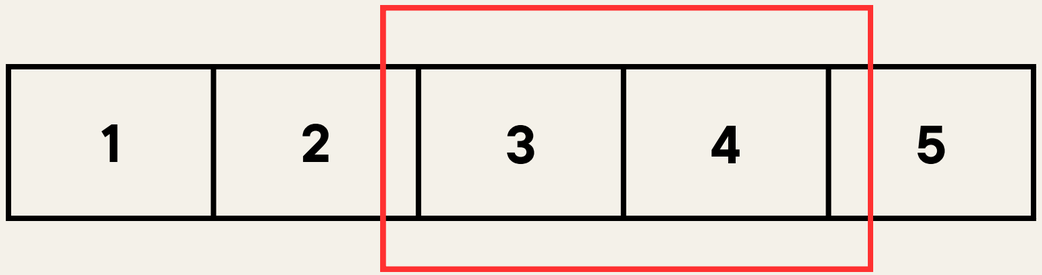
subsequences:



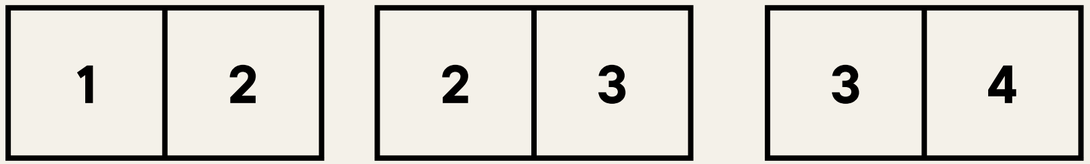
Data Slicing

length = 2

original sequence:



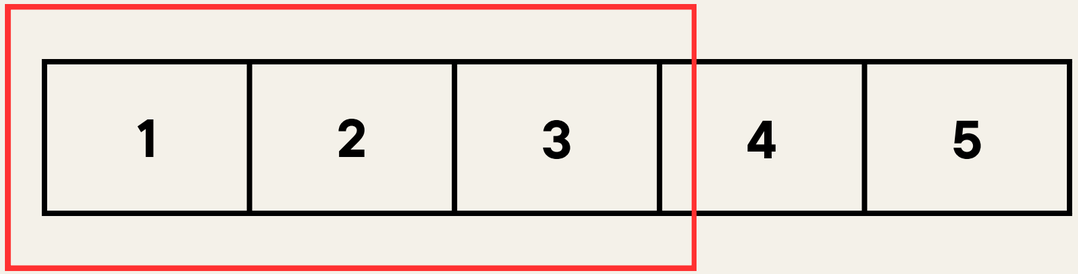
subsequences:



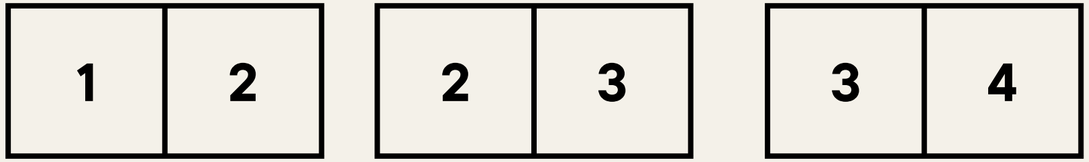
Data Slicing

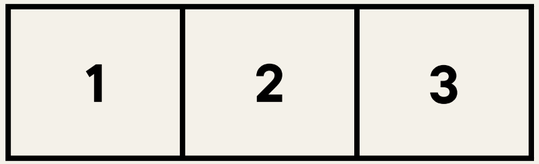
length = 3

original sequence:



subsequences:

