Stanford | Computer Science

"This Game is in the Refrigerator": Predicting NBA Game Outcomes

Jesse A. Rodriguez
Department of Mechanical Engineering, Stanford University, Stanford CA USA
jrodrig@stanford.edu

ABSTRACT

Three NN architectures are proposed along with a novel feature design to predict the outcomes and popular betting metrics of NBA basketball games. The most predictive feature design consisted of the statlines of the top 3 players of each team in their last 4 games. The binary classifier NN performed the best in predicting game outcomes, achieving 59.8% accuracy on the test set. The exponential score predictor achieved 59.1% accuracy and reproduced the score distribution of the test set well, indicating that it may be effective in predicting the over/under for NBA games. The softmax score predictor only achieved 57.6% test accuracy but it managed to reproduce the margin of victory distribution well. Ultimately, a mismatch between the train set and the dev/test sets as well as an overall lack of data likely led to the unremarkable performance of the three models.

DATASET AND FEATURES

VISITOR: Los Angeles Lakers (20-3)

	POS	MIN	FG	FGA	3P	3PA	FT	FTA	OR	DR	TOT	A	PF	ST	TO	BS	+/-	PTS
23 LeBron James	F	33:45	11	23	4	9	5	7	1	6	7	8	1	0	3	1	21	31
3 Anthony Davis	F	32:09	12	21	2	6	13	15	0	9	9	2	2	2	3	3	6	39
7 JaVale McGee	C	15:25	6	7	0	0	1	1	1	2	3	0	3	0	0	2	5	13
14 Danny Green	G	21:08	1	3	1	3	0	0	0	5	5	2	2	1	1	1	5	3
1 Kentavious Caldwell-Pope	G	27:01	1	3	1	3	2	2	0	0	0	5	1	1	0	0	8	5
9 Rajon Rondo		14:32	2	2	2	2	0	0	0	3	3	3	0	2	2	1	13	6
0 Kyle Kuzma		24:26	6	13	3	6	0	0	0	6	6	1	2	0	2	0	19	15
30 Troy Daniels		12:37	3	5	3	5	0	0	0	1	1	1	3	1	1	0	7	9
4 Alex Caruso		23:46	2	4	1	1	3	4	0	1	1	3	4	1	1	0	13	8
39 Dwight Howard		21:20	1	1	0	0	3	4	1	9	10	0	5	2	2	0	14	5
2 Quinn Cook		10:07	1	3	0	1	0	0	0	0	0	1	0	0	1	0	0	2
10 Jared Dudley		03:44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0

Figure 1: Example of a box score containing many of the stats utilized in this project.

