Kobe Bryant Shot Selection

Which shots did Kobe sink?

Presented by
Divya Parmar
Pranay Bhakthula

Introduction

- Kobe Bryant is widely regarded as one of the greatest National Basketball Association players of all-time.

- He is renowned for his work ethic, competitiveness, and ability to deliver in a game’s most pressure-packed moments. Bryant won five NBA championships, was an 18-time NBA All-Star, and led the NBA in scoring twice.

- The world was stunned when he tragically passed away in 2020.

Project Overview

The focus of this project is to classify whether former National Basketball Association player Kobe Bryant made or missed a given shot attempt.

We used the following algorithms:
1. Random Forest classifier
2. Multi-layer Perceptron (MLP) classifier

In addition, we did exploratory data analysis of Kobe Bryant's shooting performance in the 2002 and 2009 seasons, in which he won NBA Championships.

Dataset Description

- This data contains the location and circumstances of every field goal attempted by Kobe Bryant took during his 20-year career (spanning the regular season and playoffs).
- This dataset contains 30,697 rows.
- Each row corresponds to one shot attempt.
- There are 25 features, most of which are categorical.
- Target variable : shot_made_flag

- Source: Kaggle

Data Pre-Processing

- Dropped 5000 observations since target variable shot_made_flag has 5000 inherent missing values. These rows were intentionally held out for a Kaggle competition, but we have no use for unlabelled data. Of the labelled observations, 11,465 were made and 14,232 were missed (44.6% conversion rate).

- Dropped columns such as game_event_id, game_id, lat, loc_x, loc_y, lon, minutes_remaining, seconds_remaining, minutes_remaining, team_name, team_id, game_date, game_date, matchup, opponent, and shot_id since they have no influence on target variable or have high cardinality.

- Changed shot distance maximum to 40 to reduce variable range caused by outliers.

- Label encoder is applied to convert categorical features to numeric.

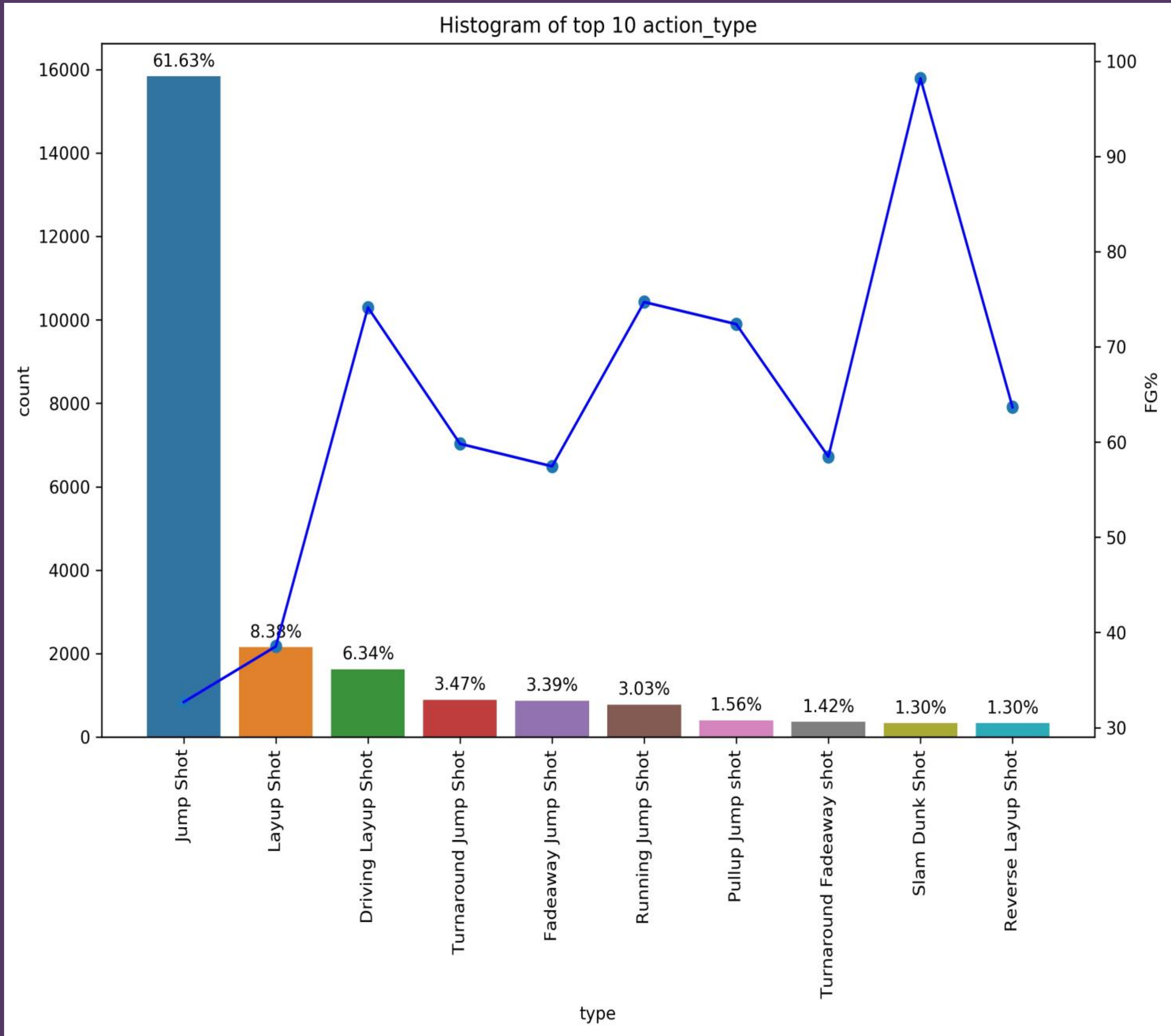
- MinMaxscaler is used to scale the values between 0 and 1. Data normalization helps model performance by avoiding large weight values.

EDA Analysis

- The graphs we are going to see are histograms on one axis arranged in descending order and Field goal% on 2nd axis.
- The percentage on top of each bar represents percentage of usage of that particular type in that column
- The line plot represents,

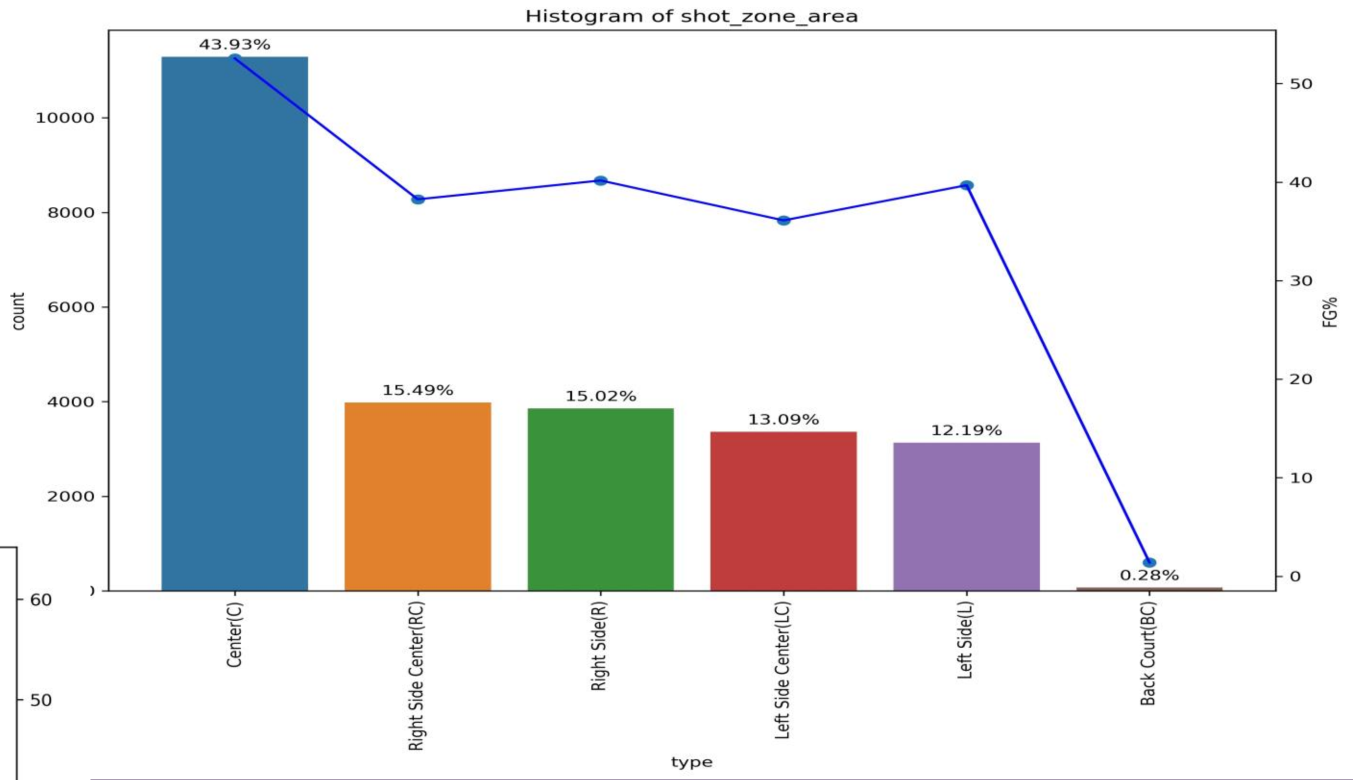
Field Goal(FG) % = (no. of successful shots in particular type) / (total no. of attempted shots in that particular type)

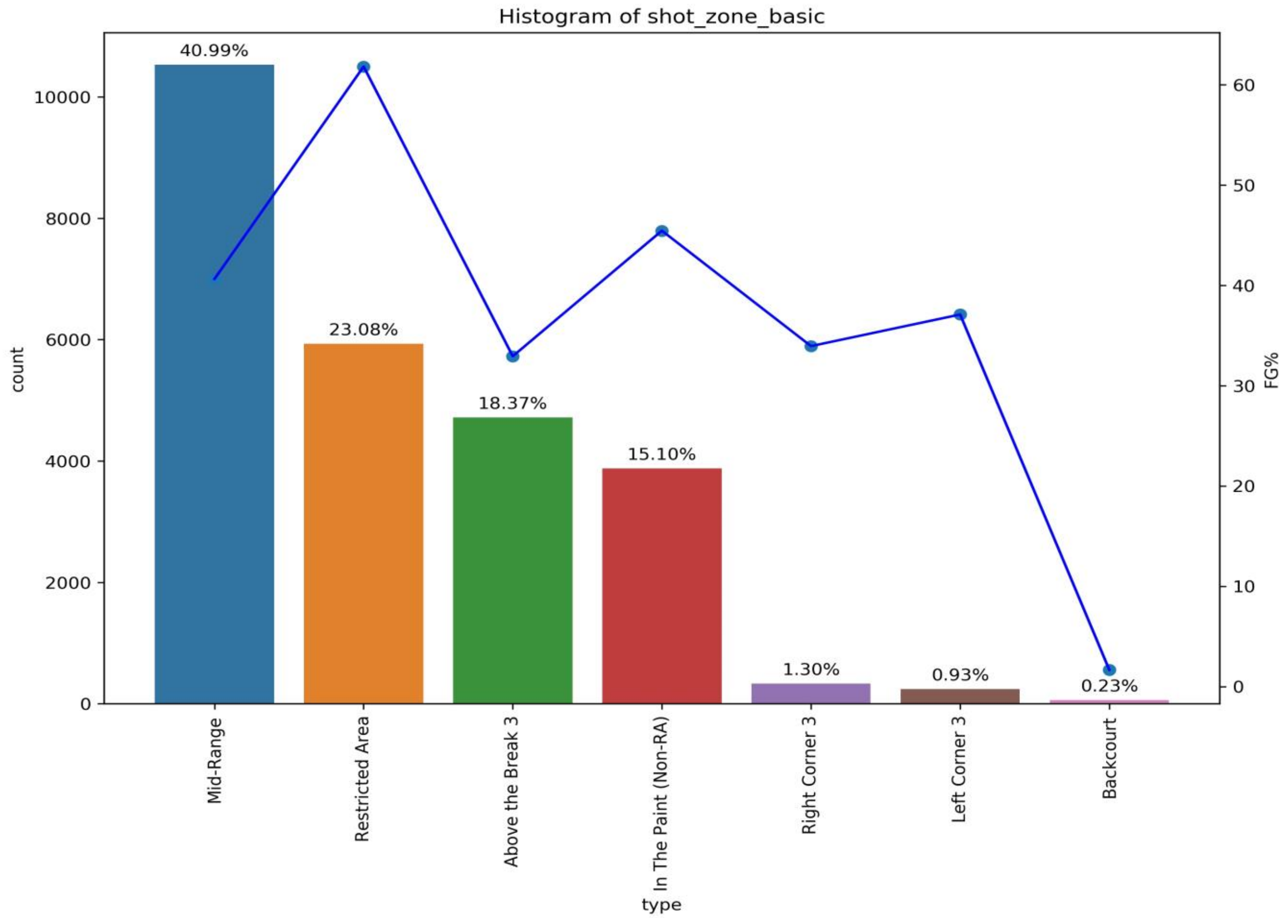
- This histogram on the right shows top 10 action types that Kobe used most during his career.
- We can observe that he uses jumpshots the most followed by layup shot.
- We can also observe that FG% is least for jumpshots and highest for slam dunk shots.



EDA Analysis

- The plot on the right is histogram of shot_zone_area. We can observe that Kobe shoots most of his shots (43.93%) from center area followed by right side center.
- The FG% is also highest for center area, so it is also his most effective area of shooting.

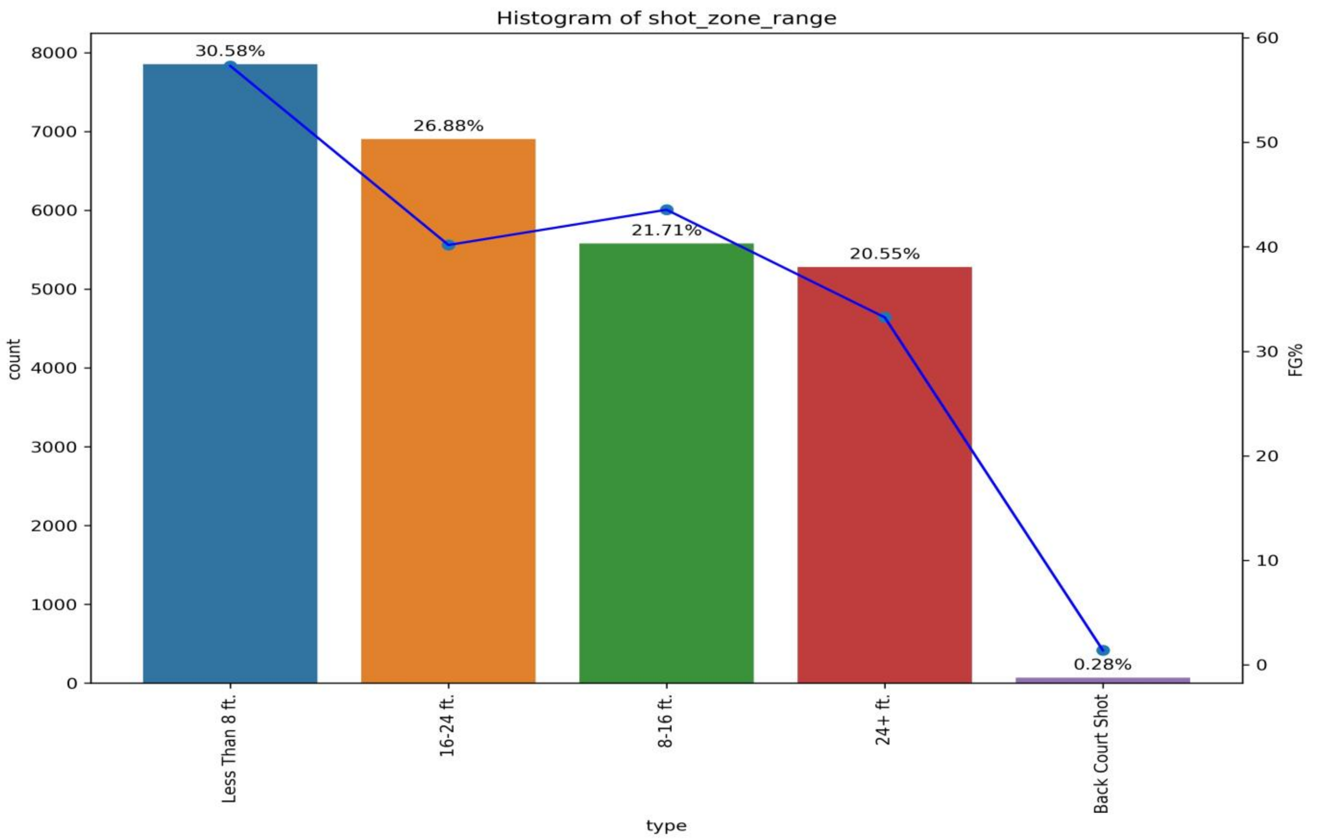


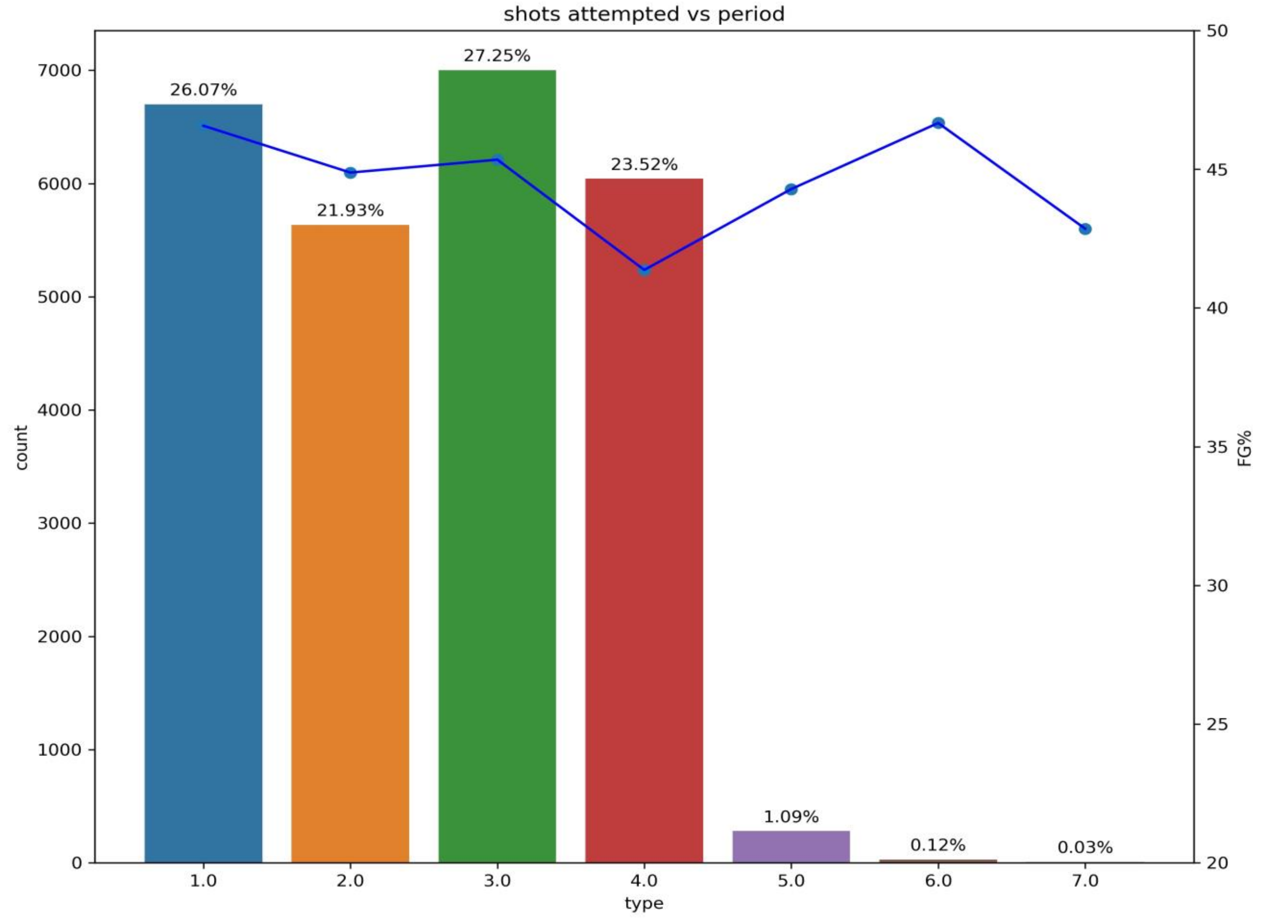


- The plot on the left is histogram of shot_zone _basic. We can observe that Kobe shoots most of his shots from Mid-Range(40.99%) followed by Restricted Area.
- We can also observe that FG% is highest for Restricted Area so that is his most effective zone of shooting. We can see that his least favourite area to shoot is Backcourt.