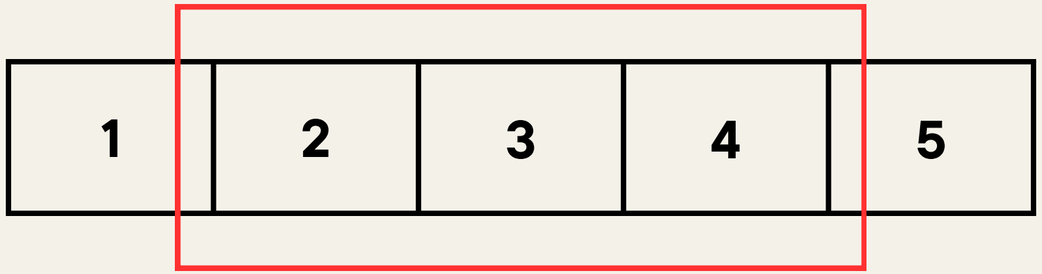




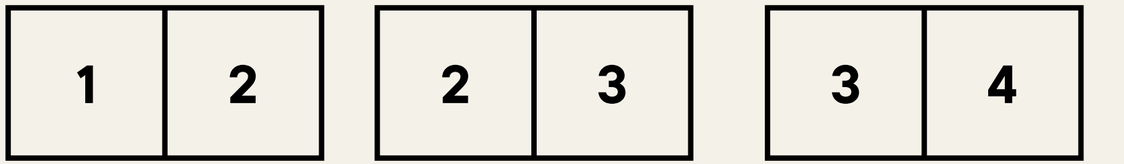
Data Slicing

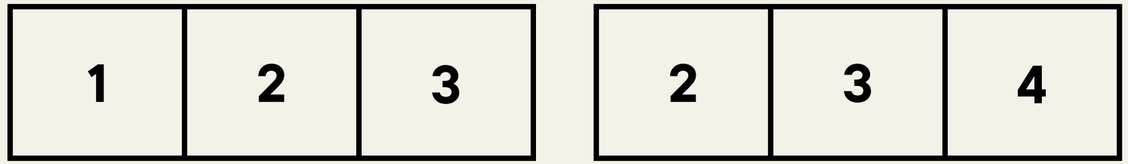
length = 3

original sequence:



subsequences:





1

⋮

5

By doing this,

Take the case of Shota Imanaga as example.

original sequences: subsequences:

2590 → 4773

Q2. Why don't you use data augmentation to generate additional data to balance the data distribution?

Why not data augmentation

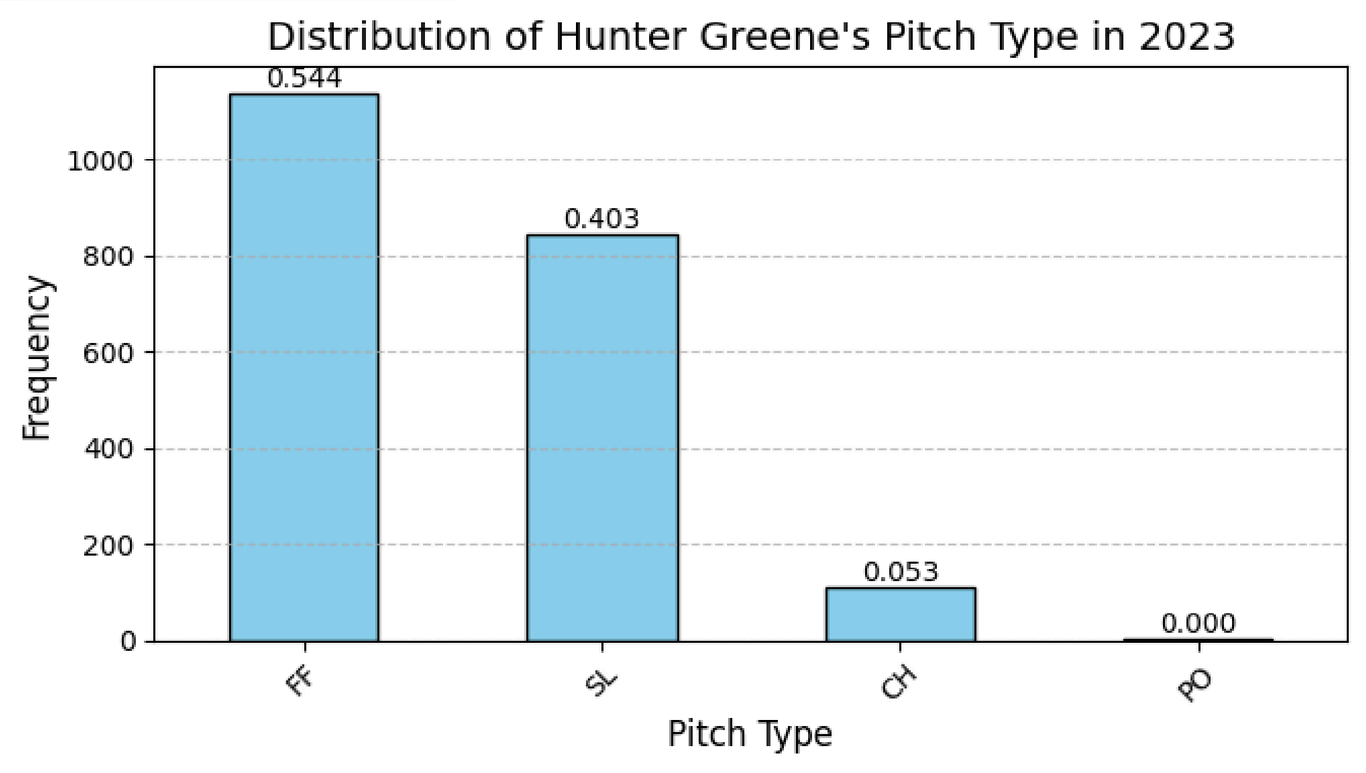
1. tried SMOTE-TS (Synthetic Minority Oversampling Technique for Time Series).
    - no significant improvements.
2. Data is based on many real-world baseball dynamic aspects.
    - Naive Augmentation might lose the relationship between pitches.

Q3. You mentioned adding the FREQUENCY FEATURE. I am curious about when you calculate the frequency—do you calculate it after splitting the data into train, validation, and test sets?

How do you prevent a situation where the distributions of the train and validation sets are significantly different?