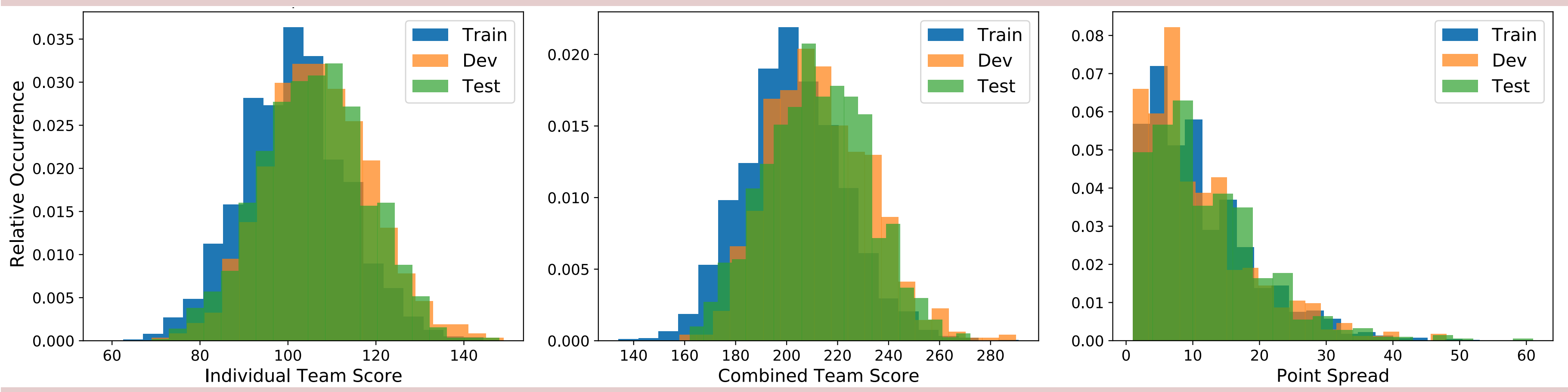


Figure 2: Histograms for individual team scores (left), combined scores (middle) and point spreads (right) from the NBA game corpus used in this project.

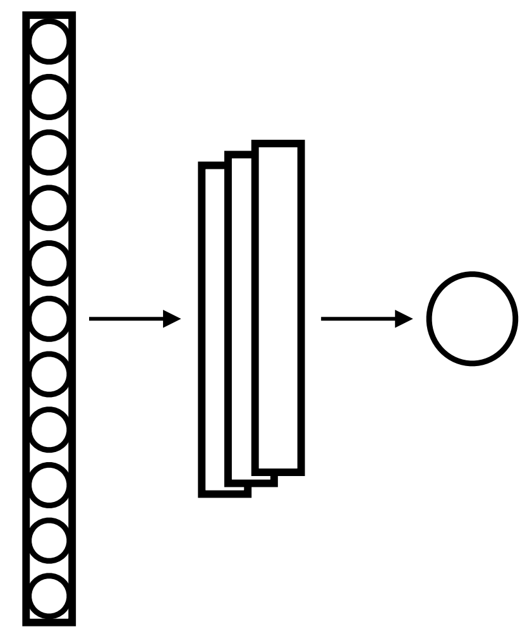
$$\underbrace{[\ \underbrace{111}_{\text{home score}}\ ,\ \underbrace{102}_{\text{away score}}\ ]}_{\text{label}}\ \underbrace{[\ \underbrace{29}_{\text{THP PTS}}\ ,\ \underbrace{8}_{\text{THP REB}}\ ,\ \underbrace{5}_{\text{THP AST}}\ ,\ \ldots\ \underbrace{17}_{\text{2HP PTS}}\ ,\ \underbrace{3}_{\text{2HP REB}}\ ,\ \ldots\ \underbrace{26}_{\text{TAP PTS}}\ ,\ \underbrace{4}_{\text{TAP REB}}\ ,\ \ldots]}_{\text{feature vector}}$$

Figure 3: Example of an non-standardized feature vector for $n_g = 1$ and $n_p = 2$, where THP is top home player, 2HP is second-ranked home player, and TAP is top away player, and the stats present are the statlines for the players in question from the last game that they played

- The base dataset for this work is the complete statline for every player in each game from the 2012-2013 NBA season to the 2017-2018 season.
- A 80/10/10 split for train/dev/test was chosen, which results in dataset sizes of about ~5800/725/725 games depending on which feature parameters are chosen. The train set is the first 80% chunk of the games by chronological order and the test/dev set examples are sampled uniformly from the remaining 20%.
- Each feature vector contains the statlines of the top n_p players (ranked by Pts) that are on the roster for the game in question in their previous n_g games for each team.