EDA Analysis

- The plot on the right is histogram of shot_zone_range. We can observe that Kobe shoots most of his shots (30.58 %) from Less that 8ft range followed by 16-24ft range.
- The FG% is also highest for Less that 8ft range, so it is also his most effective range of shooting.
- The effectiveness decreases the farther he is from the basket.

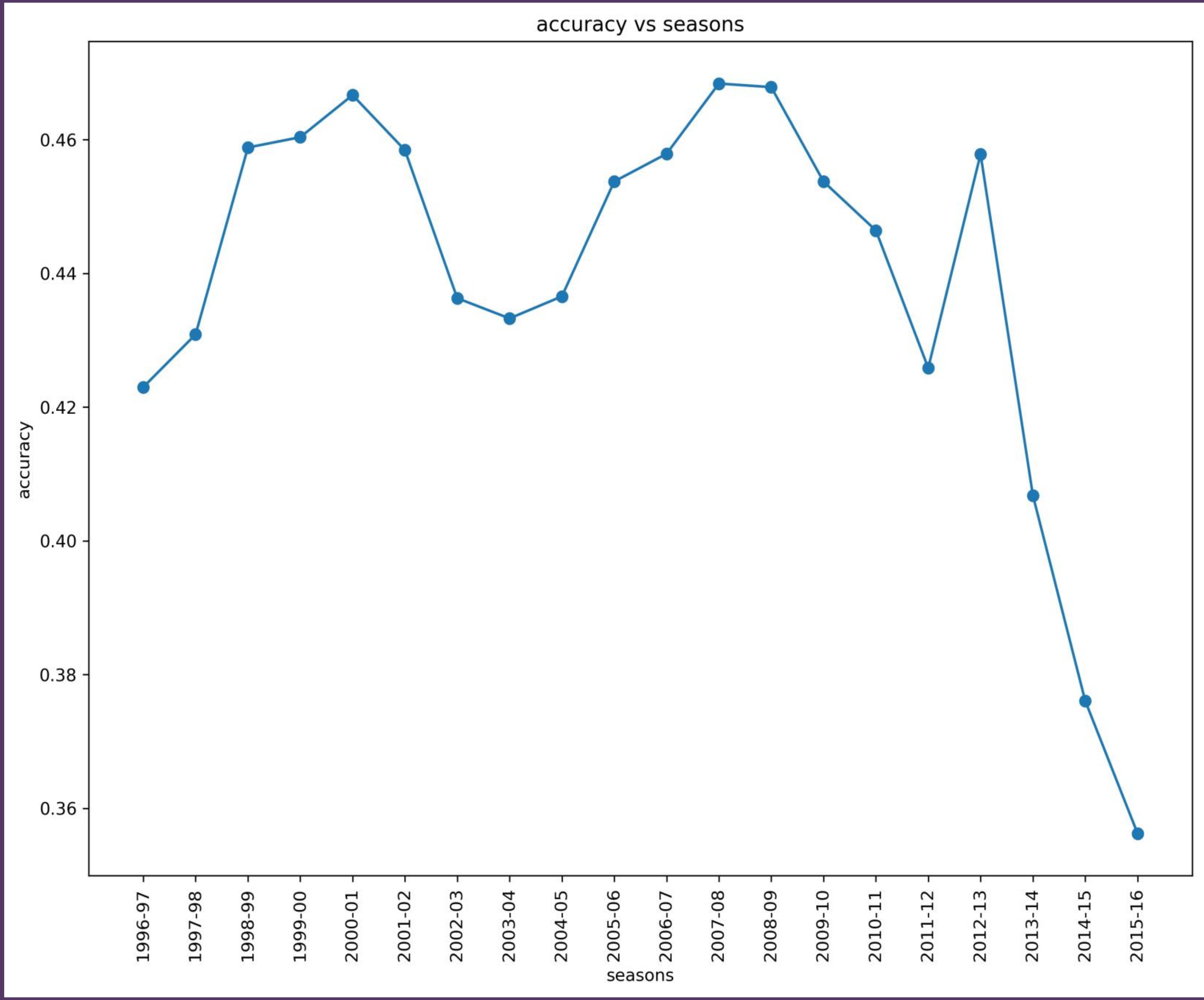




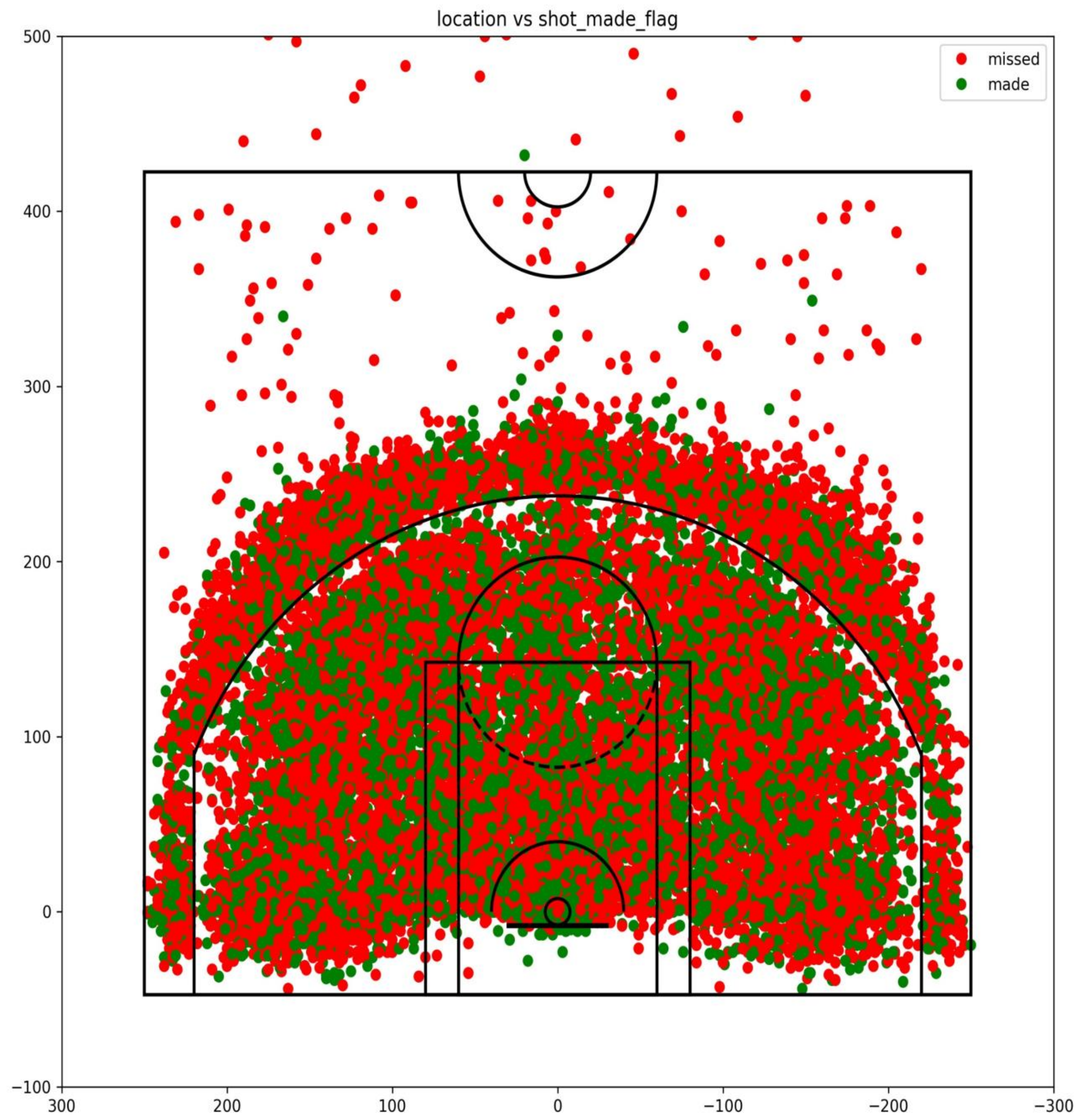
- The plot on the left is histogram of period. We can observe that Kobe is generally more active in 1^{st} and 3^{rd} quarters and most inactive for 2^{nd} quarter. This is technique he follows so he conserve his energy in 2^{nd} quarter and make most out of 3^{rd} quarter and perform clutch moments in 4^{th} quarter.
- We can also observe that FG% is highest for 1^{st} quarter so that is his most effective period.

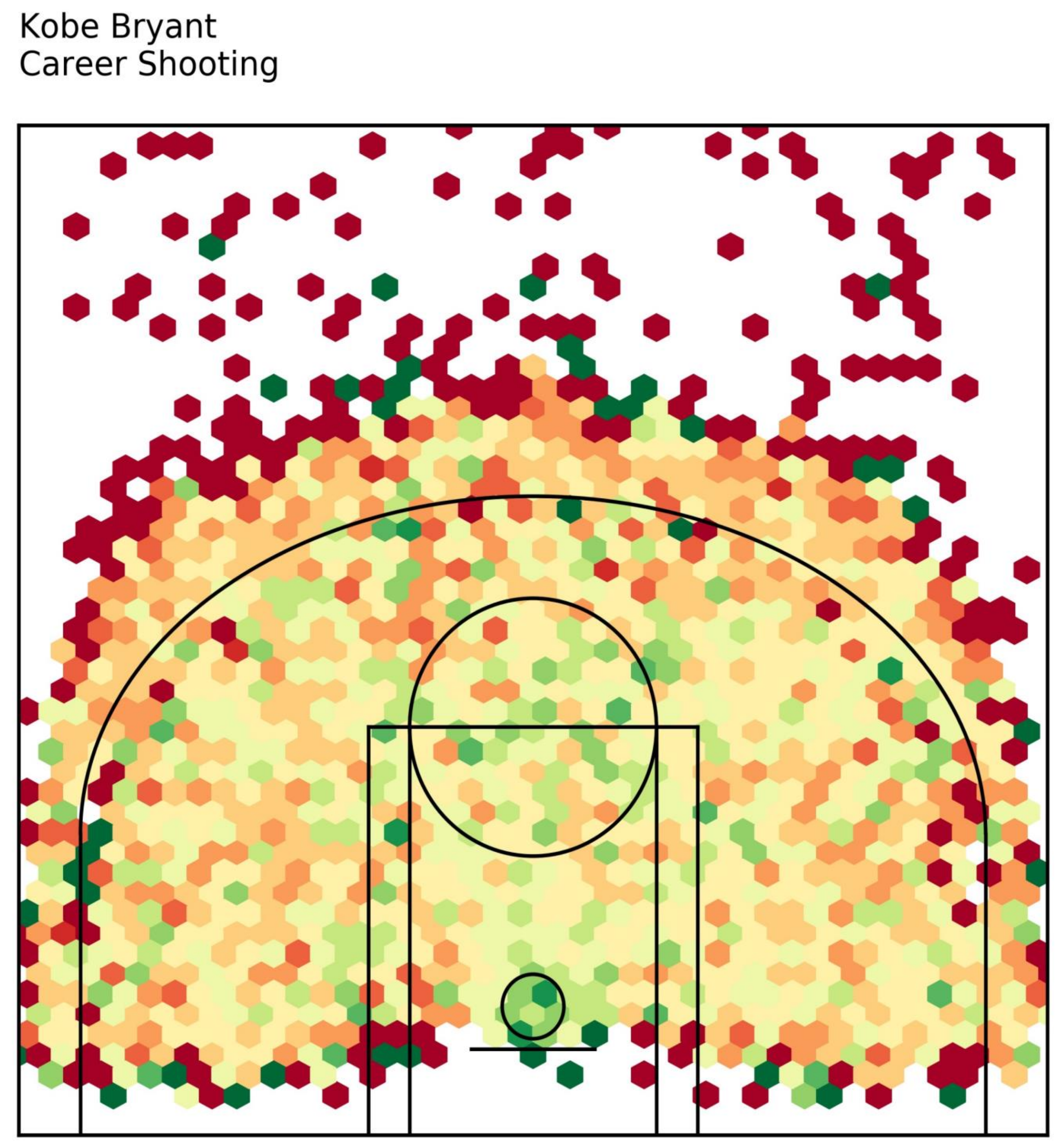
EDA Analysis

- The plot on the right shows accuracy of shots attempted by Kobe for various seasons.
- We can observe that Kobe performed with highest accuracy in the seasons 1999-00, 2000-01 and 2001-02 where he won championships those years to perform a three-peat.
- He again peaked in seasons 2007-08, 2008-09 and 2009-10 where he won championships in years 2009 and 2010.
- We can observe that his accuracy faded in his final years, reaching his lowest in 2015-16 season.
- Kobe’s best shooting seasons coincided with his team winning the NBA Championship.