Frequency Feature?

- Basic knowledge of the pitcher
- The frequency data are from previous year



Q4. I noticed that the length of the subsequences you use for data slicing ranges from two to six.

Would it be better to use longer sequences for RNN training?

- We have tried to cut the sequence by batter and by Date.
- the size of the subsequences: Batter:2~6, Date: 20 ~ 30.
- We find out Batter Cutting Scheme with smaller performs better.

Q5. In the raw data structure, is there any data column(feature) about the batter or the catcher, apart from the batter's ID? Since we think the batter and the catcher might influence the pitch type thrown by pitcher.

Q6. In the current baseball matchup, the pitcher's strategy is closely tied to the catcher's decisions. Have you incorporated this relationship into your model during training?

jldbc/**pybaseball**

Pull current and historical baseball statistics using Python (Statcast, Baseball Reference, FanGraphs)

44 Contributors | 392 Used by | 1k Stars | 339 Forks

fielder_2.1: catcher id

```
Index(['pitch_type', 'game_date', 'release_speed', 'release_pos_x',
       'release_pos_z', 'player_name', 'batter', 'pitcher', 'events',
       'description', 'spin_dir', 'spin_rate_deprecated',
       'break_angle_deprecated', 'break_length_deprecated', 'zone', 'des',
       'game_type', 'stand', 'p_throws', 'home_team', 'away_team', 'type',
       'hit_location', 'bb_type', 'balls', 'strikes', 'game_year', 'pfx_x',
       'pfx_z', 'plate_x', 'plate_z', 'on_3b', 'on_2b', 'on_1b',
       'outs_when_up', 'inning', 'inning_topbot', 'hc_x', 'hc_y',
       'tfs_deprecated', 'tfs_zulu_deprecated', 'fielder_2', 'umpire', 'sv_id',
       'vx0', 'vy0', 'vz0', 'ax', 'ay', 'az', 'sz_top', 'sz_bot',
       'hit_distance_sc', 'launch_speed', 'launch_angle', 'effective_speed',
       'release_spin_rate', 'release_extension', 'game_pk', 'pitcher.1',
       'fielder_2.1', 'fielder_3', 'fielder_4', 'fielder_5', 'fielder_6',
       'fielder_7', 'fielder_8', 'fielder_9', 'release_pos_y',
       'estimated_ba_using_speedangle', 'estimated_woba_using_speedangle',
       'woba_value', 'woba_denom', 'babip_value', 'iso_value',
       'launch_speed_angle', 'at_bat_number', 'pitch_number', 'pitch_name',
       'home_score', 'away_score', 'bat_score', 'fld_score', 'post_away_score',
       'post_home_score', 'post_bat_score', 'post_fld_score',
       'if_fielding_alignment', 'of_fielding_alignment', 'spin_axis',
       'delta_home_win_exp', 'delta_run_exp'],
      dtype='object')
```

jldbc/**pybaseball**

Pull current and historical baseball statistics using Python (Statcast, Baseball Reference, FanGraphs)