NN ARCHITECTURES

Binary Classifier



Sigmoid Output Activation

$$a^{[L]} = \frac{1}{1+e^{-Z}}$$

Binary Cross Entropy Loss

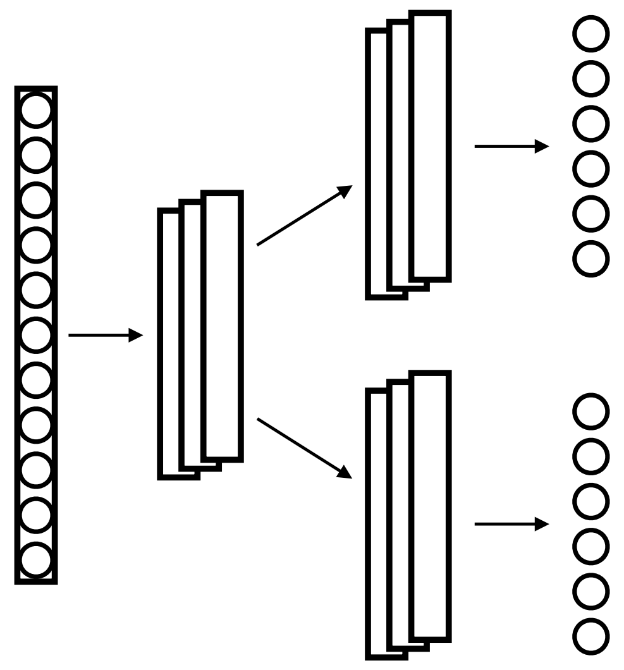
$$J = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\textbf{log}\hat{y}^{(i)} + (1-y^{(i)})\textbf{log}(1-\hat{y}^{(i)})\right)$$

Number of Nodes

$$[n_x; 2; 5; 5; 1]$$

Figure 4: Binary classifier fully-connected neural network with 3 hidden layers and a sigmoid output layer that utilizes the binary cross entropy loss.

Softmax Score Predictor



Softmax Output Activation

$$a_k^{[L]} = \frac{e^{Z_k}}{\sum_i e^{Z_i}}$$

Categorical Cross Entropy Loss

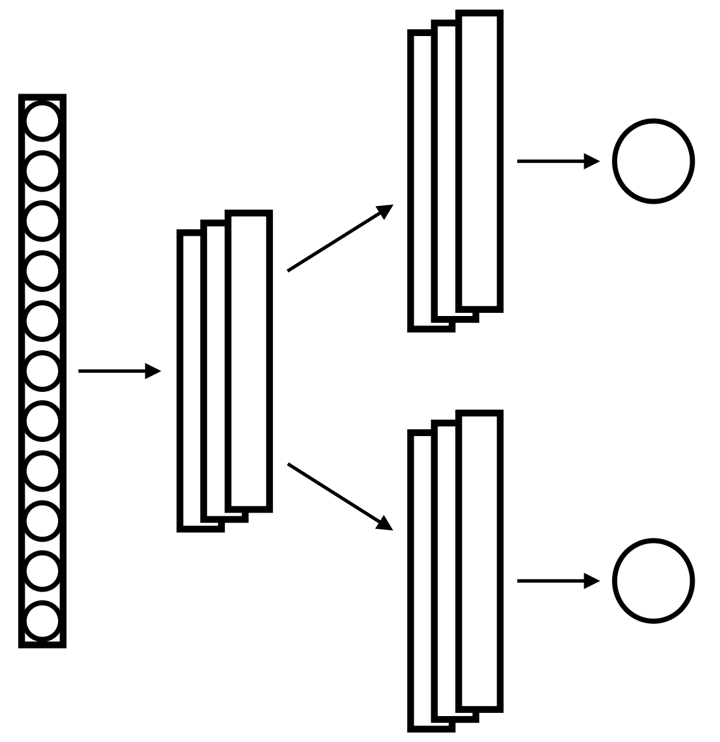
$$J = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\textbf{log}\hat{y}^{(i)}$$

Number of Nodes

$$[n_x; 200; 100; 50; 20; 10; 5; \textbf{fork}; 5; 5; 92]$$

Figure 5: Softmax score predictor with 9 fully connected layers and a fork after layer 6. The output layer is a softmax classifier where the classes are the possible final scores of each team based on the max and min scores of the dataset.

Exponential Score Predictor



Exponential Output Activation

$$a^{[L]} = e^{Z}$$

MSE Loss

$$J = \frac{1}{m}\sum_{i=1}^{m} ||\hat{y}^{(i)} - y^{(i)}||_2^2$$

Number of Nodes

$$[n_x; 6; 5; 5; \textbf{fork}; 5; 5; 1]$$

Figure 6: Exponential score predictor with 6 fully connected layers and a fork after layer 3. This model utilizes the summed mean squared error of each of the forked outputs as the loss.