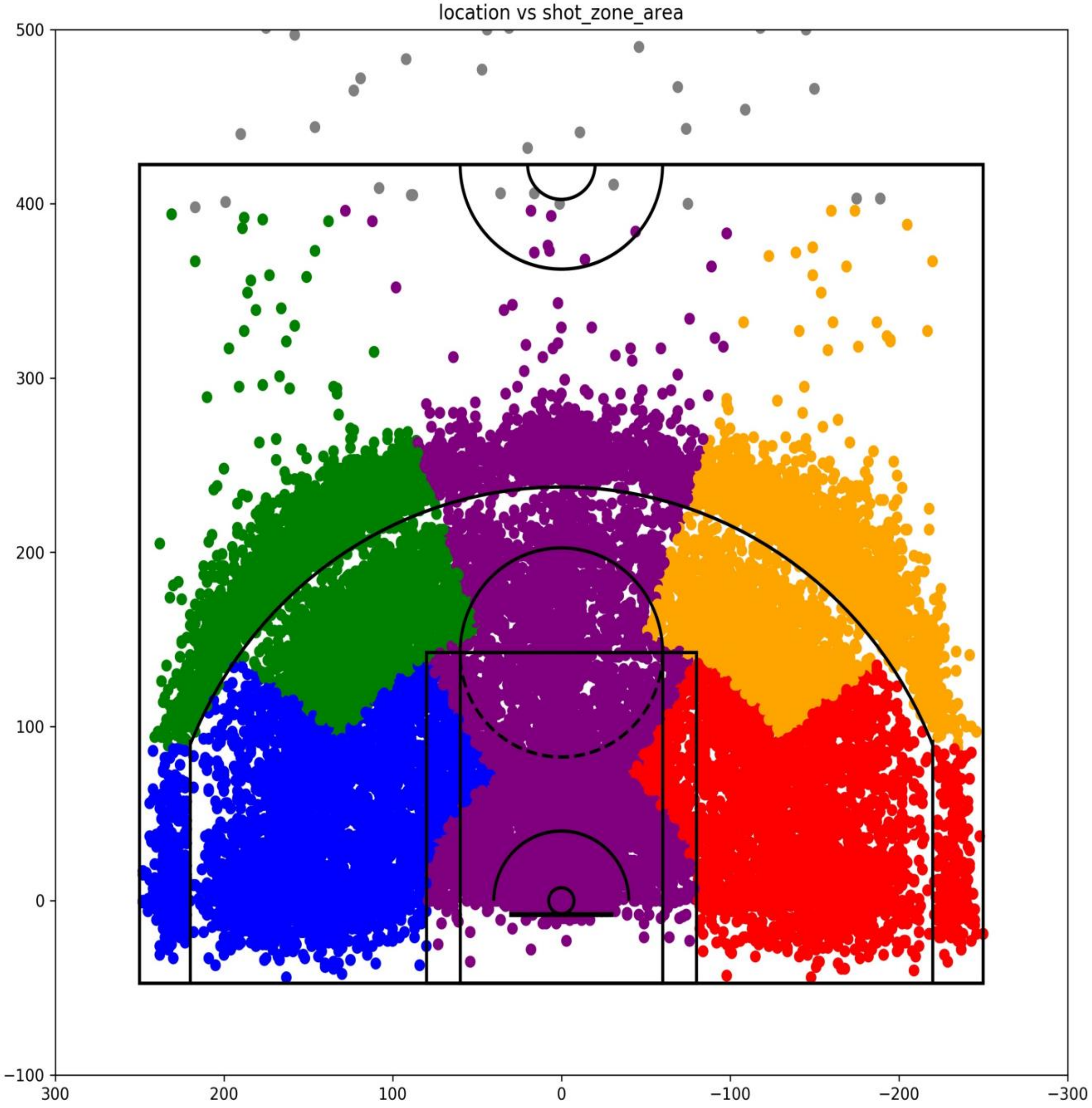


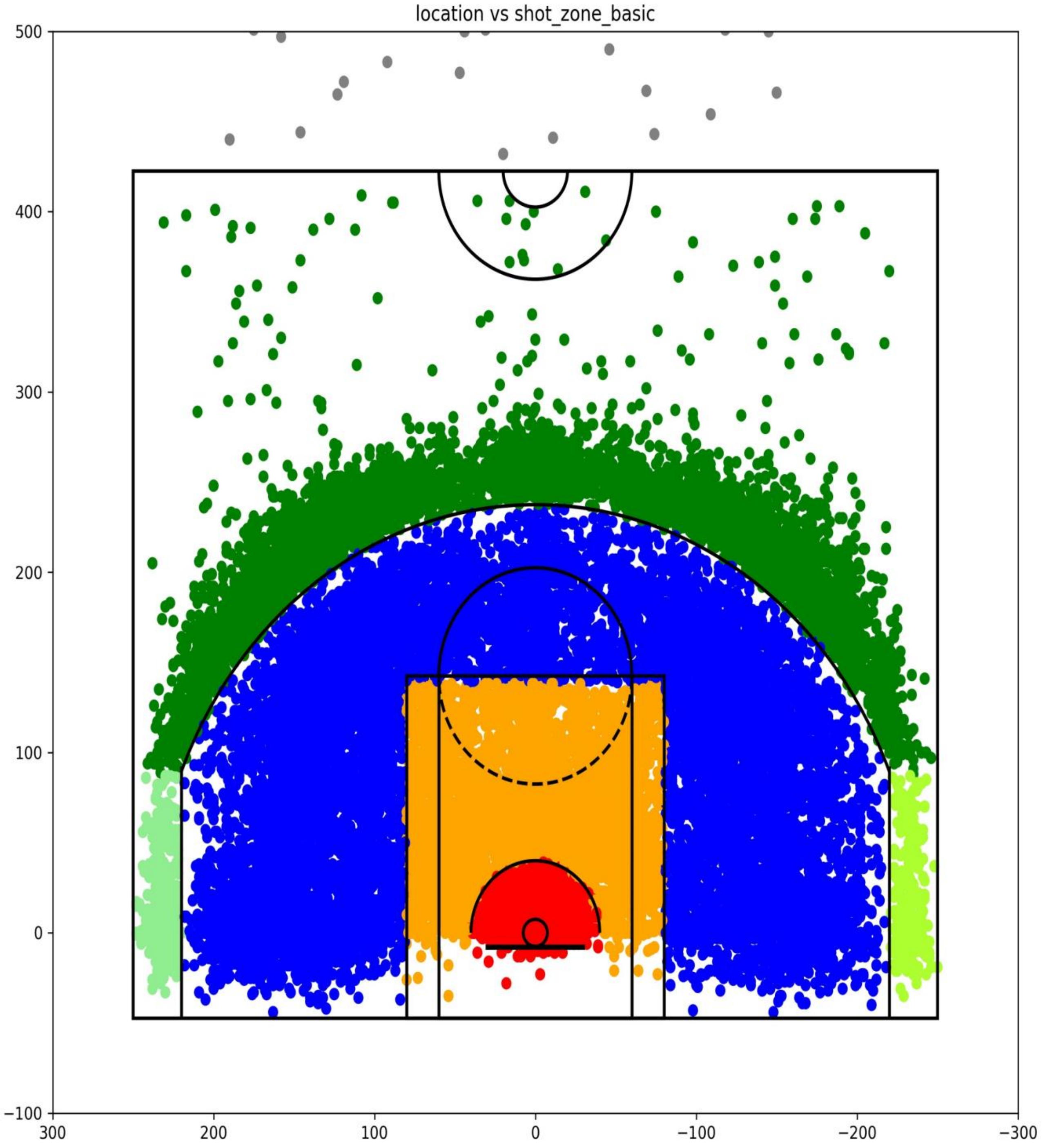
EDA Analysis





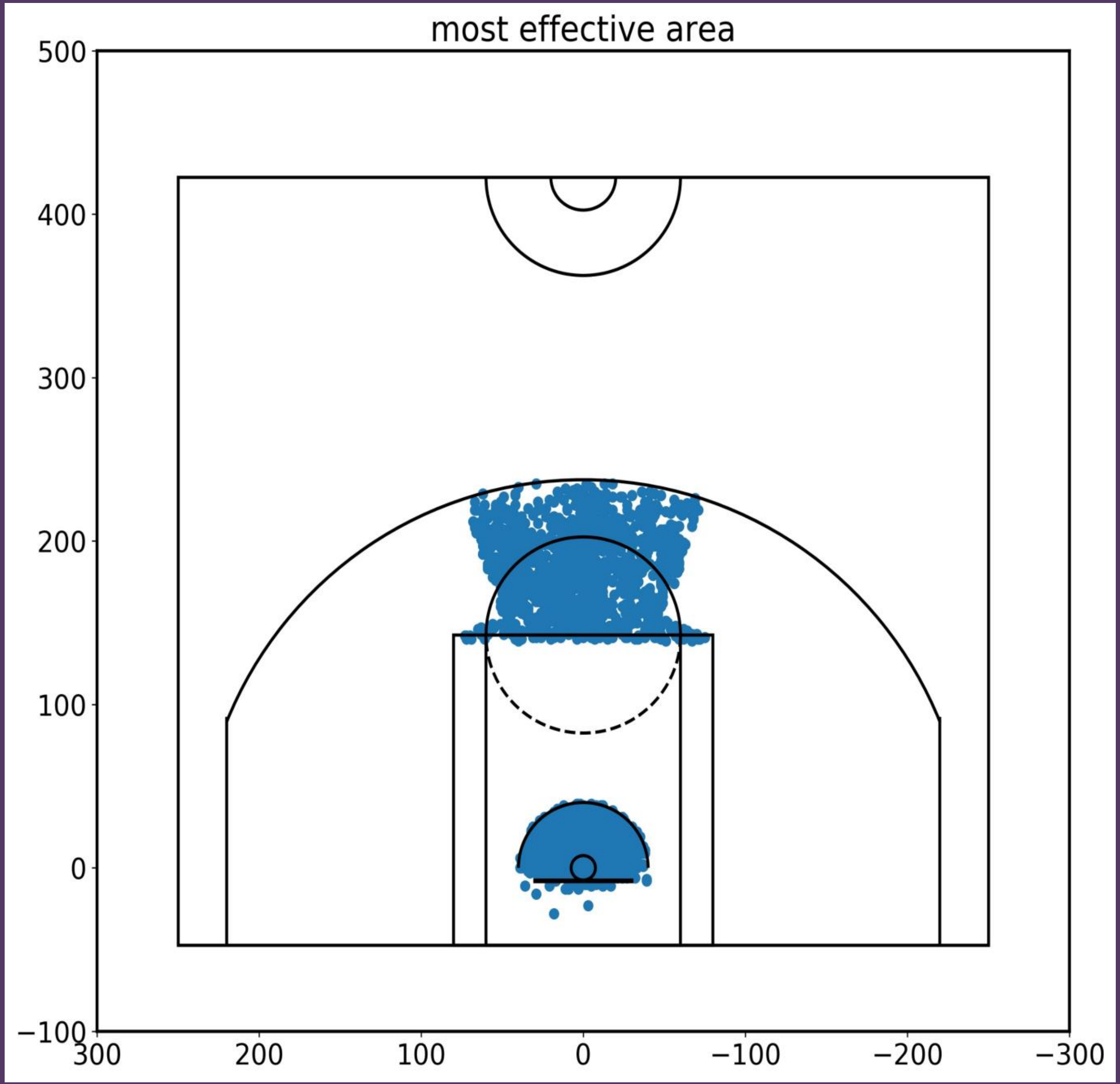
EDA Analysis





EDA Analysis

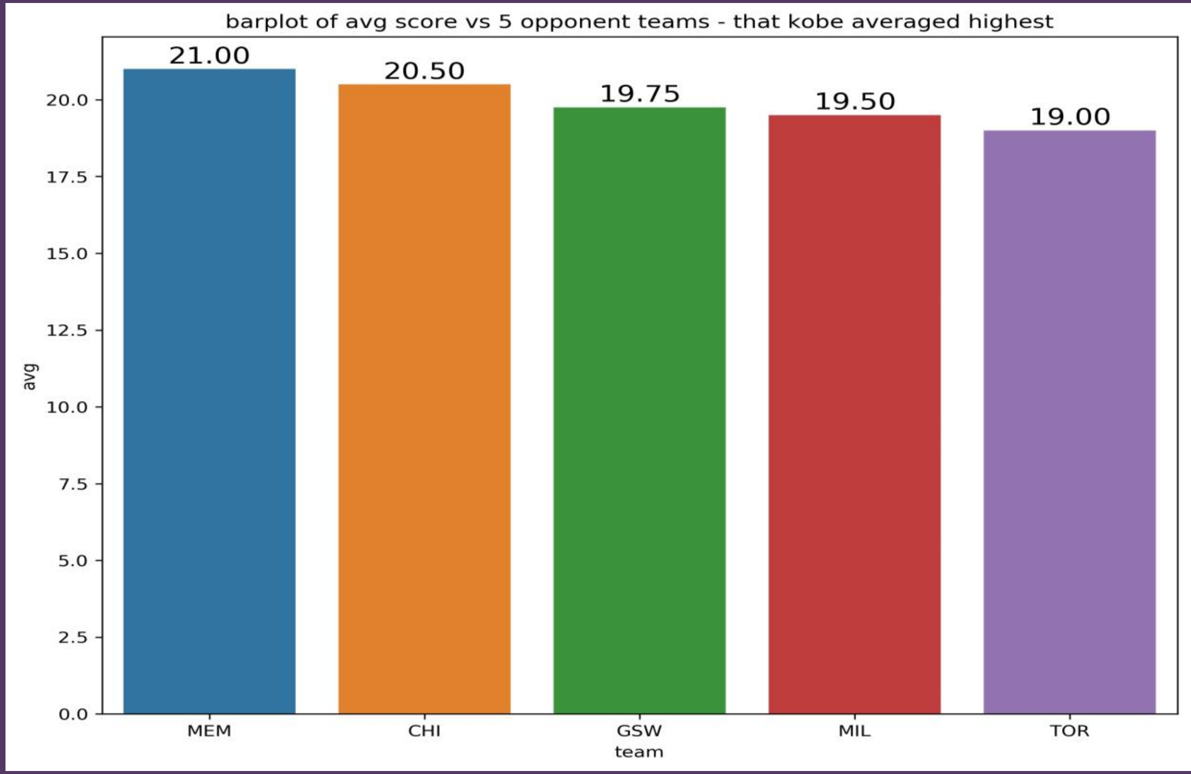
- The chart on the right shows the most effective areas of Kobe's play.
- From the previous charts of location of shots and histogram graphs of various areas, ranges we can come to the conclusion that the most effective area for Kobe is shooting from center area, from Mid-range and Restricted area. The most used action type is Jump shot.
- For any opponent team playing against Kobe, the best tactic is to make him stay away from this region as much as possible and also since he uses Jump shot the most it is better to assign him a taller opponent to counter his jump shot.

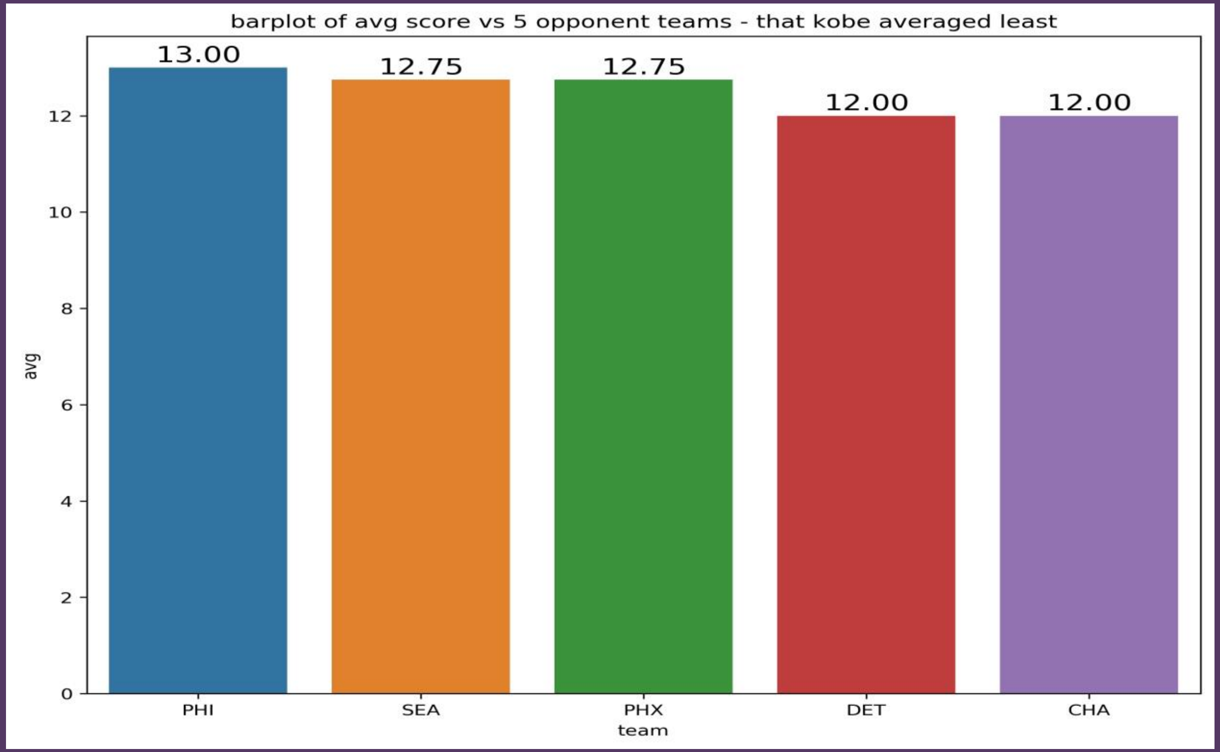


Opponent Analysis

- Now we will look from the opponent’s point of view, who are a championship finalist in the years **2002 and 2009** against Kobe’s team(LA Lakers), and analyze Kobe’s strengths and weakness from the data of regular season in those years to come up with a plan to defend Kobe better.
- We chose 2002 and 2009 years because we wanted to analyze the Kobe’s play in the years when his teammate, another NBA star Shaquille O’Neal, was on his team (2002) and not on his team (2009).
- Shaq was more dominant than Kobe in year 2002, as Kobe was still young and less experienced compared to Shaq. In the year 2009, Kobe was peaking in his career and he was the most dominant player in his side. So we can compare his play both those years.
- In this analysis, we will look at data against the team that Kobe played the best and team that Kobe performed the least.

Opponent Analysis - 2002

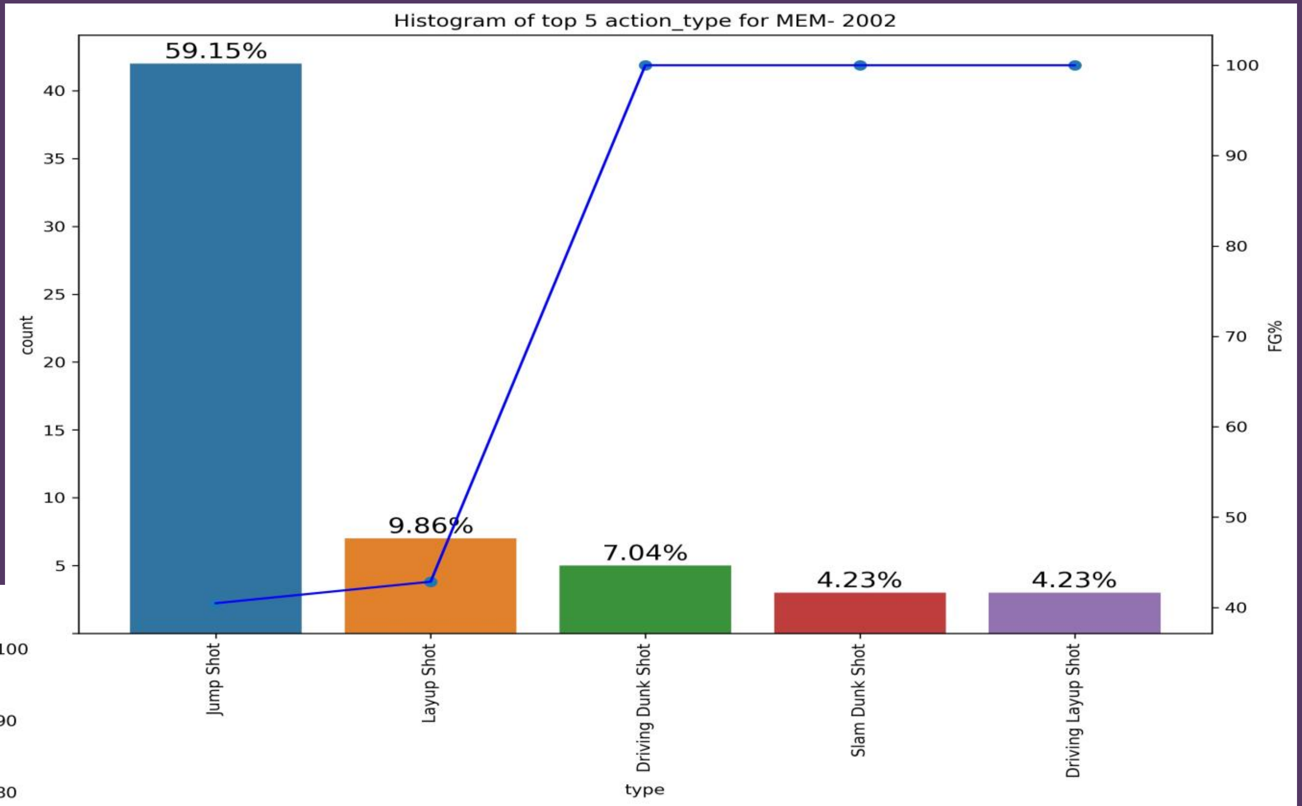


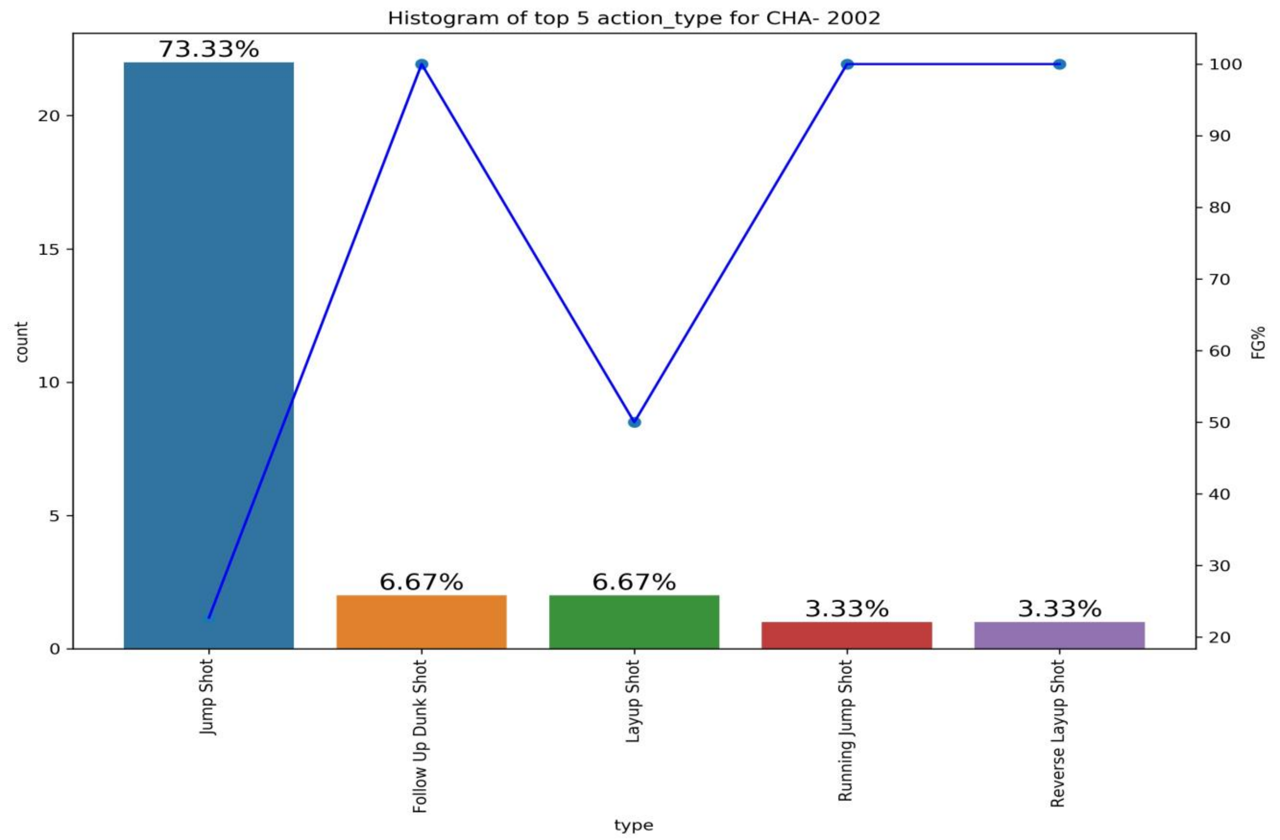


- Above are the bar graphs showing the 5 teams against which Kobe averaged the most and least points against.
- In the year 2002, Kobe scored the most against MEM(Memphis Grizzlies) and the least against CHA(Charlotte Hornets).

Opponent Analysis - 2002

- From the graph on the right, we can observe that against MEM his most used shot is jump shot (59.15%) with FG% of around 40% followed by layup shot.
- But though used less the dunks have highest FG%, so they are most effective shot type.