44 Contributors | 392 Used by | 1k Stars | 339 Forks

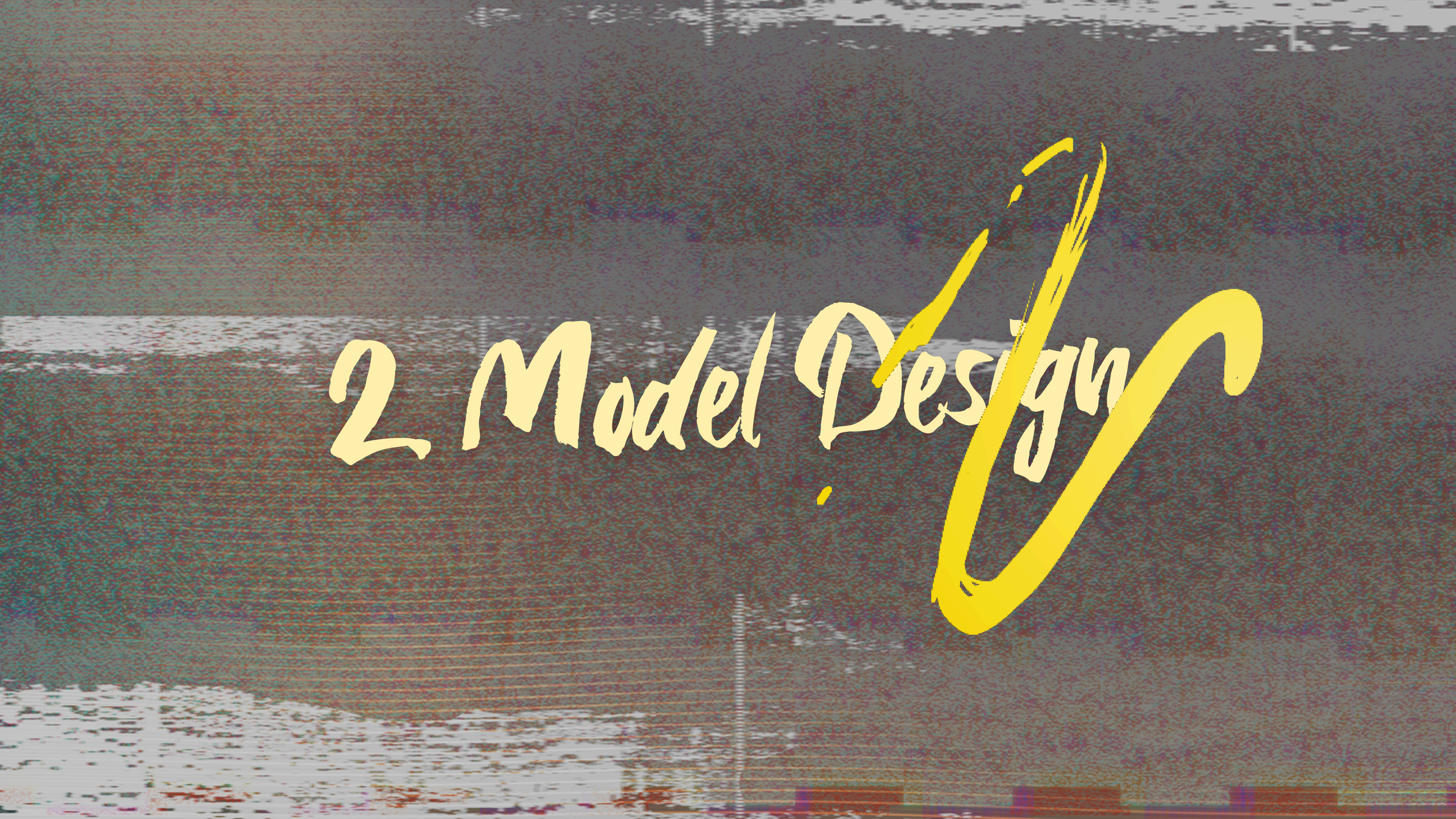
batter id, hit_distance...

```
Index(['pitch_type', 'game_date', 'release_speed', 'release_pos_x',
       'release_pos_z', 'player_name', 'batter', 'pitcher', 'events',
       'description', 'spin_dir', 'spin_rate_deprecated',
       'break_angle_deprecated', 'break_length_deprecated', 'zone', 'des',
       'game_type', 'stand', 'p_throws', 'home_team', 'away_team', 'type',
       'hit_location', 'bb_type', 'balls', 'strikes', 'game_year', 'pfx_x',
       'pfx_z', 'plate_x', 'plate_z', 'on_3b', 'on_2b', 'on_1b',
       'outs_when_up', 'inning', 'inning_topbot', 'hc_x', 'hc_y',
       'tfs_deprecated', 'tfs_zulu_deprecated', 'fielder_2', 'umpire', 'sv_id',
       'vx0', 'vy0', 'vz0', 'ax', 'ay', 'az', 'sz_top', 'sz_bot',
       'hit_distance_sc', 'launch_speed', 'launch_angle', 'effective_speed',
       'release_spin_rate', 'release_extension', 'game_pk', 'pitcher.1',
       'fielder_2.1', 'fielder_3', 'fielder_4', 'fielder_5', 'fielder_6',
       'fielder_7', 'fielder_8', 'fielder_9', 'release_pos_y',
       'estimated_ba_using_speedangle', 'estimated_woba_using_speedangle',
       'woba_value', 'woba_denom', 'babip_value', 'iso_value',
       'launch_speed_angle', 'at_bat_number', 'pitch_number', 'pitch_name',
       'home_score', 'away_score', 'bat_score', 'fld_score', 'post_away_score',
       'post_home_score', 'post_bat_score', 'post_fld_score',
       'if_fielding_alignment', 'of_fielding_alignment', 'spin_axis',
       'delta_home_win_exp', 'delta_run_exp'],
      dtype='object')
```