- In this work, three distinct Neural Network architectures are explored.
- Each model uses ReLU activations in the hidden layers.
- The weight kernels are initialized via Xavier Initialization and the biases are initialized at 0.
- The models are optimized via the Adam algorithm with a minibatch size of 128, a learning rate of 0.001, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
- Regularization methods experimented with include L2 and L1.
- Due to the small size of the dataset, HPC resources were not required and the models were trained on a personal laptop with Keras.

RESULTS

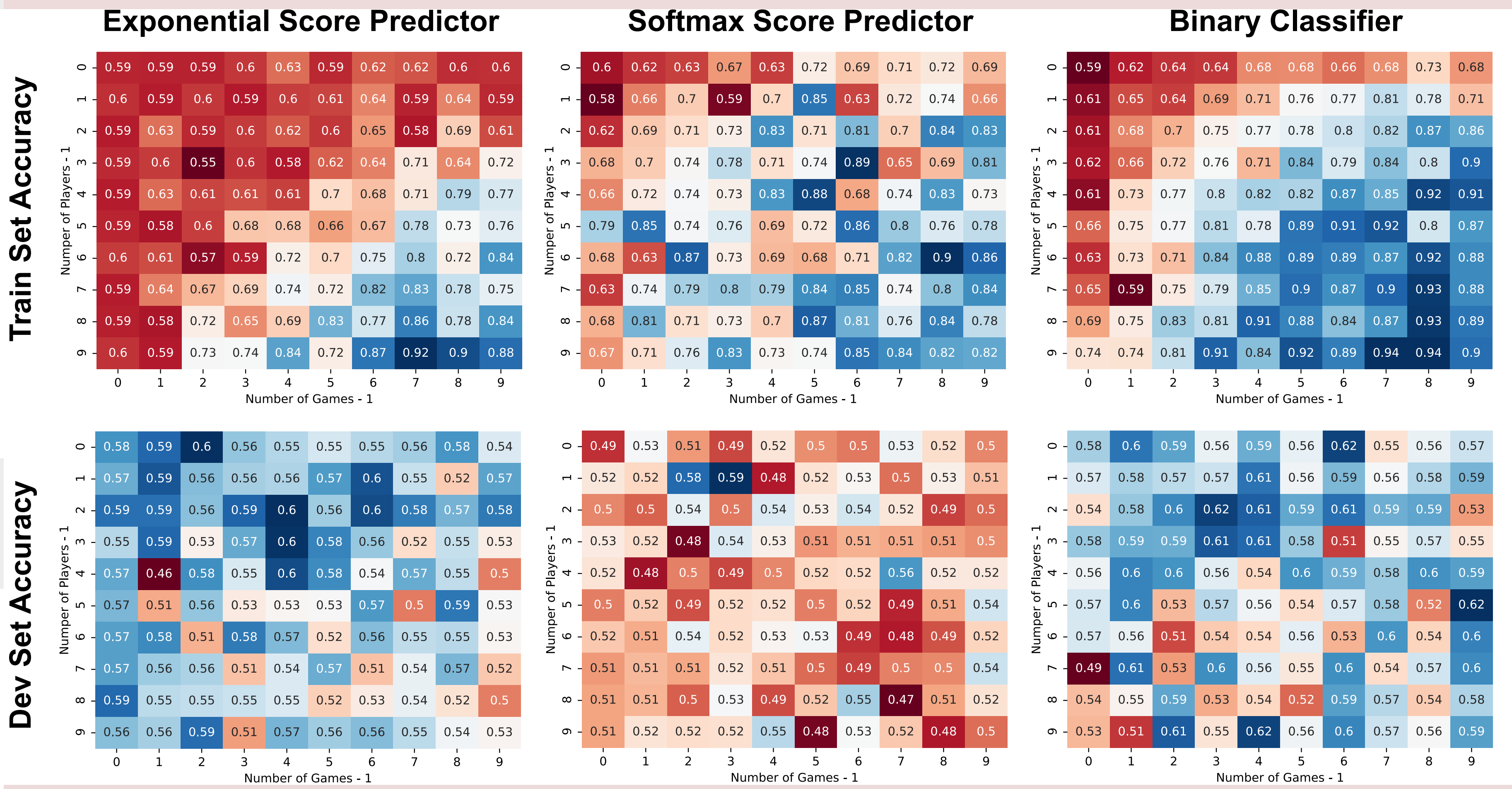


Figure 7: Dev set performance of the 3 NN models for each n_g, n_p pair.