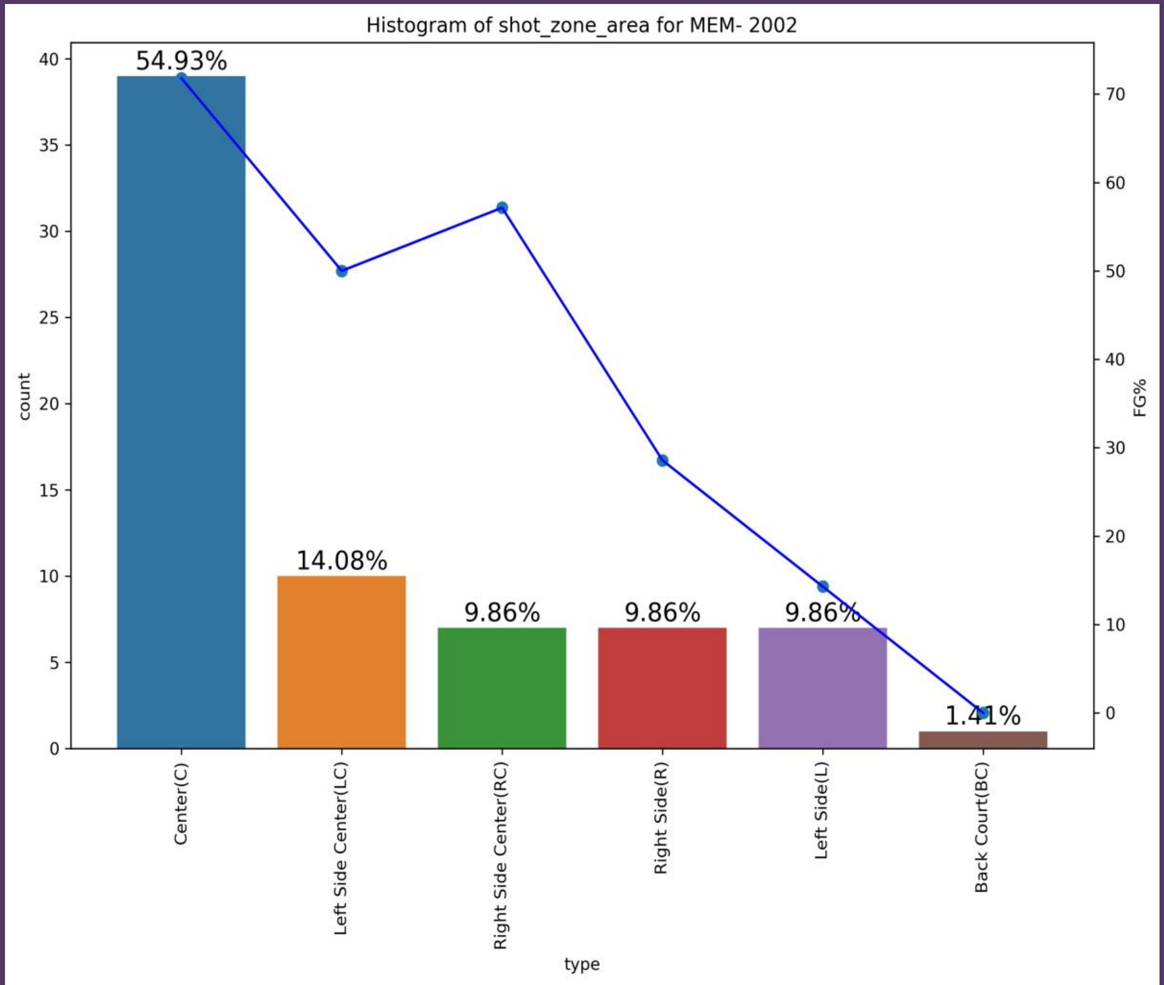


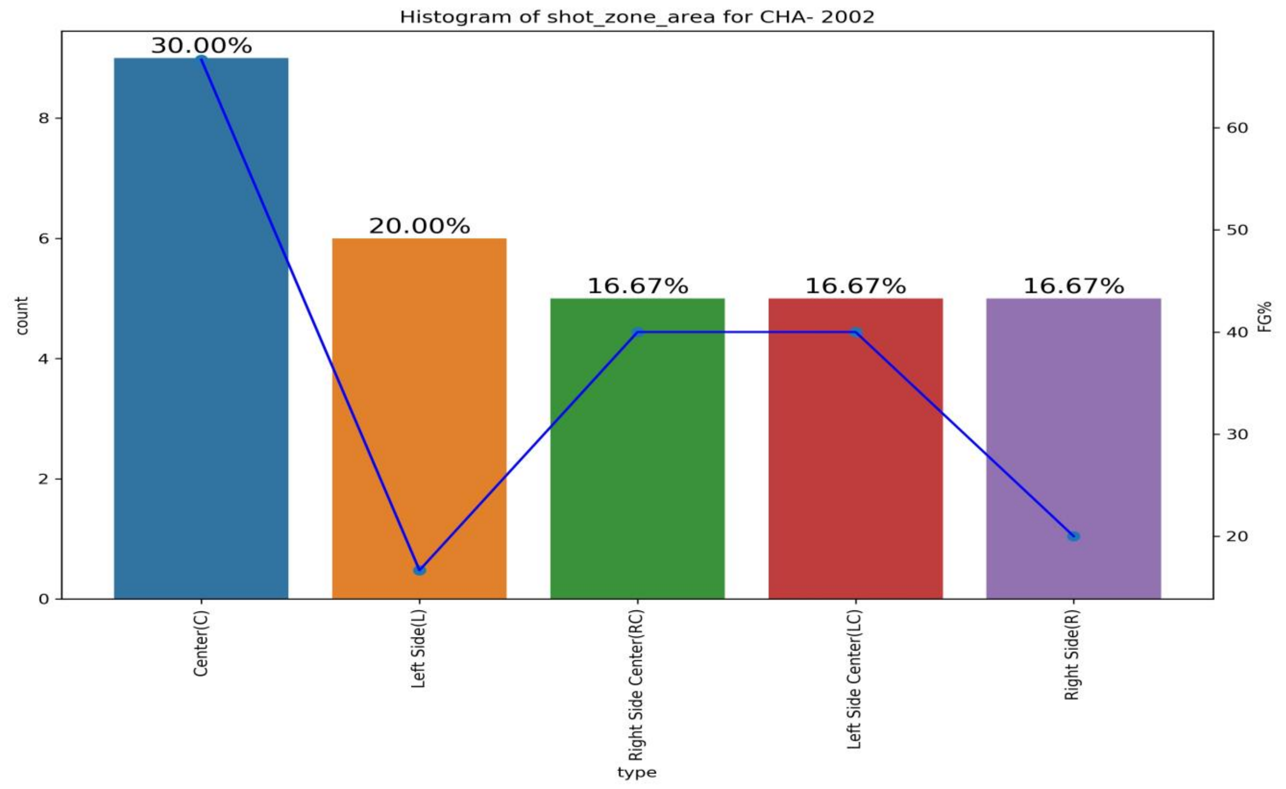


- From the graph on the left, we can observe that against CHA his most used shot is jump shot (73.33%) with FG% of around 20% followed by dunk shot.
- From this we can observe that this team have not allowed him to shoot the jump shot with high accuracy, possibly by assigning him a taller opponent.

Opponent Analysis - 2002

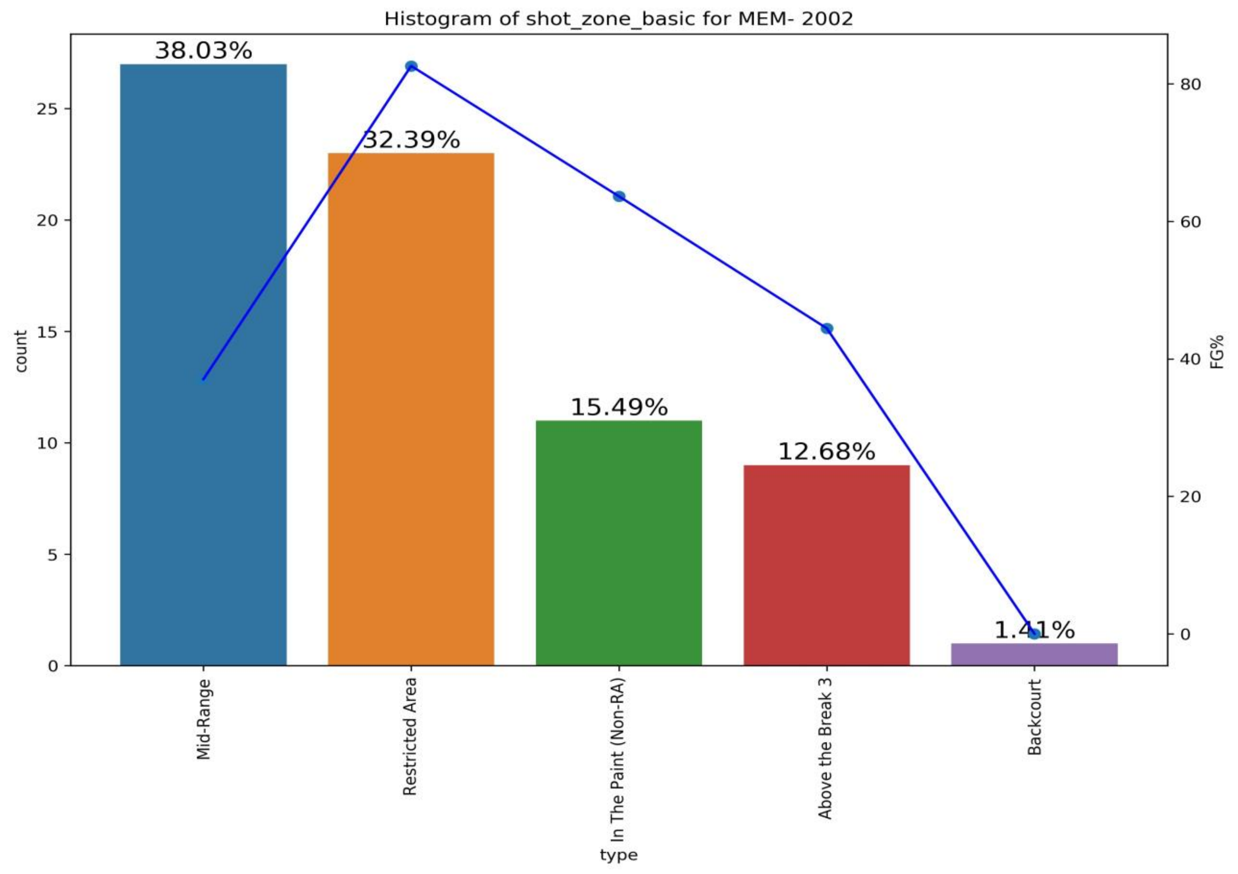
From both the graphs we can observe that Kobe shoots mostly from the Center area, also with highest FG% followed by his shots from left side so the takeaway points is to force him away from these areas to right side if possible.

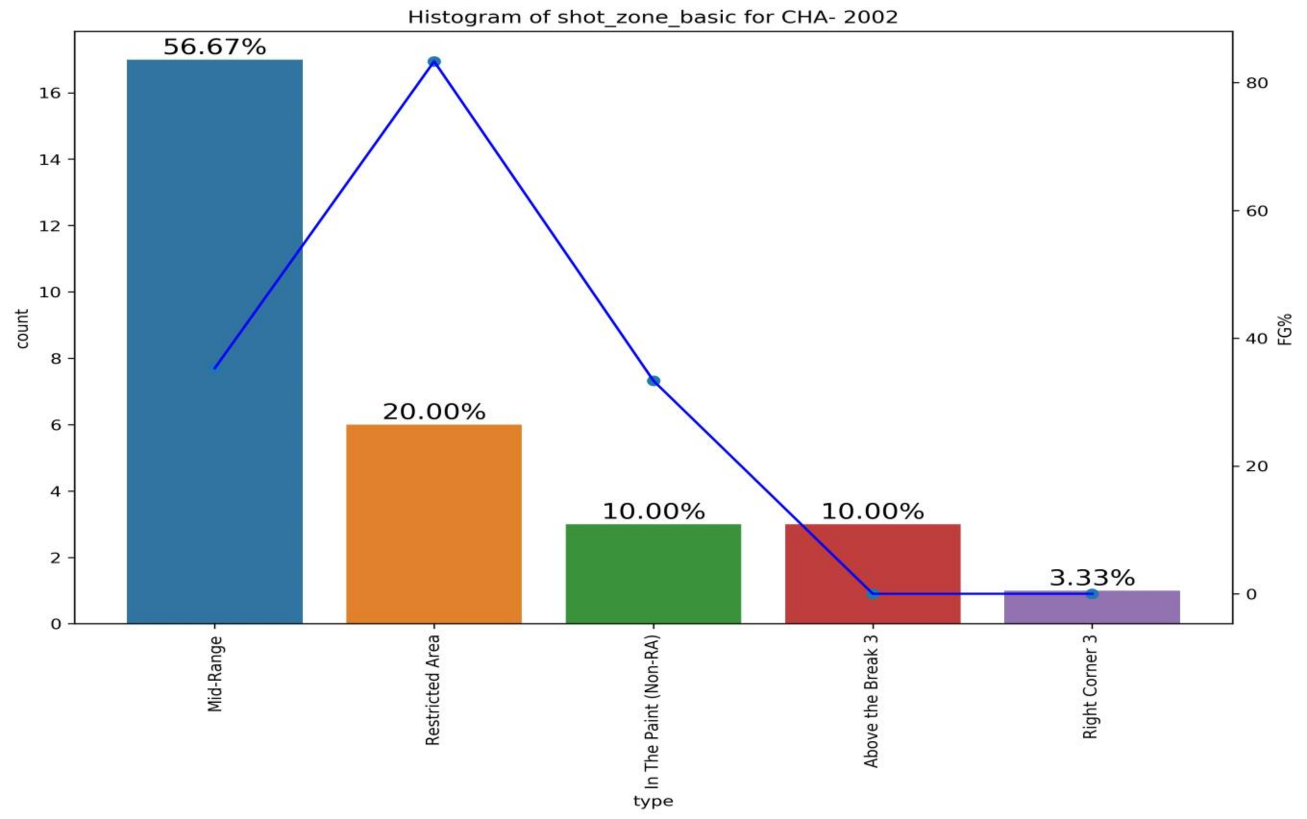




Opponent Analysis - 2002

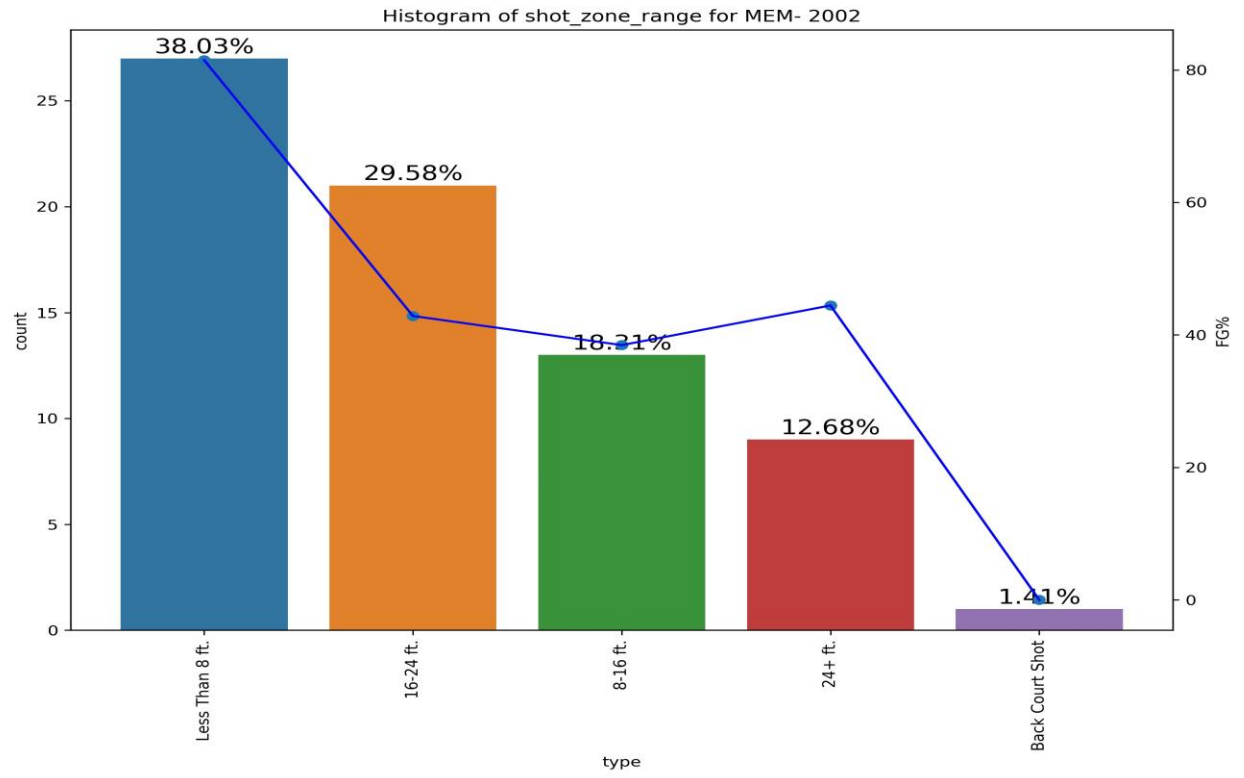
From both the graphs we can observe that Kobe shoots mostly from Mid-Range area, followed by from Restricted Area also with highest FG% so the takeaway points is to force him away to shoot from longer regions.

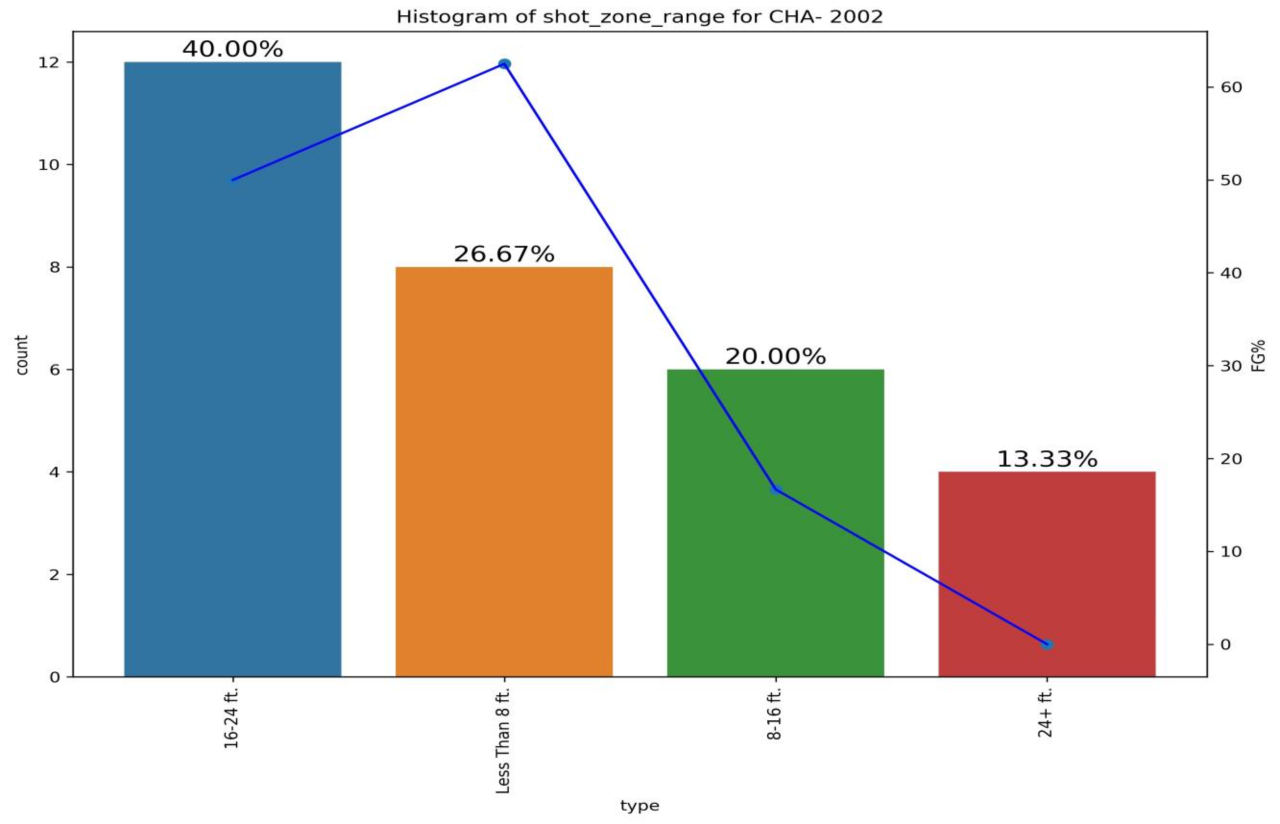




Opponent Analysis - 2002

From the graphs on right, we can observe that Kobe shot most shots from ‘Less than 8ft’ against MEM and from graph below we can see that CHA team restricted Kobe from shooting from ‘Less than 8ft’ and that allowed them to make Kobe less effective.