woba:
Weighted On Base Average（加權上壘率）

babip:
Batting Average on Balls put Into Play（場內被安打率）

iso:
Isolated Power（純長打率）

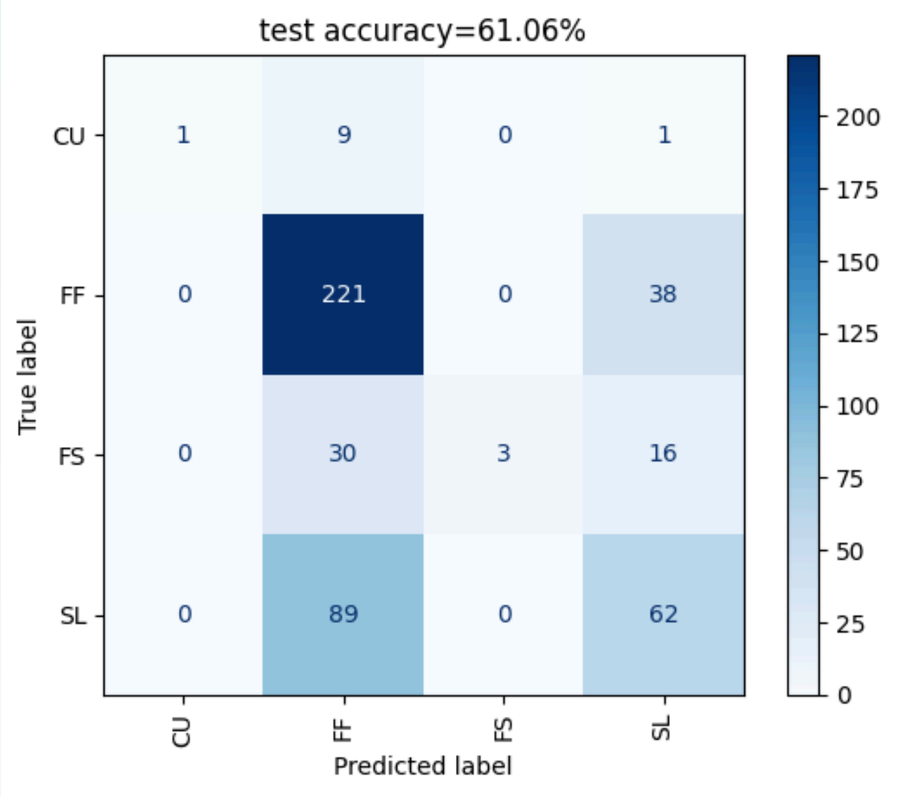

2 Model Design

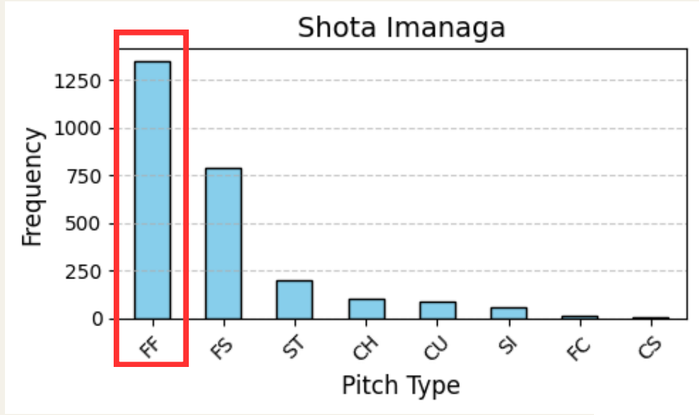
Q7. In the discussion part, you notice that the prediction of the normal model focuses only on fastballs, and you decide to add the specialist model to normal RNN ones and create a voting system that shows a slightly better performance.

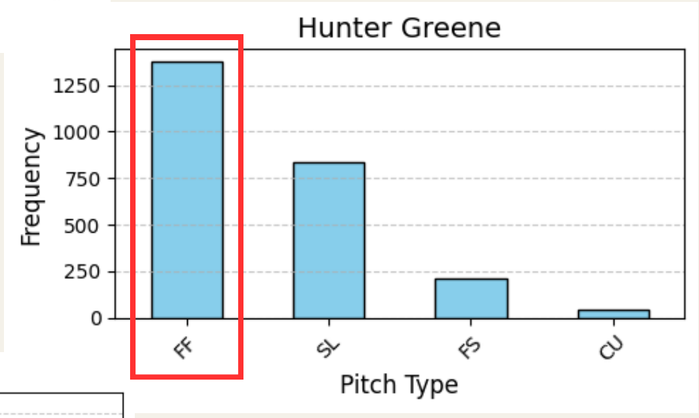
Q7. (cont.) I also noticed in that part, in which you compare the original ones with the new ones (with the specialist model), that the two confusion matrices didn't have the same amount of ground truth, I believe that adding the specialist model does gain some improvement but maybe compare it with the same amount of data that can get a clearer vision of the improvement?

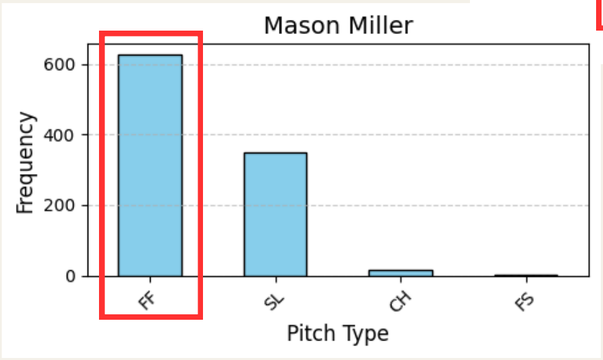
Accuracy and Loss Are Not Enough: Confusion Matrix