Model	n_g,n_p pair	Train Set Accuracy	Dev Set Accuracy	Test Set Accuracy
Exponential	(4,3)	62%	60%	59.1%
Softmax	(4,2)	59%	59%	57.6%
Binary	(4,3)	75%	62%	59.8%

Figure 8: Accuracies on the train/dev/test sets for each model based on best n_g, n_p pair determined above.

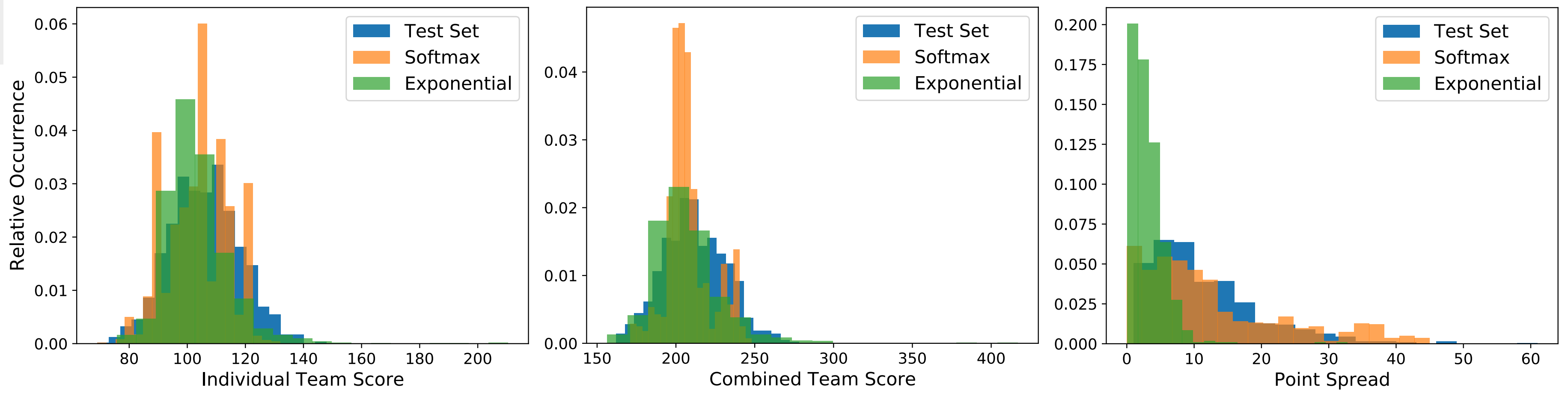


Figure 9: Histograms for individual team scores (left), combined scores (middle) and point spreads (right) from the true test set values and the predictions from each model.

CONCLUSIONS/FUTURE WORK

- The best features contained the stats of each team's top ~3 players in their last 4 games.
- The binary classifier nearly achieved a test performance on par with human experts. The exponential score predictor reproduced the score distribution of the test set quite well, indicating that it may be effective in predicting the over/under for NBA games. The softmax score predictor performed relatively poorly on the previous two tasks but managed to reproduce the margin of victory distribution remarkably well.
- Omission of some stats along with the inclusion of other stats that weren't present in the dataset would likely improve performance by shrinking the size of the feature vectors and making the information present more potent.
- Ultimately, the main issue is the lack of training data and the mismatch between the train and dev/test sets. To tackle both of these problems, it may be effective to artificially generate data via NBA basketball video games.
- The work presented here is a good first step toward considering a new dataset paradigm in NBA basketball prediction, and it is primed to be built upon.