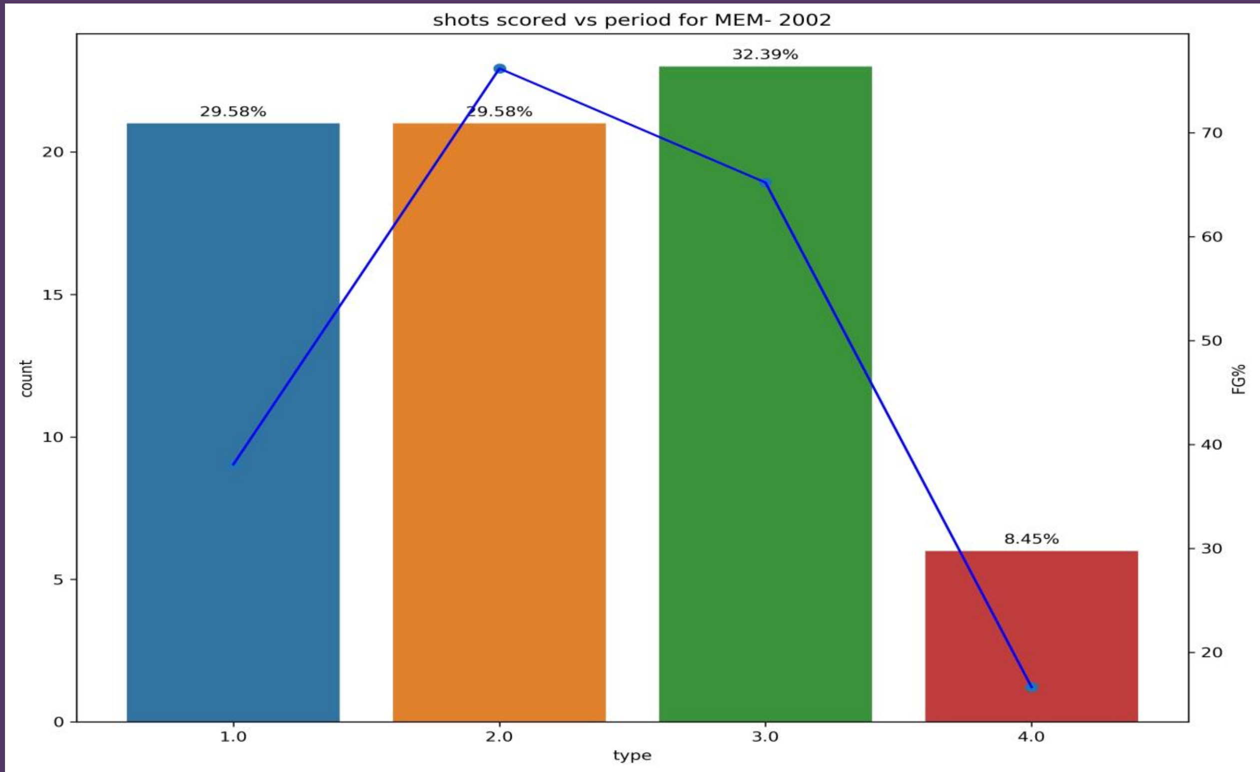


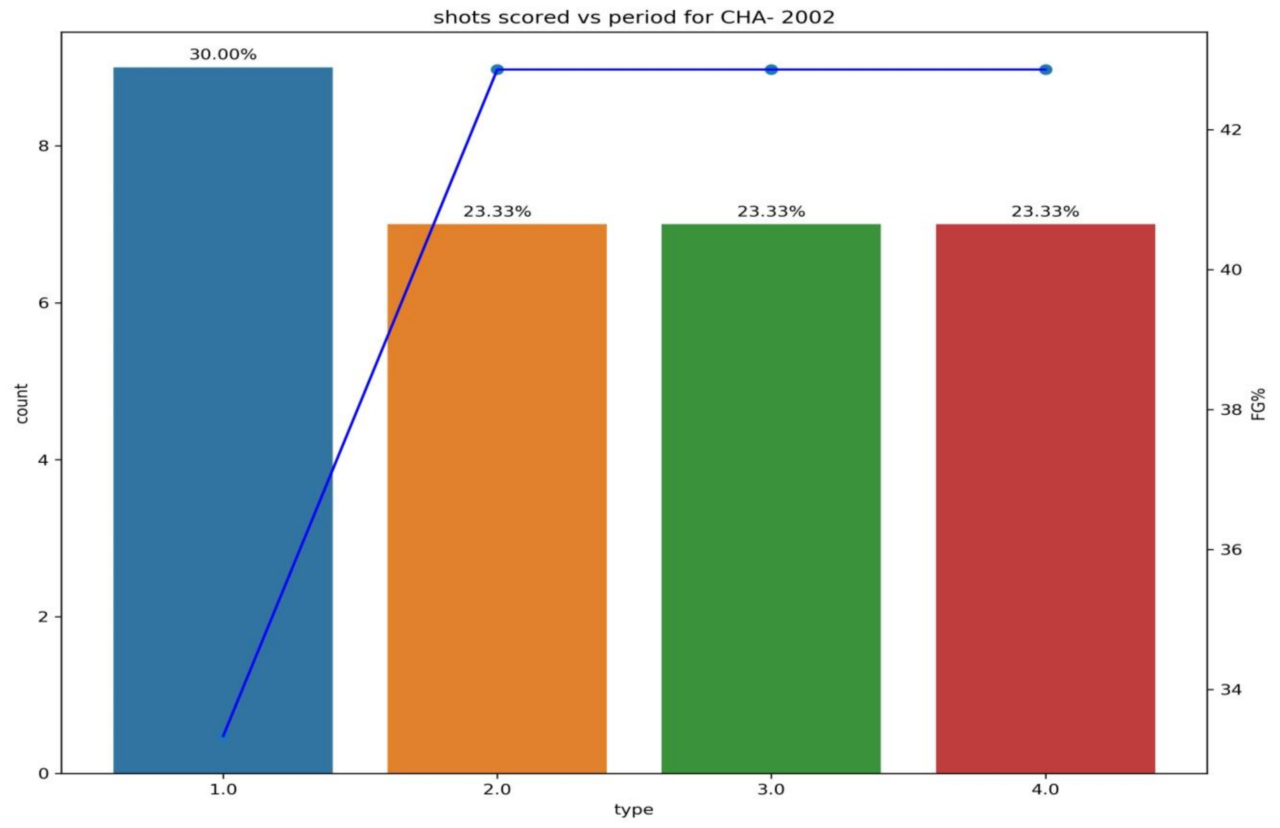


So the takeaway point is to guard him so that he does not reach ‘Less than 8ft’ area from where he is most effective.

Opponent Analysis - 2002

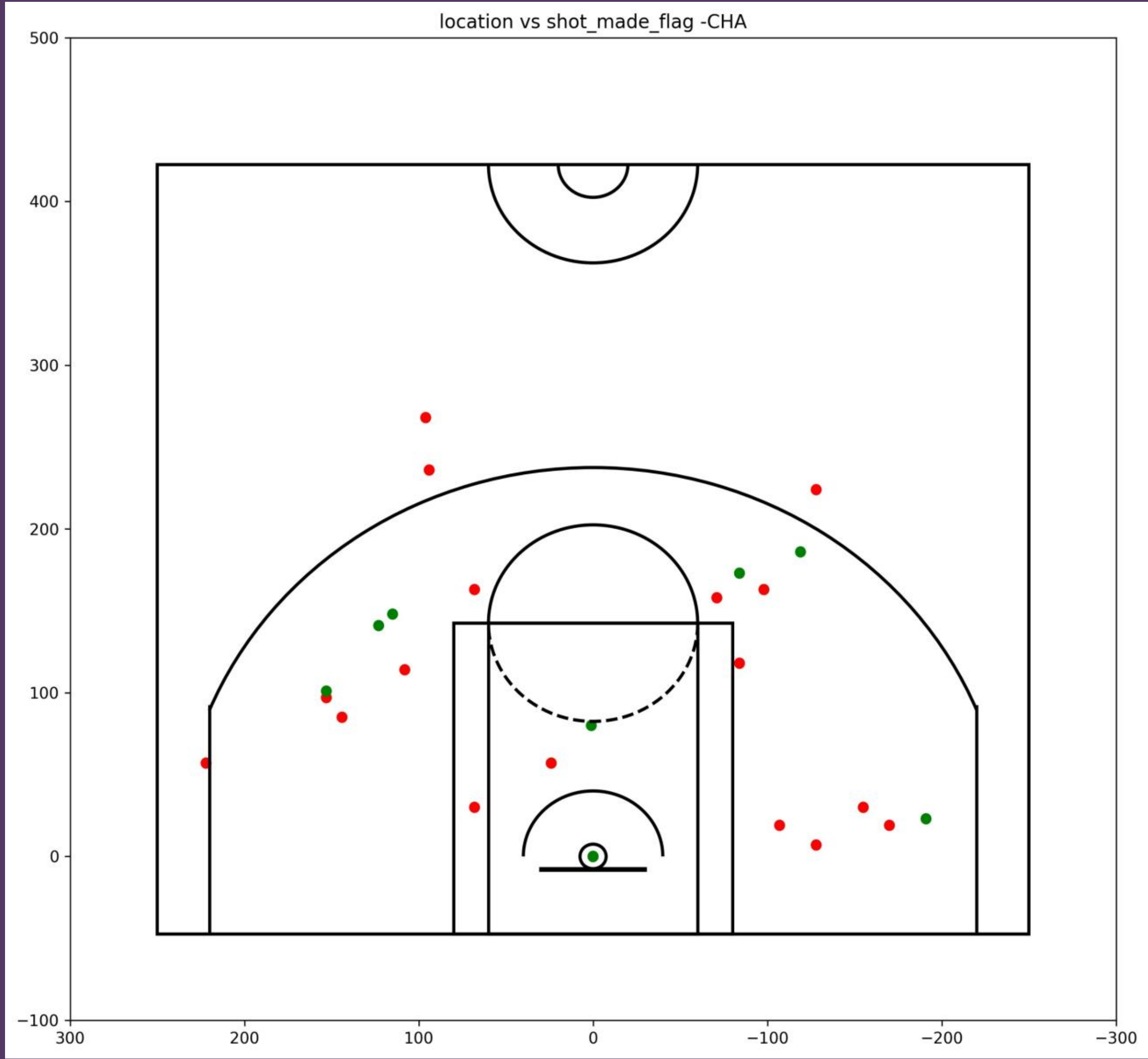
Against MEM we can observe that Kobe played evenly in 1st and 2nd quarter and was most active in 3rd quarter and had highest FG% in 2nd quarter

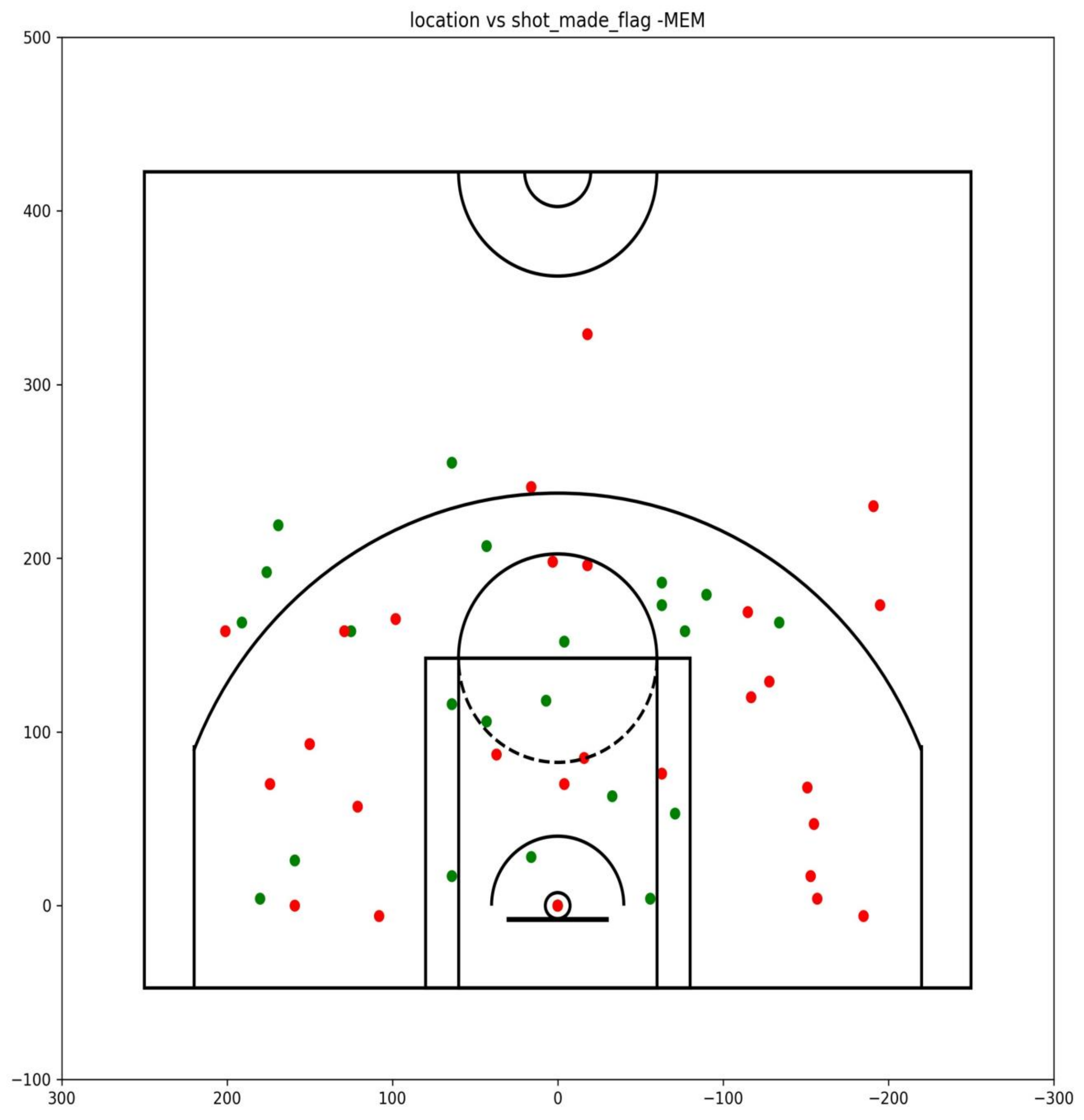




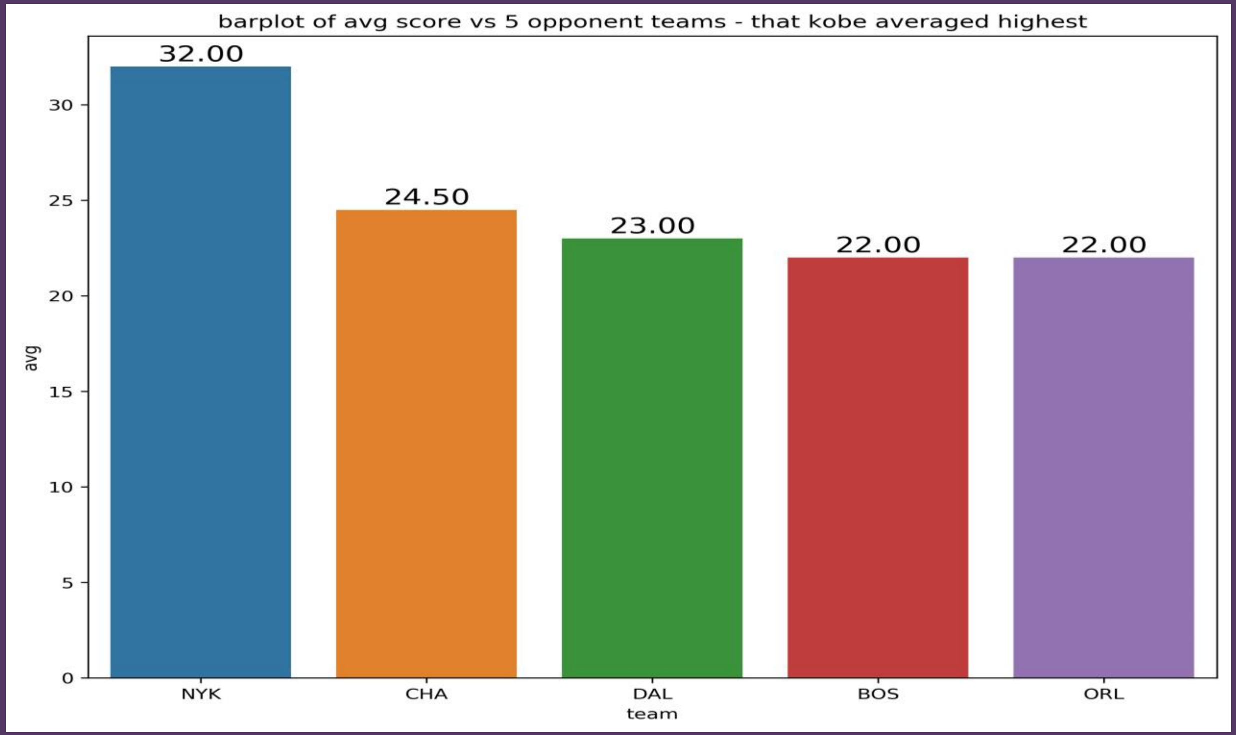
Against CHA we can observe that they have restricted him in 2nd and 3rd quarter with much less FG% in all quarters, so the takeaway point is to put more pressure in 1st quarter and make him score with less accuracy where he has more energy so he might perform less in following quarters where he has less energy and also additional pressure to perform more to compensate form 1st quarter.