Hunter Greene: 61.06%

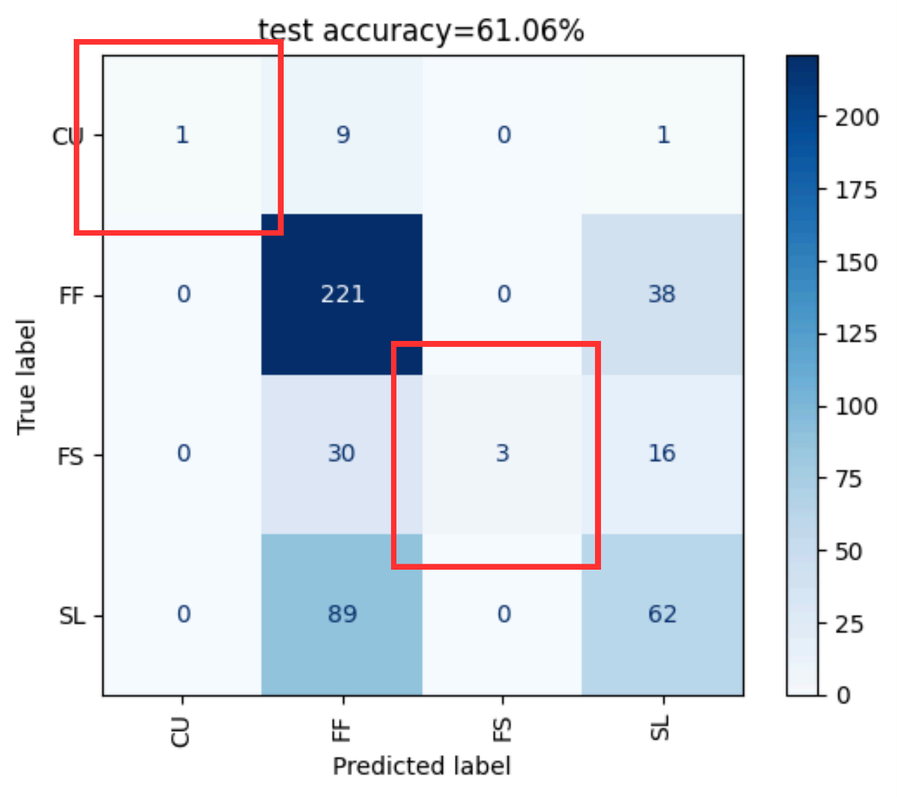




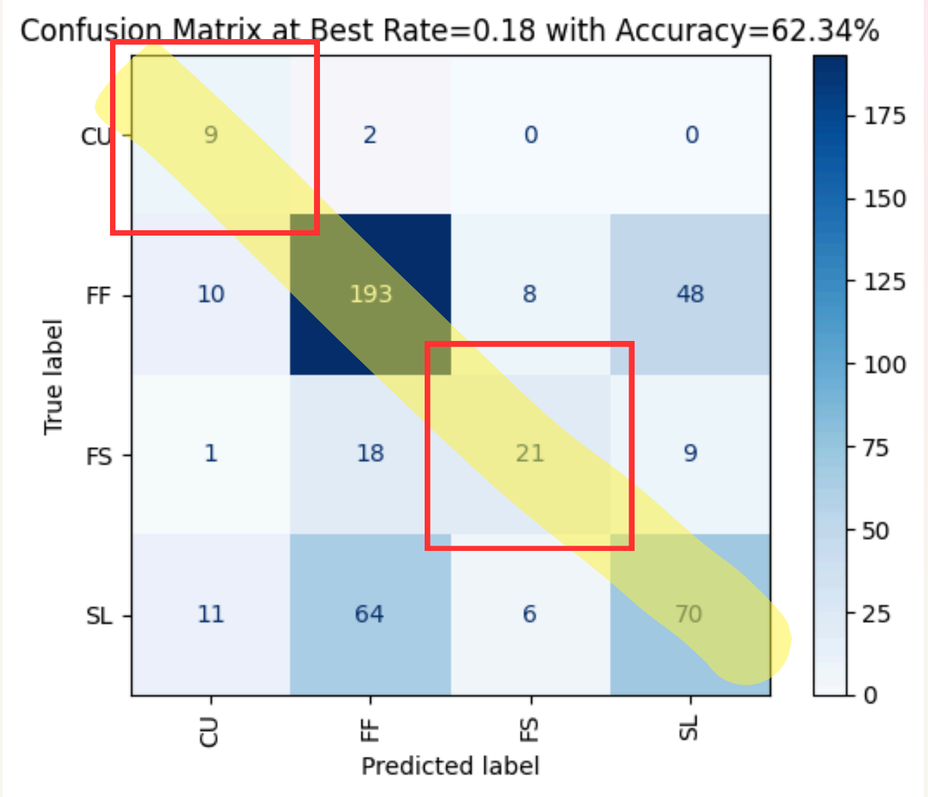




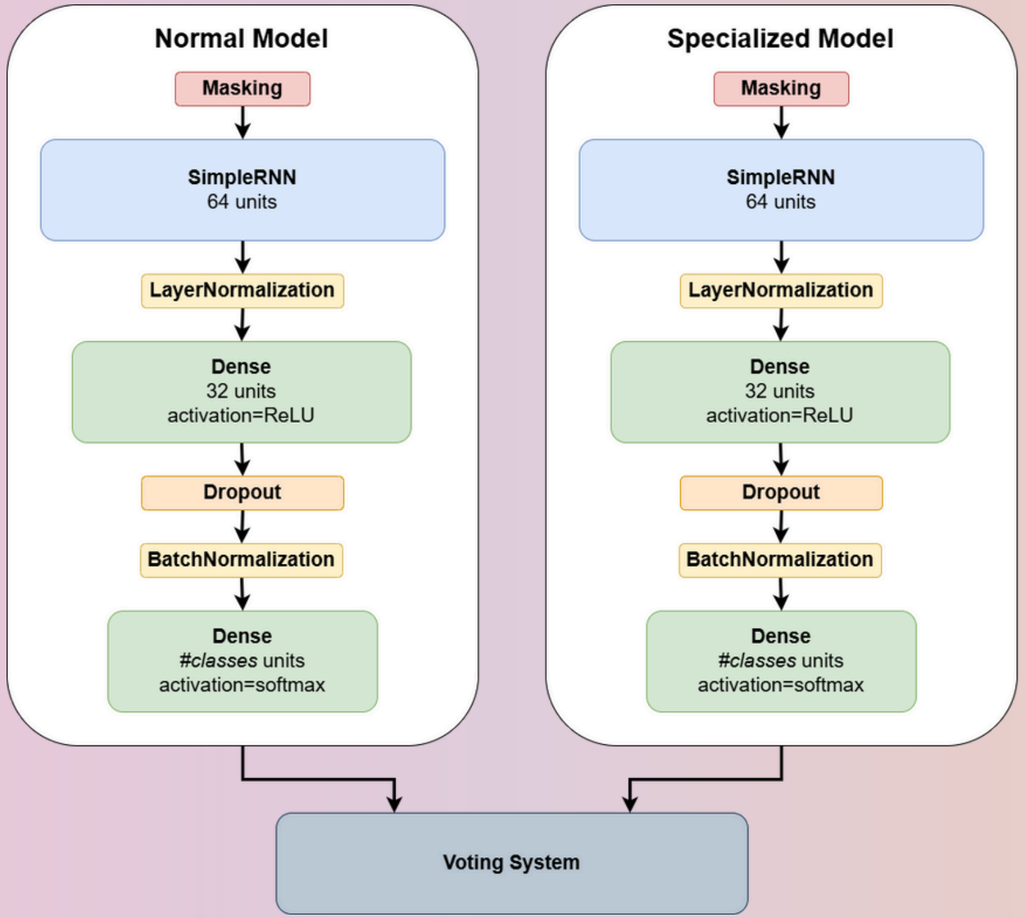
Simple RNN: 61.06%



Adding Specialist Model: 62.34%



Q8. Can you explain why you choose only two base learners in a voting system with weighted averaging ?



Why 2 Base Learners in Voting System

1. The word "voting" here is a bit informal. A more precise term would be "ensemble".
2. We chose two base learners for generalizability
    - One for all pitch types
    - One for less common pitch types (excluding the 2 most frequent FS and CU)

3. Result Analysis

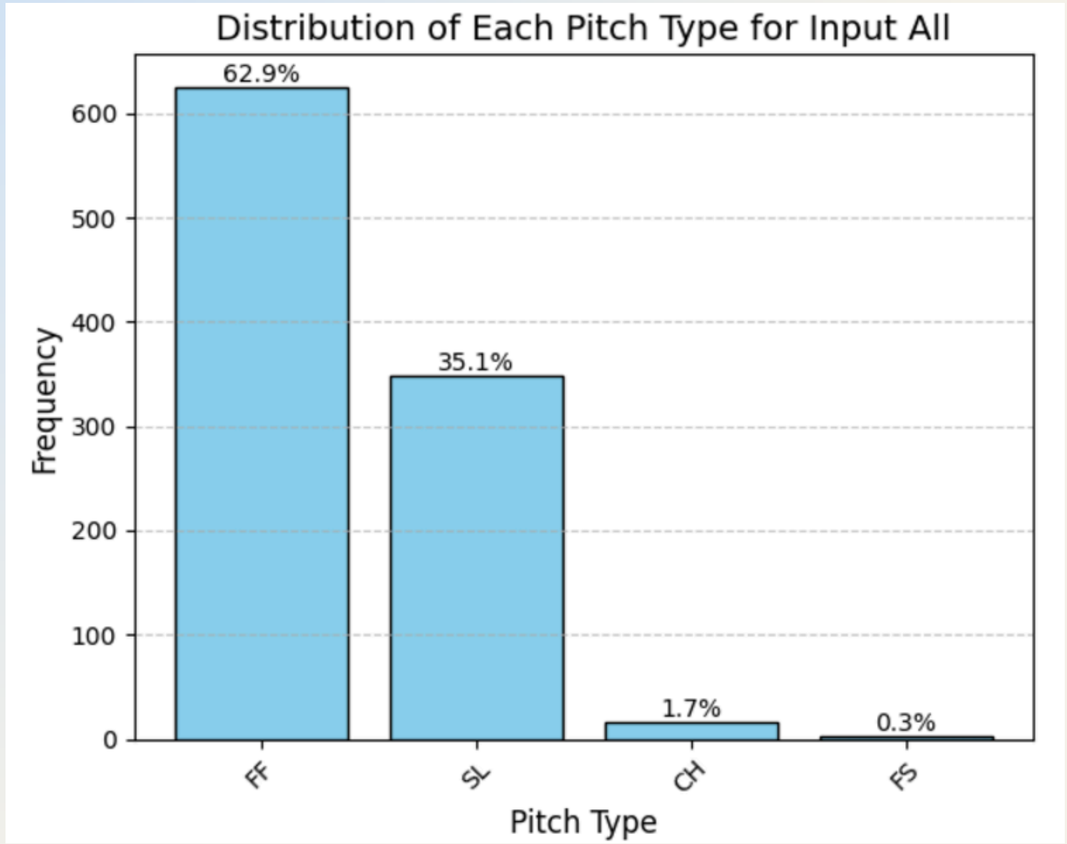
Q9. can you explain again why you think the ones with fewer pitch types have better results, and also in the video, I noticed that the Miller one has better results than the other two, could you also discuss this part to show the possible reason that Miller's stand out?