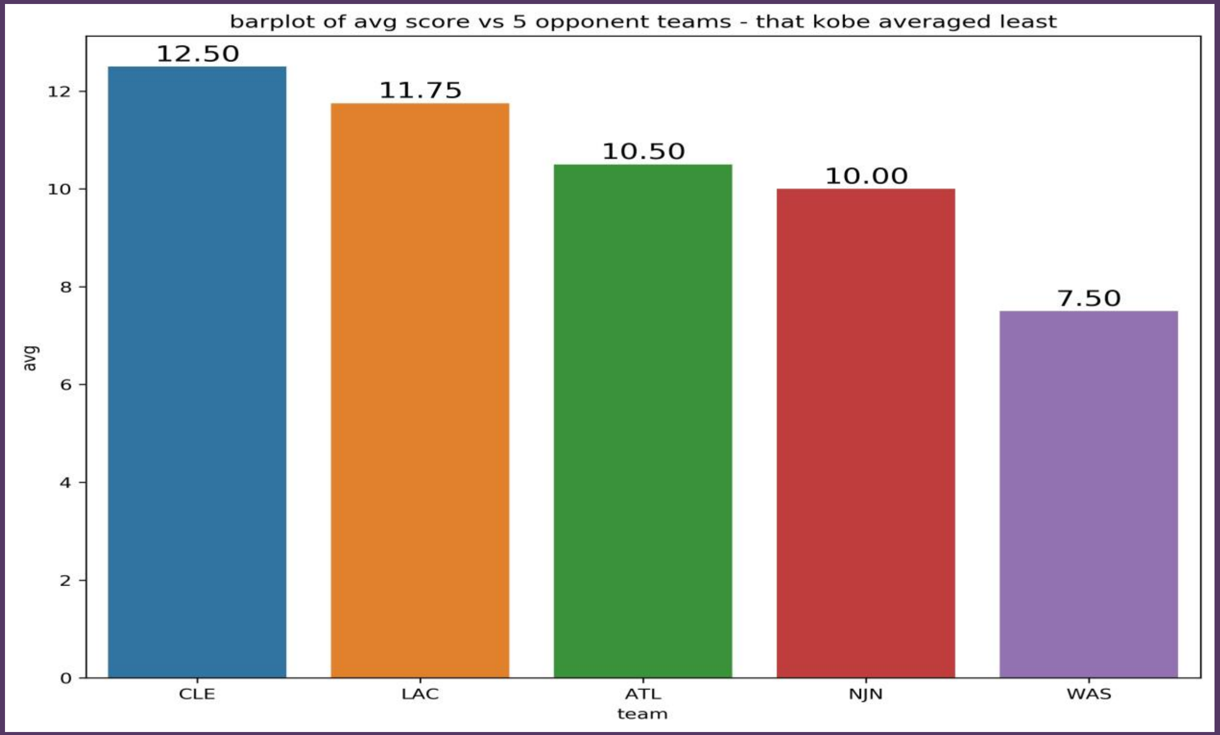
Opponent Analysis - 2002





Opponent Analysis - 2009

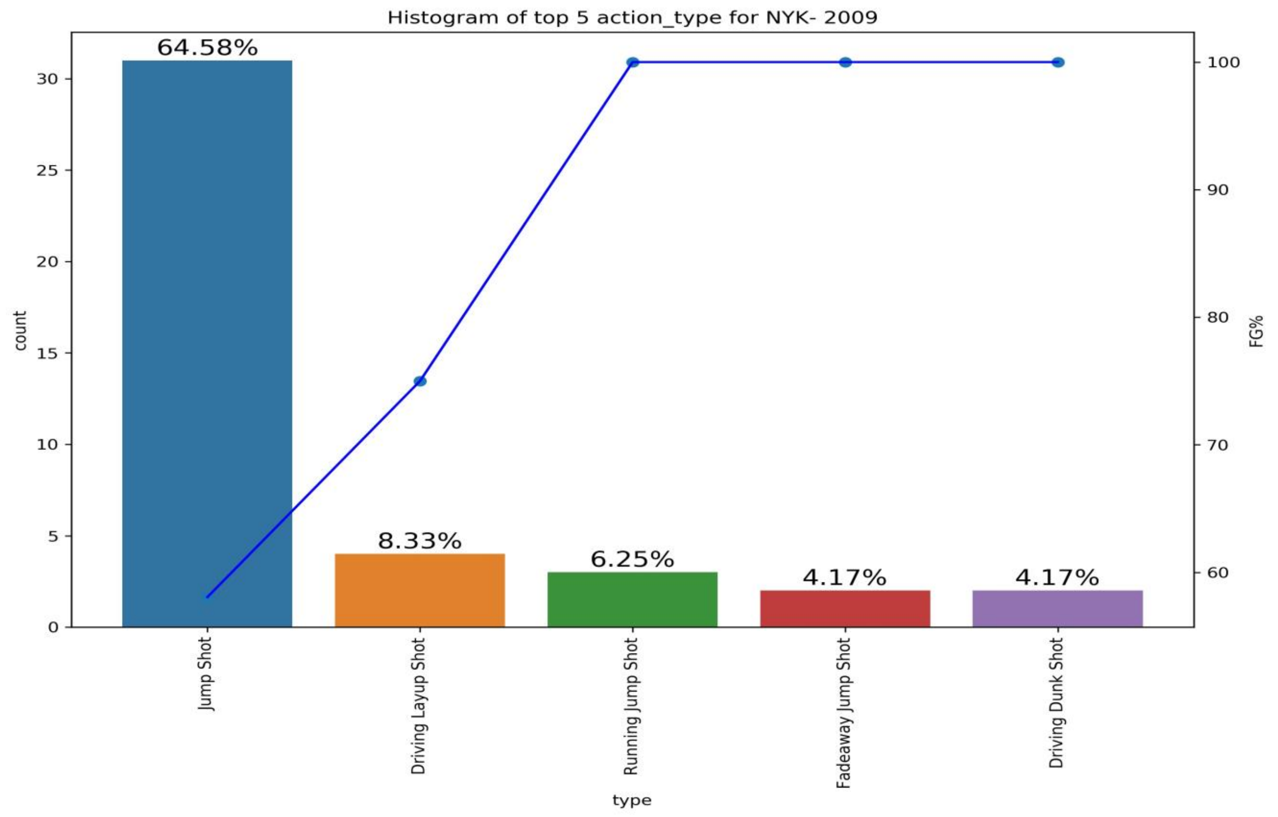


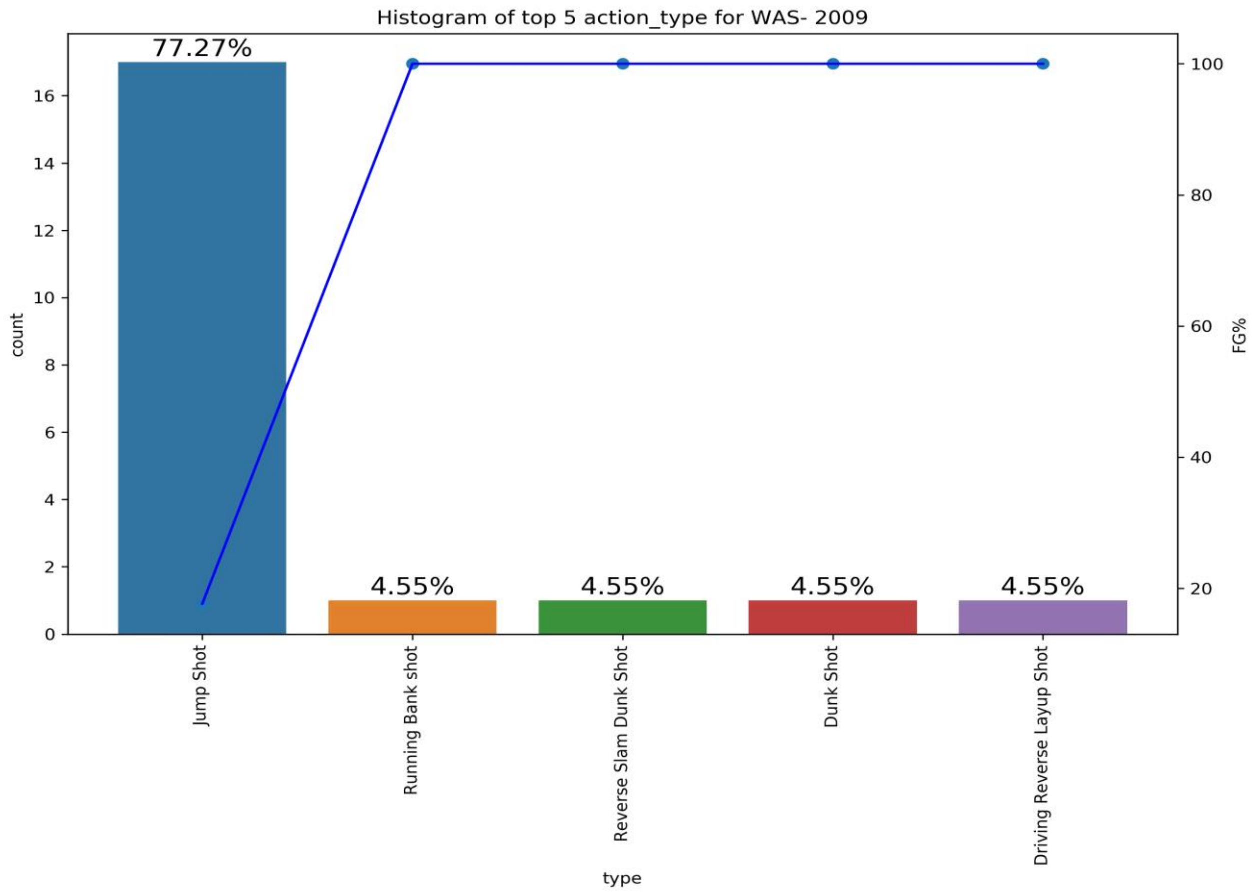


- Above are the bar graphs showing the 5 teams against which Kobe averaged the highest and 5 teams he averaged the least.
- In the year 2009, the top team that Kobe performed best is NYK(New York Knicks) and bottom team is WAS(Washington Wizards).

Opponent Analysis - 2009

- From the graph on the right, we can observe that against NYK his most used shot is jump shot (64.58%) with FG% of around 50% followed by layup shot.
- But though used less the dunks have highest FG%, so they are most effective shot type.

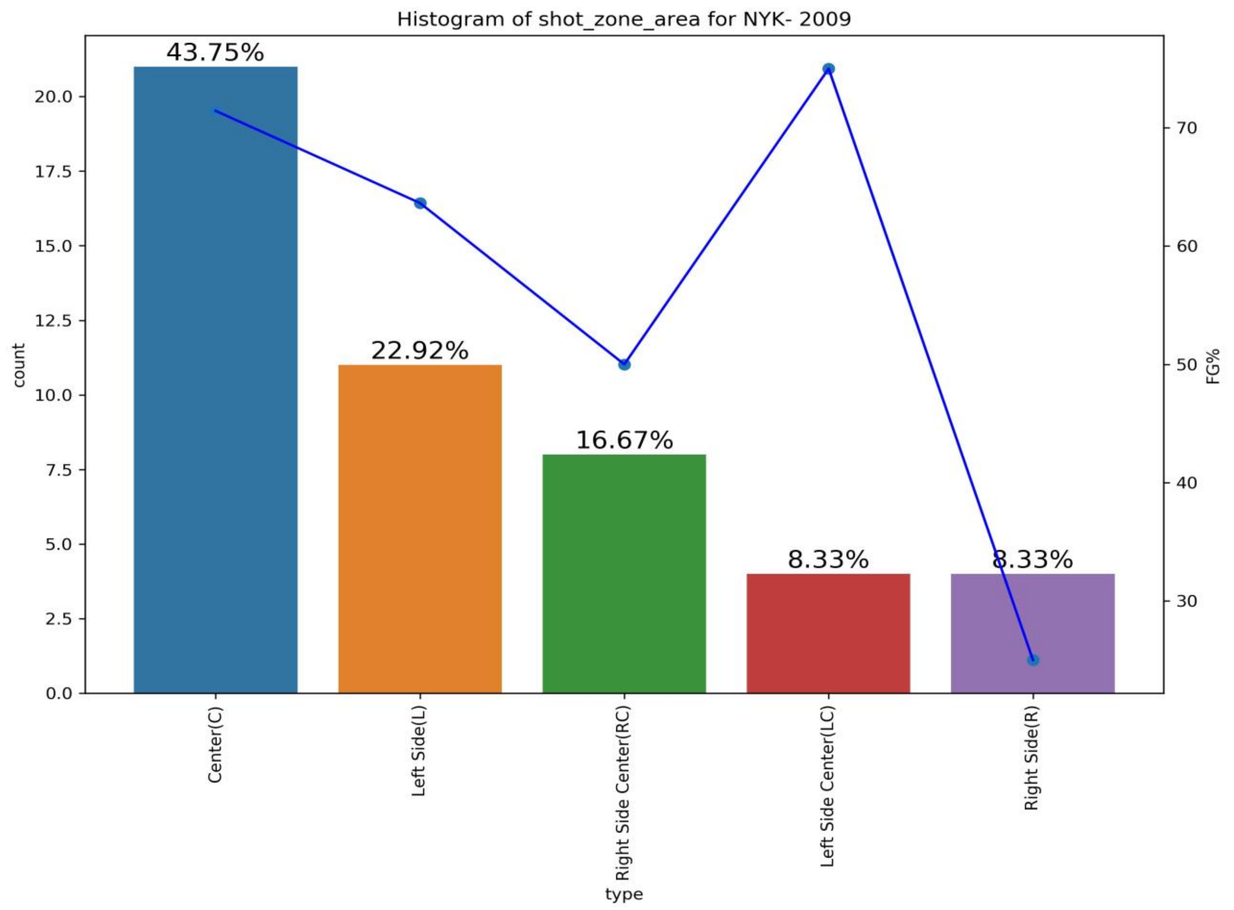


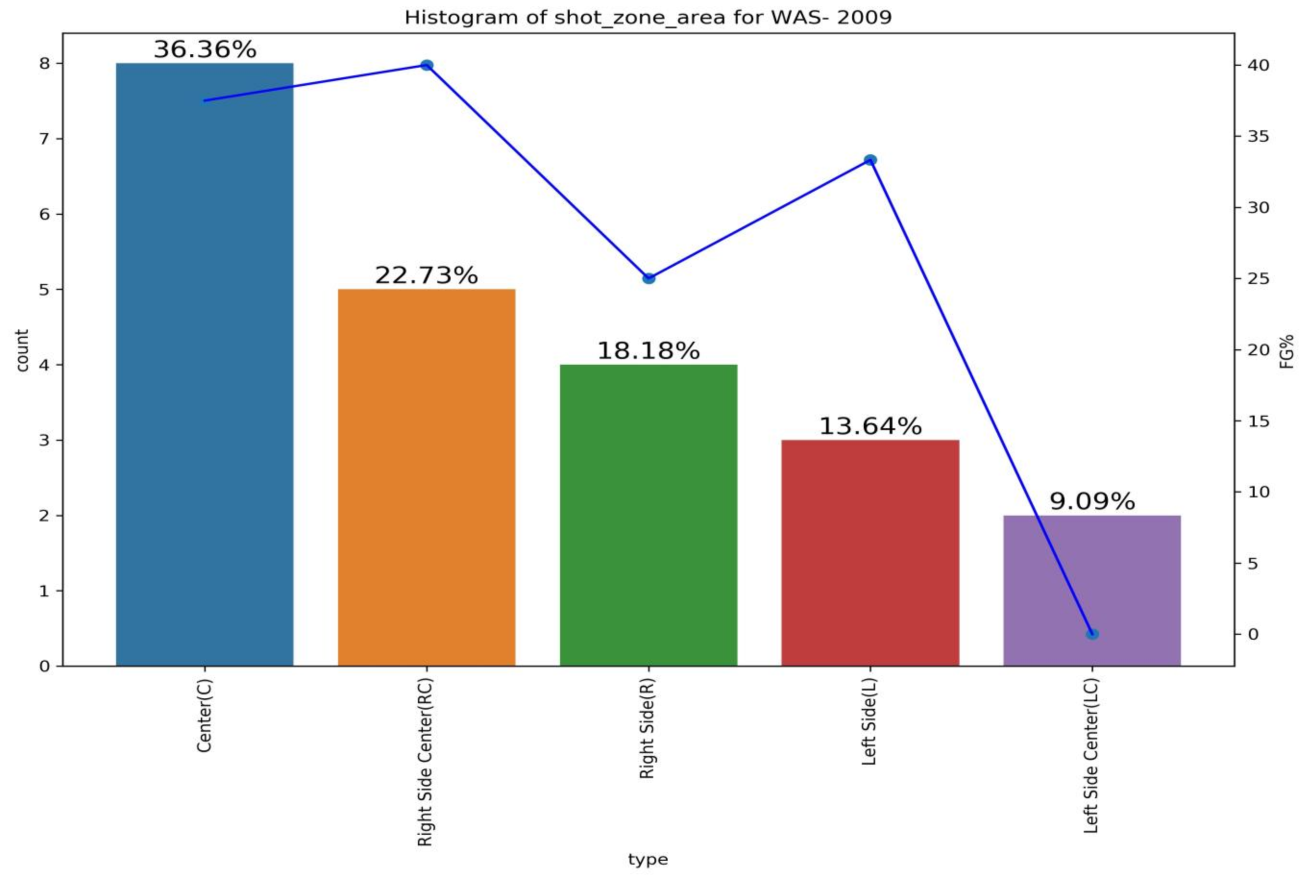


- From the graph on the left, we can observe that against WAS his most used shot is jump shot (77.27%) with FG% of around 20% followed by running Bank shot.
- From this we can observe that this team have not allowed him to shoot the jump shot with high accuracy, possibly by assigning him a taller opponent.

Opponent Analysis - 2009

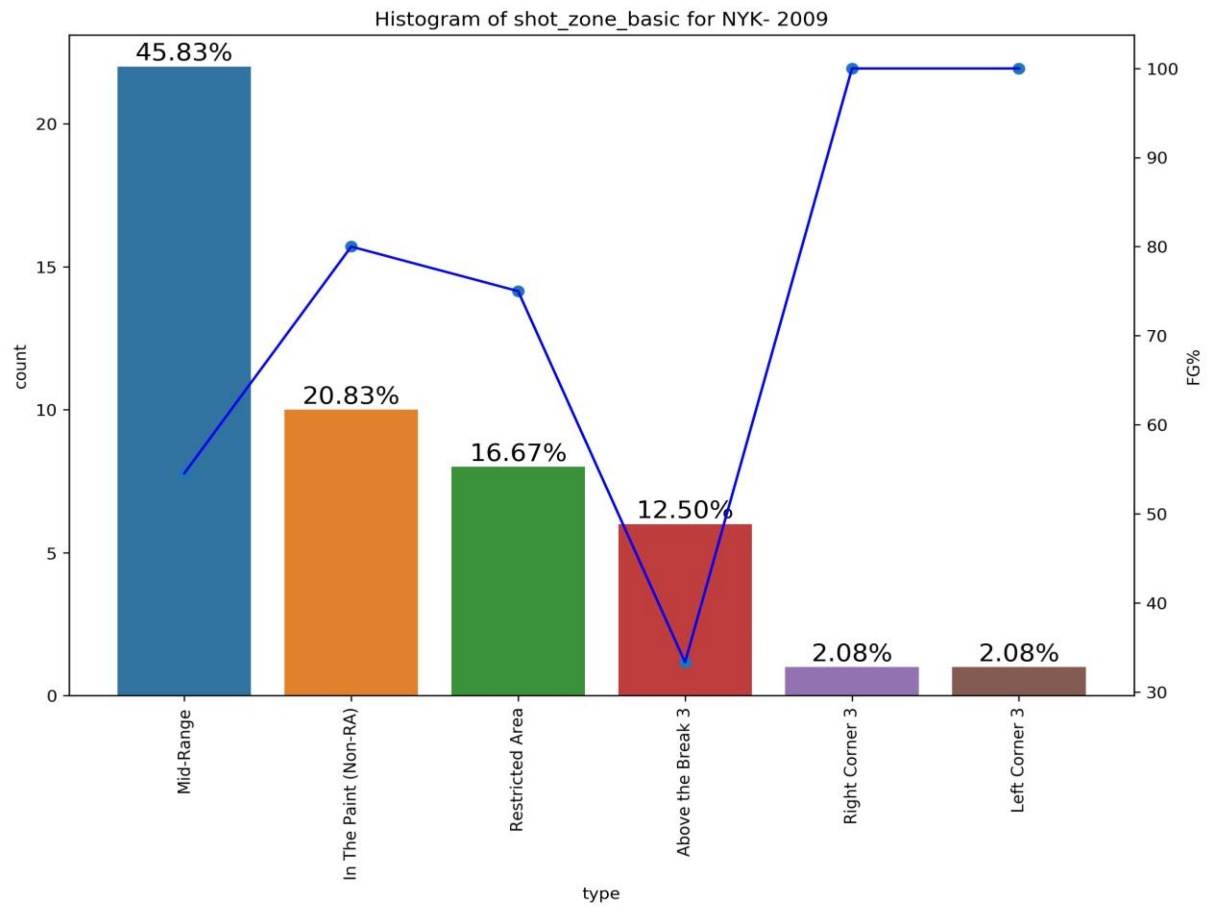
From both the graphs we can observe that Kobe shoots mostly from the Center area, also with highest FG% followed by his shots from left side so the takeaway points is to force him away from these areas to right side if possible which Washington was able to do effectively and restrict him to low average.