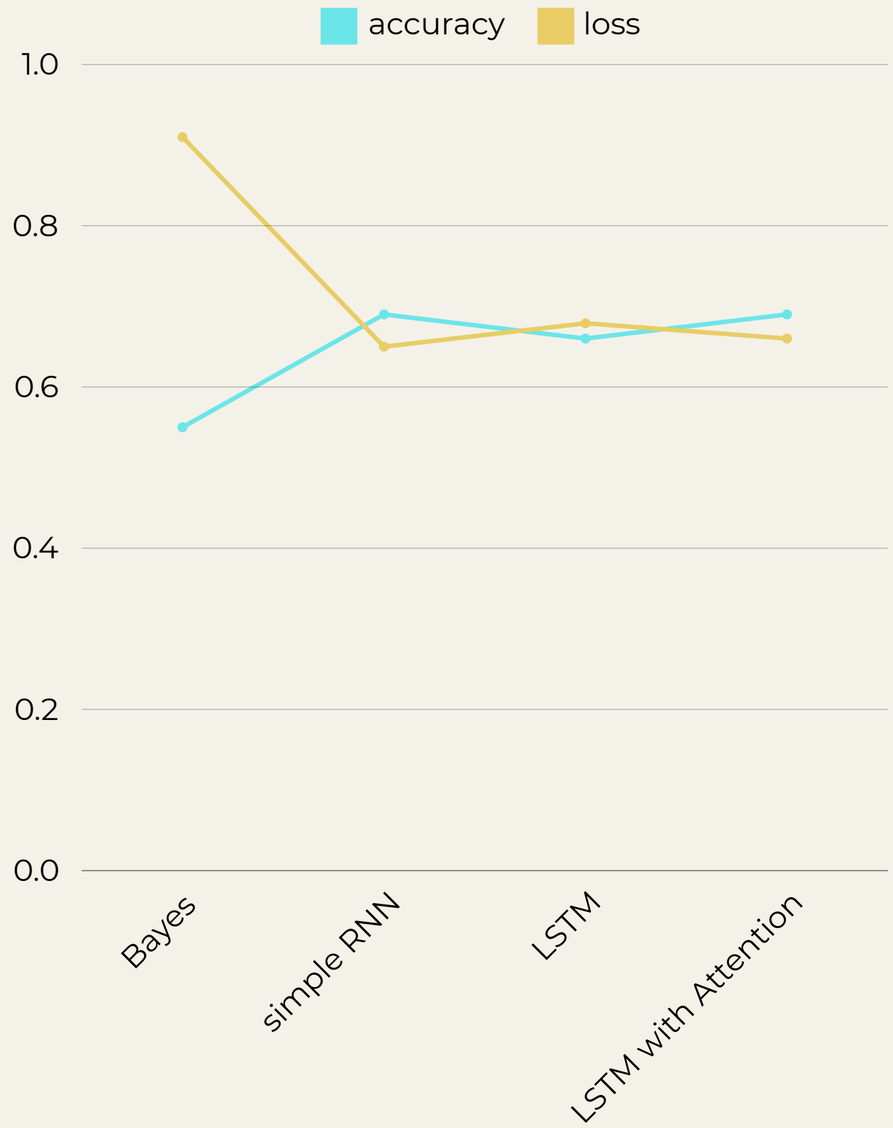
When a pitcher has a limited variety of pitch types, the model has fewer categories to distinguish, leading to higher accuracy.

Miller's better results can be attributed to his fewer pitch types, which generally improves accuracy.

Q10. Is 60% a good accuracy compared to the accuracy of easier guessing methods?

MASON MILLER






THANKS FOR LISTENING
TEAM23