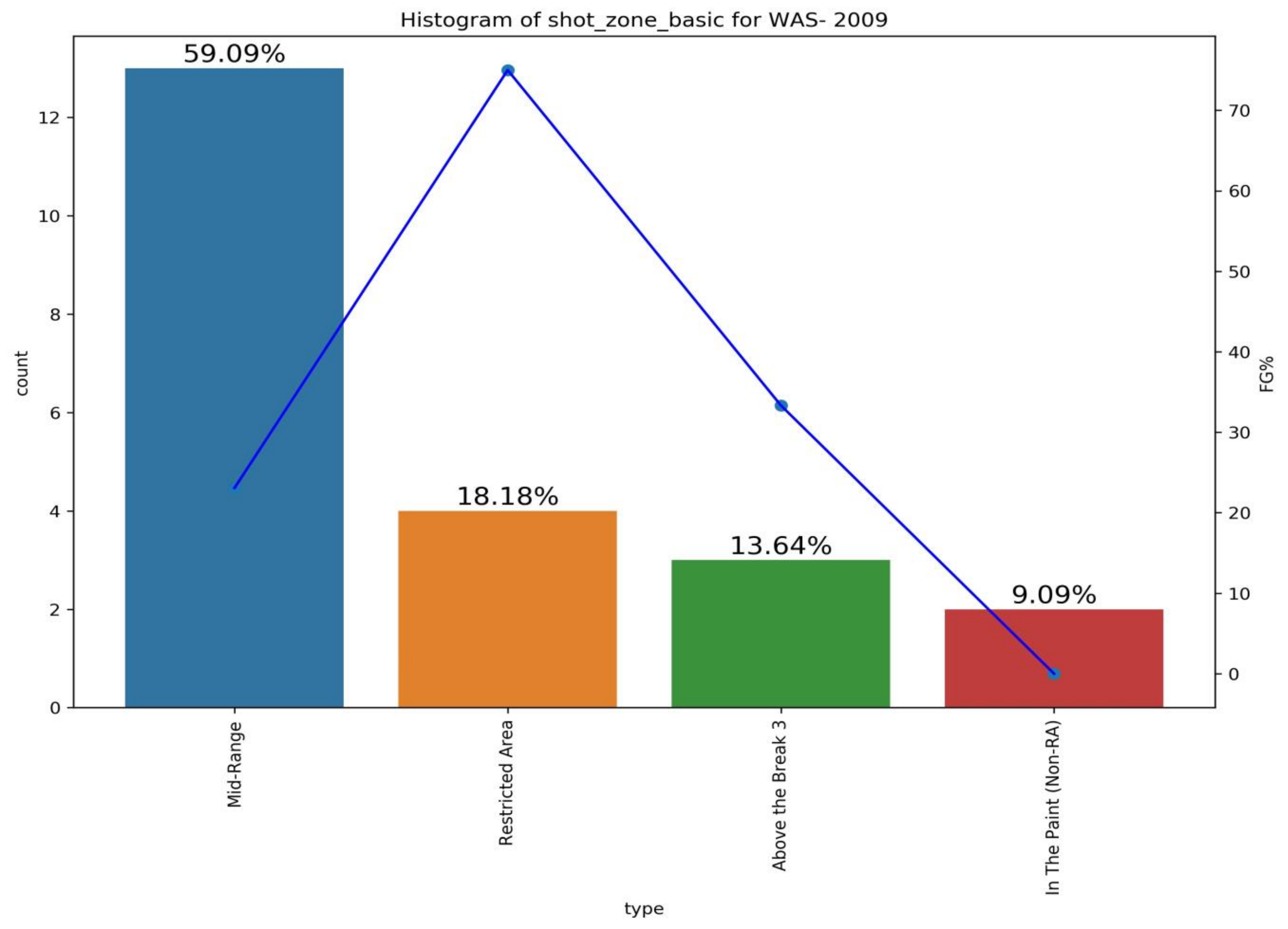




Opponent Analysis - 2009

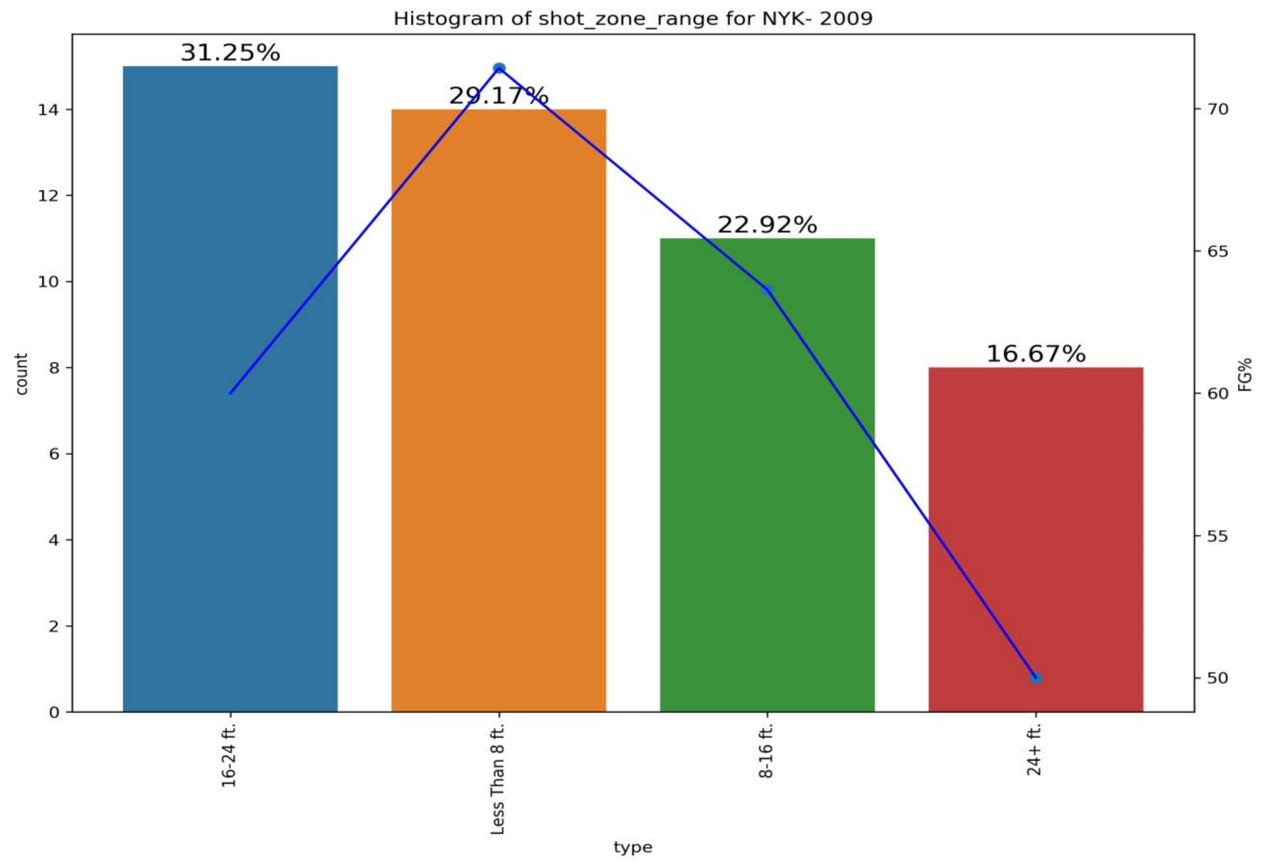
From both the graphs we can observe that Kobe shoots mostly from Mid-Range area, followed by from Restricted Area also with highest FG% so the takeaway points is to force him away to shoot from longer regions.

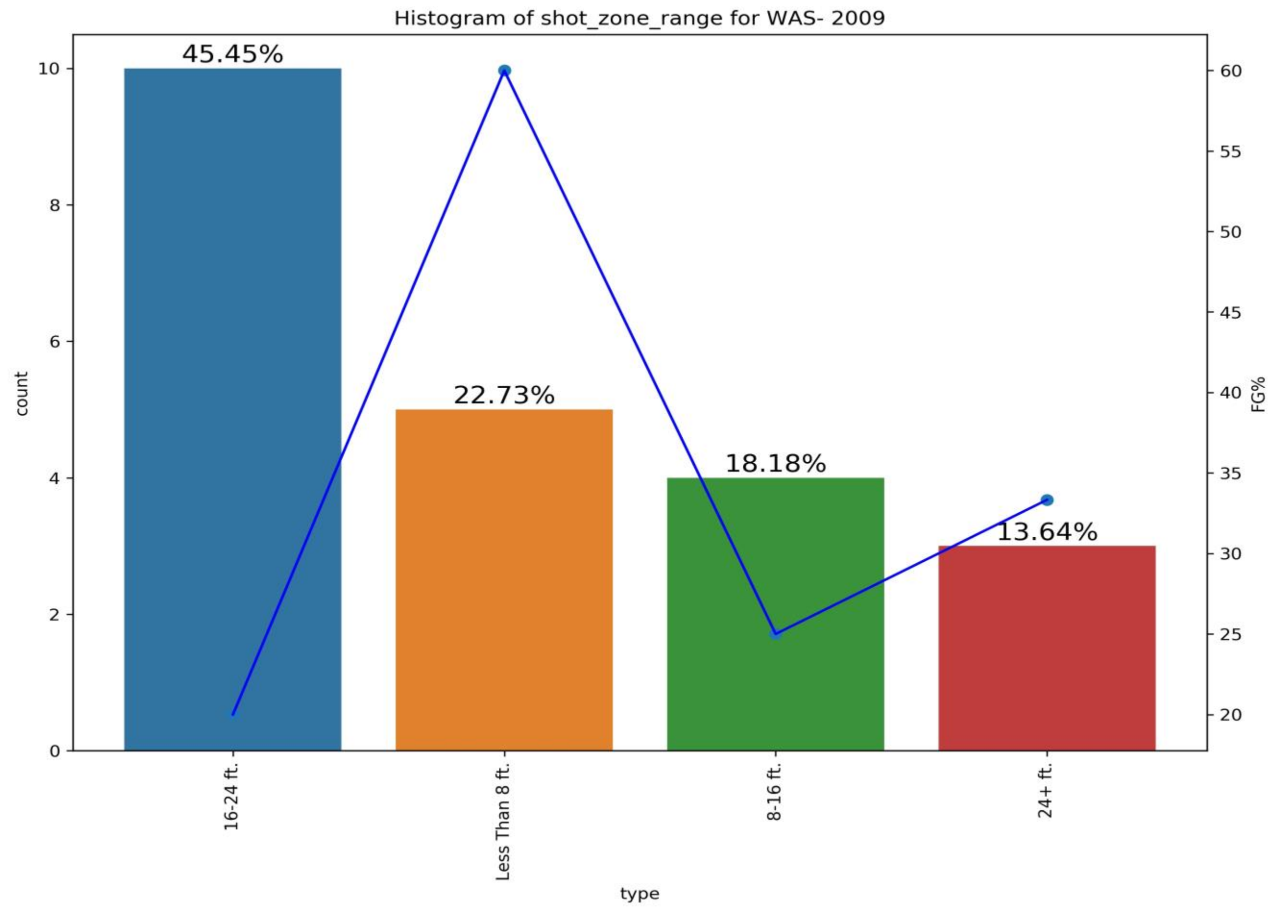




Opponent Analysis - 2009

From the both graphs, we can observe that Kobe shot most shots from ’16-24 ft’ against both teams but shots from ‘Less that 8ft’ have highest FG%.

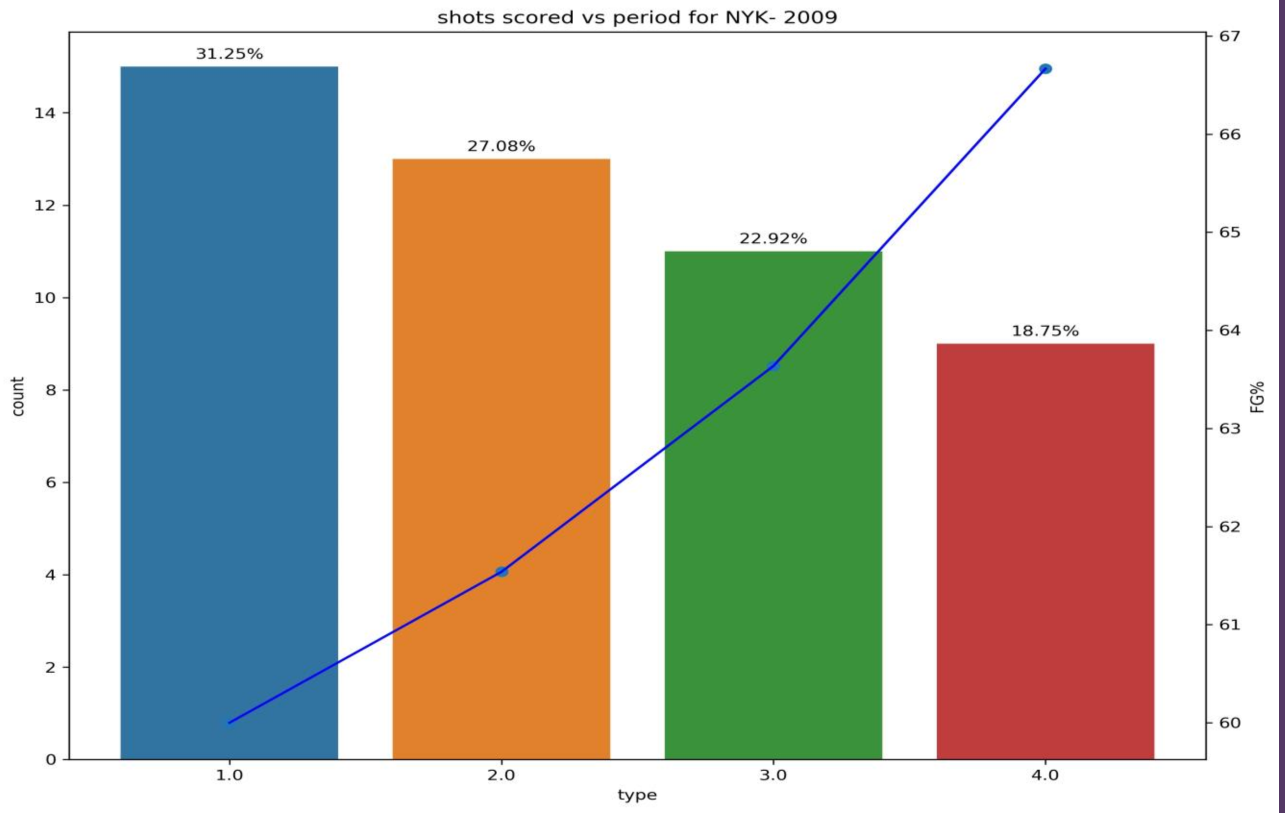


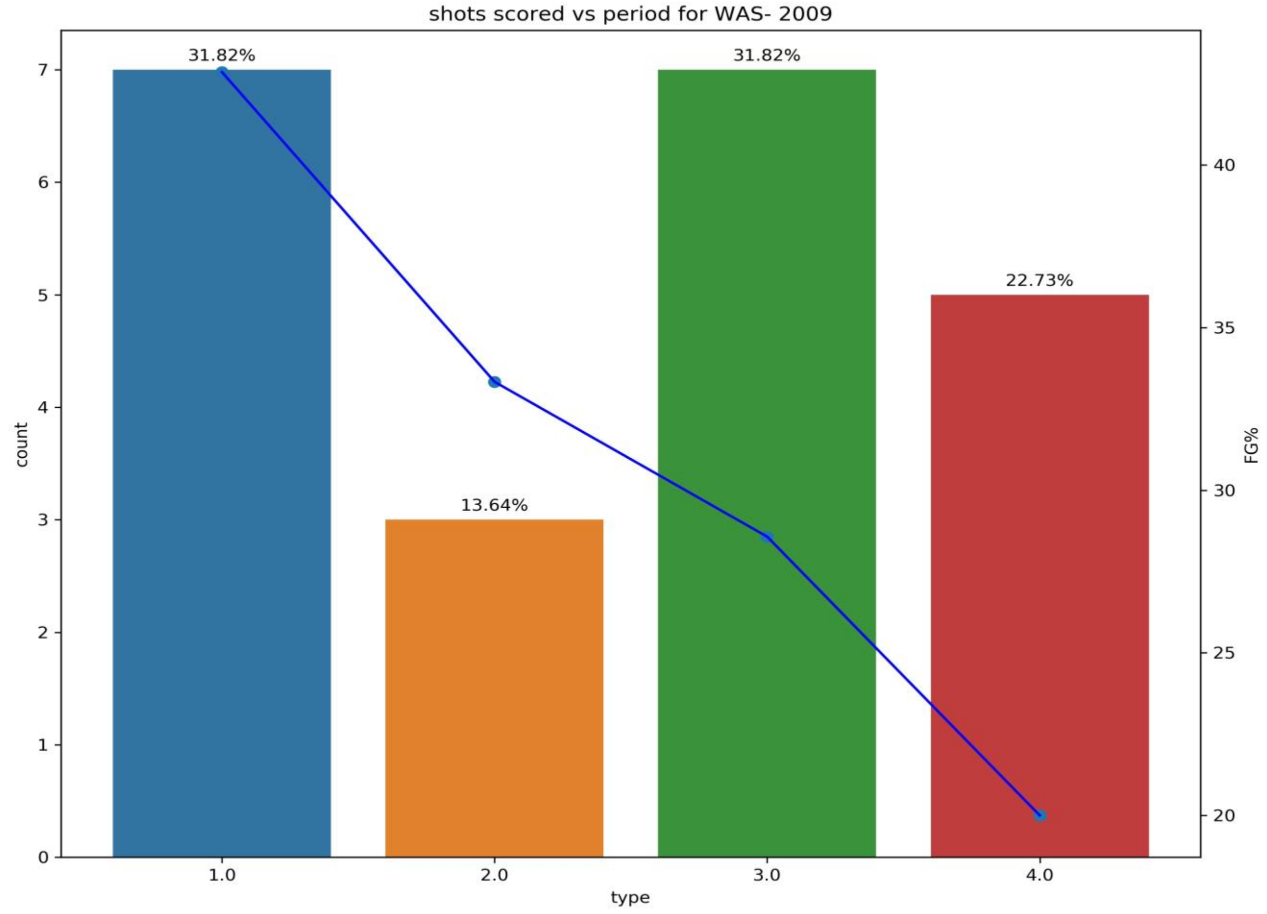


So the takeaway point is to guard him so that he does not reach ‘Less than 8ft’ area from where he is most effective.

Opponent Analysis - 2009

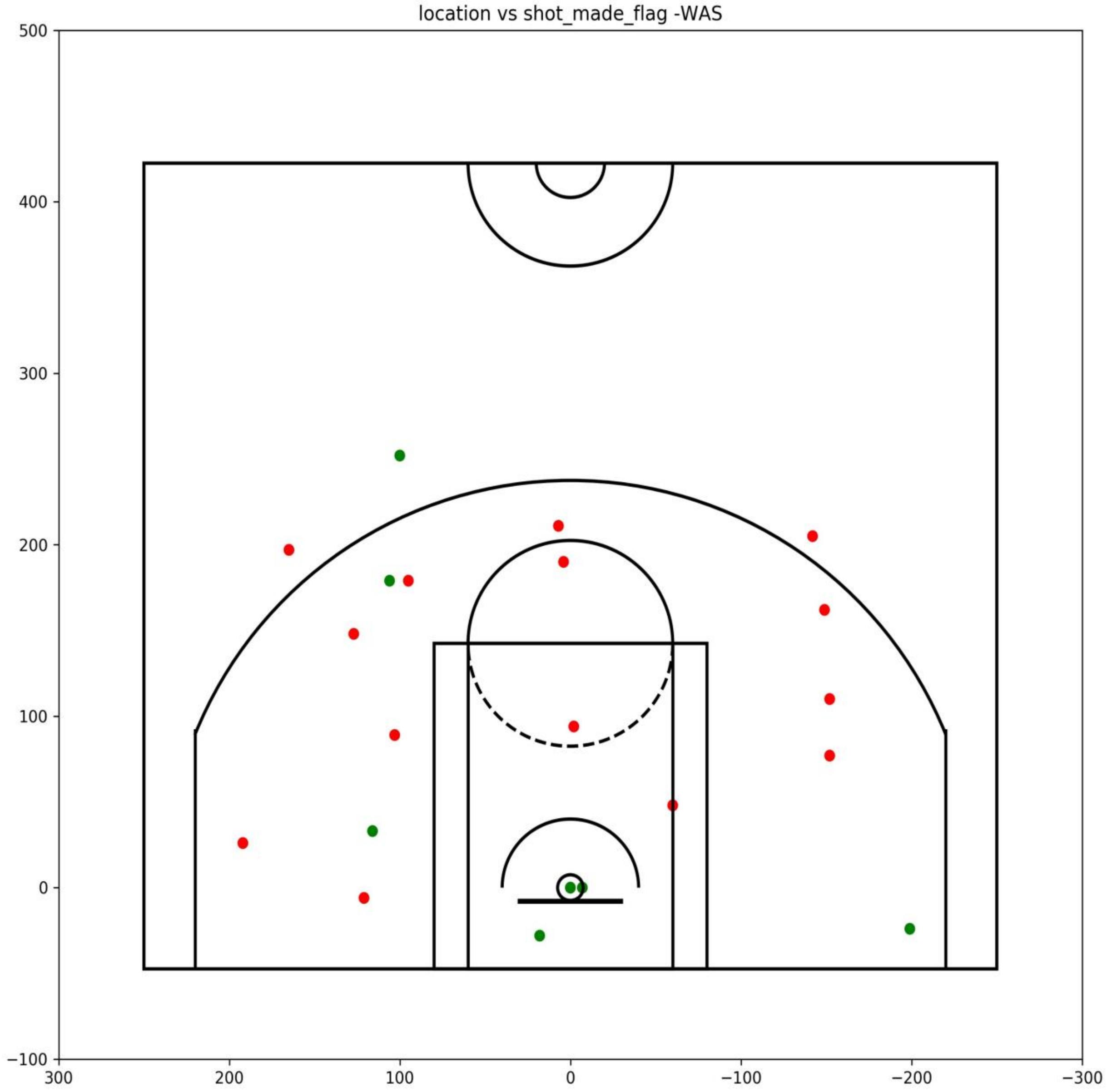
Against NYK we can observe that number of shots Kobe took decreased per quarter but his FG% increased as period changed.

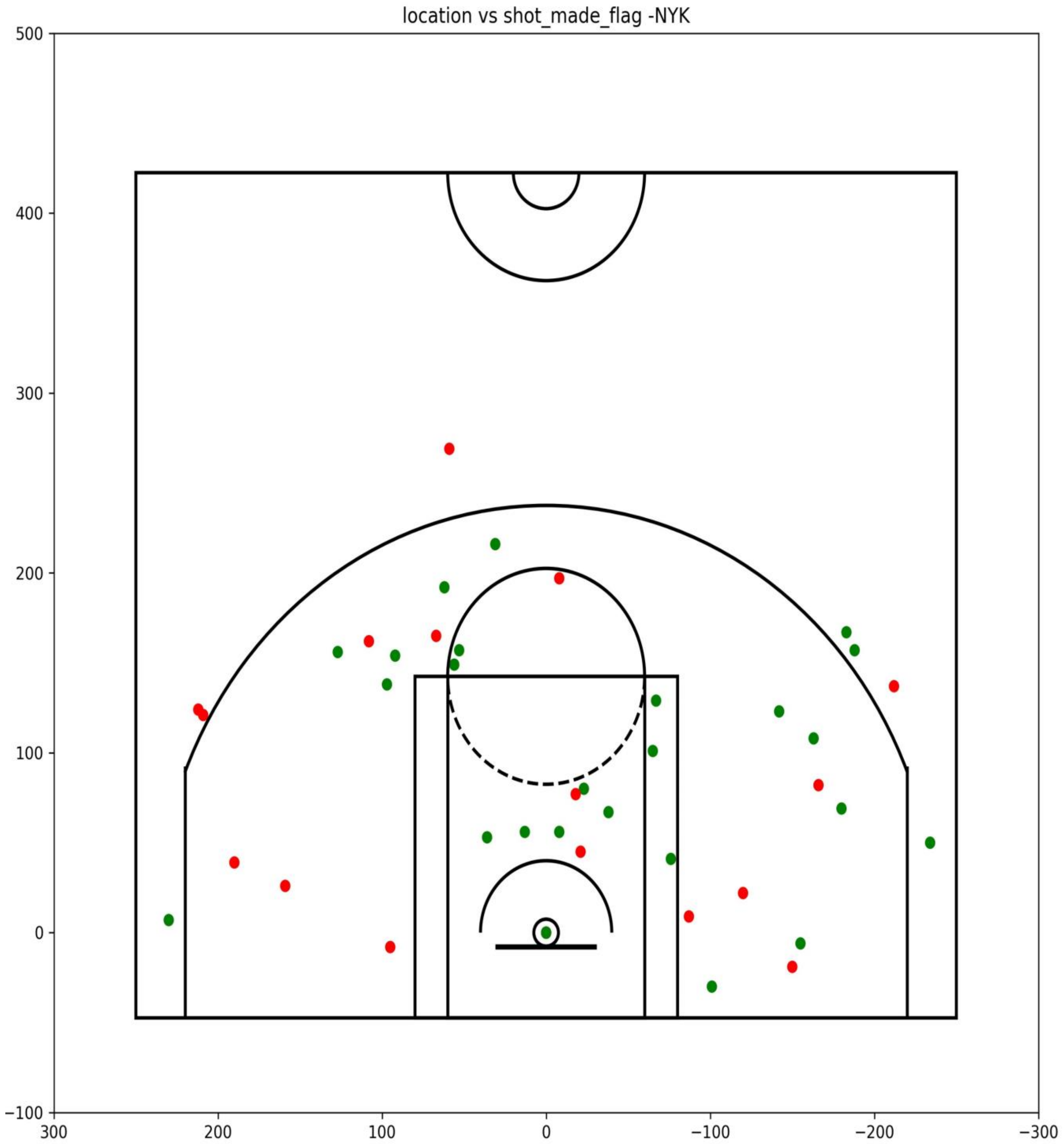




Against WAS we can observe that they have restricted him in 2nd and 4th quarter with much less FG% in all quarters, so the takeaway point is to put more pressure in 2nd quarter and make him score with less accuracy so he might perform less in following quarters where he has less energy and also additional pressure to perform more to compensate form 2nd quarter.

Opponent Analysis - 2009



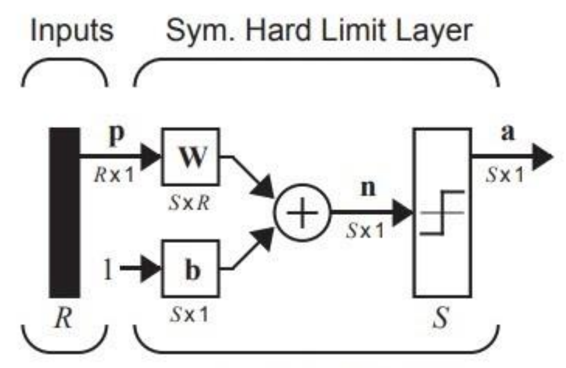


Model Building – Random Forest Classifier

- The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then it collects the votes from different decision trees to decide the final prediction.
- We can change the no-of estimators and get results in the form of classification report and confusion matrix. We can perform feature reduction by comparing the importance of the features to get more efficient models.
- The random forest classifier algorithm is being built using the Scikit Learn Package.

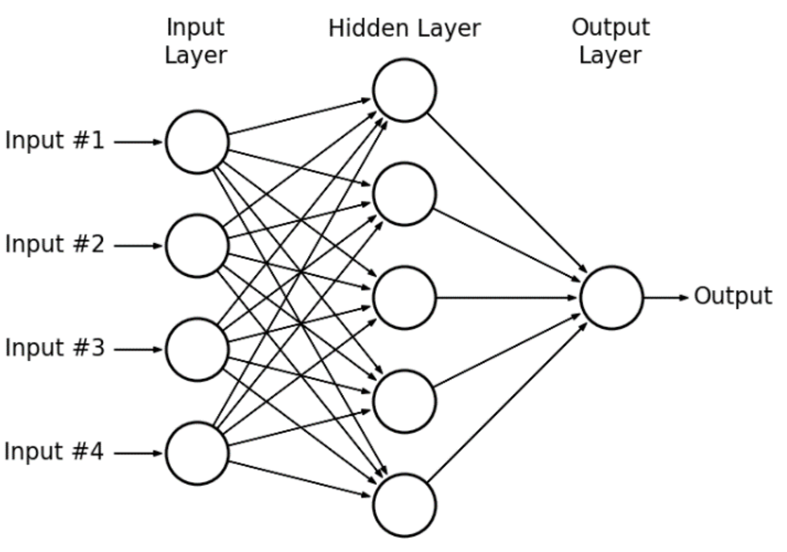
Model Building – Multi-Layer Perceptron (MLP) Classifier

- A perceptron is a neuron with a weight and bias, and the transfer function is a hardlims. This creates a single decision boundary, but we can create more complex decision boundaries through more neurons and layers.
- We can change parameters like activation, hidden layers, solver, momentum, early stopper to optimize the model
- We used the scikit-learn (sklearn) package in Python



$\mathbf{a} = \mathbf{hardlims}(\mathbf{Wp} + \mathbf{b})$

Figure 3.1 Single-Layer Perceptron



Performance measure

- Confusion matrix
- Classification report
- Accuracy
- Mean squared error
- ROC_AUC

Results – Random Forest vs MLP

model	precision		recall		accuracy	ROC_AUC
	0's	1's	0's	1's		
random forest - all features	0.64	0.56	0.67	0.53	60.67	63.47
random forest - imp 6 features	0.64	0.57	0.68	0.53	61.27	64.03
MLP - random_state=100	0.65	0.71	0.86	0.43	67	68.55
MLP - random_state=100 - imp 6 features	0.65	0.72	0.87	0.41	67	69.34

model	f1-score		confusion matrix			
	0's	1's	TN	FP	FN	TP
random forest - all features	0.65	0.54	2866	1415	1617	1812
random forest - imp 6 features	0.66	0.55	2914	1367	1619	1810
MLP - random_state=100	0.74	0.53	3070	497	1641	1217
MLP - random_state=100 - imp 6 features	0.74	0.52	3109	458	1680	1178

Results – Parameter Changes in MLP

MLP classifier - random_state=100,max_iter=200					precision		recall		accuracy	ROC_ AUC	mean squared error
activation	hidden layers	solver	momentum	early_stopping	0's	1's	0's	1's	%		
default=relu	default=(100)	default =adam	default=0.9	default=False	0.65	0.71	0.86	0.43	67	68.55	0.332
	100|100				0.65	0.72	0.87	0.42	67	68.46	0.331
	100|100|100				0.65	0.7	0.85	0.44	67	67.72	0.334
	100|100|100|100				0.65	0.66	0.81	0.46	65	66.66	0.345
logistic					0.61	0.64	0.85	0.33	62	63.23	0.38
		sgd			0.62	0.64	0.85	0.34	62	63.71	0.377
				TRUE	0.64	0.71	0.88	0.37	65	67.06	0.346
			0.95		0.65	0.71	0.86	0.43	67	68.55	0.332
logistic	100|100|100	sgd	0.95	TRUE	0.56	0	1	0	56	39.19	0.444

Results – Parameter Changes in MLP

MLP classifier - random_state=100,max_iter=200					f1-score		confusion matrix			
activation	**hidden layers**	**solver**	**momentum**	**early_stopping**	0's	1's	TN	FP	FN	TP
default=relu	default=(100)	default =adam	default=0.9	default=False	0.74	0.53	3070	497	1641	1217
	100|100				0.74	0.53	3091	476	1654	1204
	100|100|100				0.74	0.54	3019	548	1601	1257
	100|100|100|100				0.72	0.54	2899	668	1549	1309
logistic					0.71	0.43	3045	522	1922	936
		sgd			0.71	0.45	3019	548	1876	982
				TRUE	0.74	0.49	3143	424	1805	1053
			0.95		0.74	0.53	3070	497	1641	1217
logistic	100|100|100	sgd	0.95	TRUE	0.71	0	3567	0	2858	0

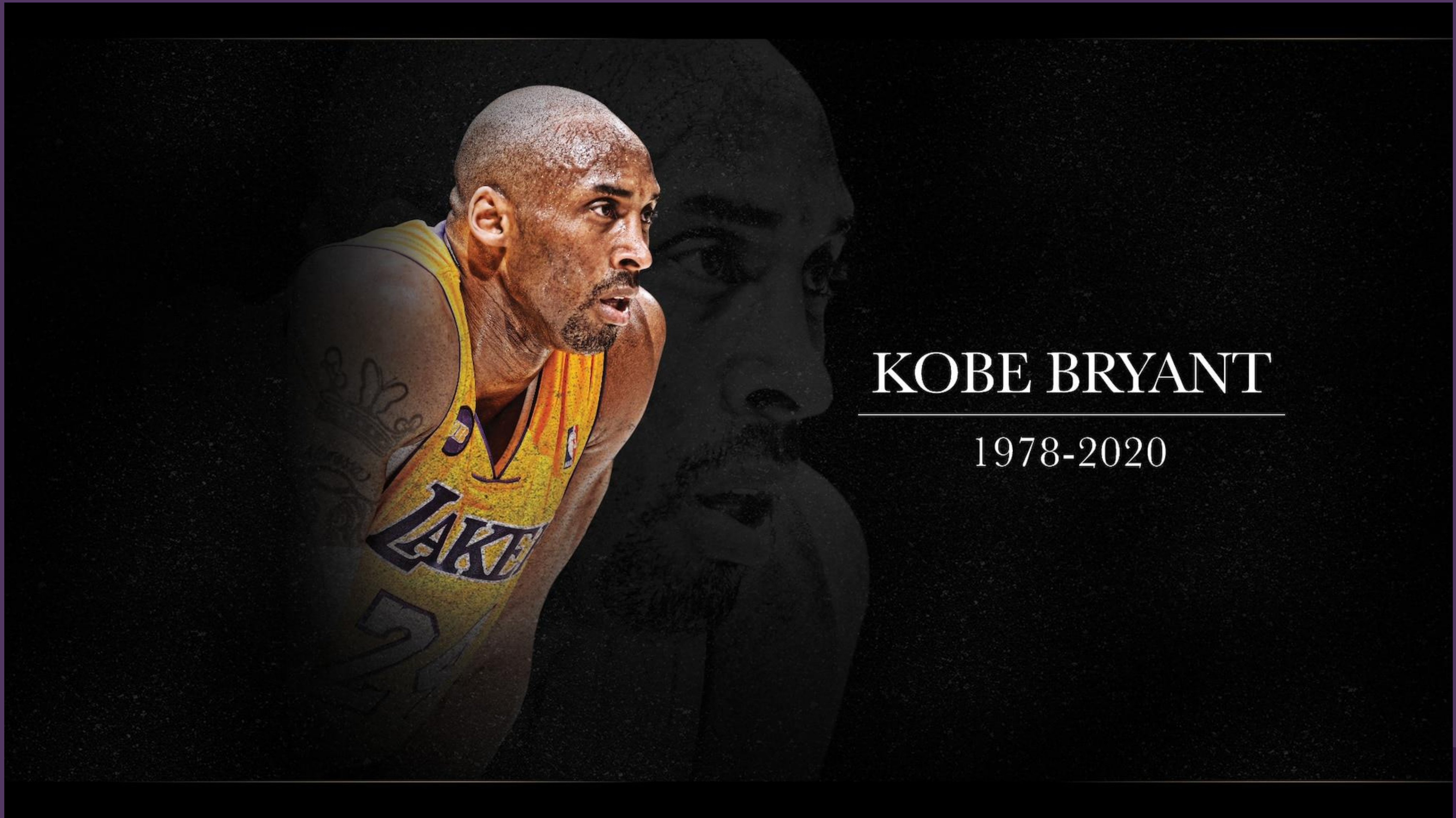
Model Summary

- Though our goal our project is to predict all the shot_made_flag variable, our focus is on the shots which are successful.
- By comparing random forest and MLP model, we can observe that precision, accuracy is higher in MLP but the f1-score and true positives is much higher in random forest model. So random forest model is better at predicting the successful shots.
- The MLP model with default parameters has fairly good results compared to all the MLP models, this model has f1-score of 0.53 and TPs of 1217.
- As we increase the number of hidden layers from default to 4 layers, the f1-score increased and TPs first decresed then increased to highest of all MLP models at 1309 for 4 layer model.
- By inclusion of activation=logistic, solver=sgd and early_stopping=True individually, the performance of the models decreases drastically. Inclusion of momentum=0.95 didn't change the results of the model.
- The worst model has activation=logistic, solver=sgd and early_stopping=True and momentum=0.95, where the f1-score,precision,recall and TPs were equal to 0.
- So the most efficient model is 3 hidden layer model as it has better f1-score, TPs than default model, but not better than 4 hidden layer model though it is faster and cost effective than the 4 hidden layer model

Conclusion

- Using a Kaggle dataset, we set out to classify Kobe Bryant's shot attempts based on various shot features.
- First, we did exploratory data analysis of our features, and we did a deeper dive on performance against specific opponents.
- We saw that Bryant's field goal percentage (made shots as a percentage of total shots) varied by shot distance, shot type, and shot zone. We decided to put those features into our model.
- For classification, we used both random forest (RF) and multi-layer perceptron (MLP) models.
- We found that MLP performed better than RF across most metrics, and this validates our decision to proceed with a neural network approach to this problem. The main area of weakness for MLP was predicting true positives.
- When looking at MLP models specifically, we experimented with activation function, hidden layers, solver, momentum and early stopping. We found the base model performed extremely well, while changing most parameters led to a decrease in model performance (the logistic activation function was extremely poor). The exception was adding layers, which generally improved performance.
- Our models did much better than simply guessing, which would have predicted "miss" and have been accurate ~55% of the time. Our best model had 67% accuracy.
- Future work on this area would include time series analysis. This would mean using past shots from the same game as well as previous games against the same opponent. That could be added into an ensemble model to improve our classification.

In Memoriam



ANY QUESTIONS?

Github Link : https://github.com/dparmar16/DATS_6202_Final_Project_Group4