Julie Athanasiadis
MSDS 458 Artificial Intelligence and Deep Learning
Dr. Syamala Srinivasan
Assignment #4
5/17/2024

Abstract

The NBA is looking to expand into a new city, and management and leadership of prospective landing spots will need to run analyses and forecast the success of the team to appeal to potential fans and the NBA. One crucial component that will need to be evaluated is salaries. This research uses box scores from NBA games from the 2019-2020 season through 2022-2023 season, for a total of 85,129 rows of data of performance metrics, date information, age, and salary. A Random Forest and k-mean model was used to create natural clusters of the data based on salary, and then an RNN model was developed to predict salary classification. After conducting many experiments, including six highlights in this paper, it was determined that LSTM and GRU models with multiple layers perform better than simple RNN and models with less layers, and dropout regularization was not useful. The highest performing model was a GRU model consisting of seven layers with 512/256/128/64/32/64 hidden neurons, batch normalization, early stopping, L2 regularization in the first dense layer, resulting in accuracy of .5396 and loss of 1.2755. The processing time was one of the top times with 13.3 mins. However, overfitting was still present with a higher training accuracy of .6825. Further research and more experimenting are encouraged to find higher performing models and address overfitting.

Introduction

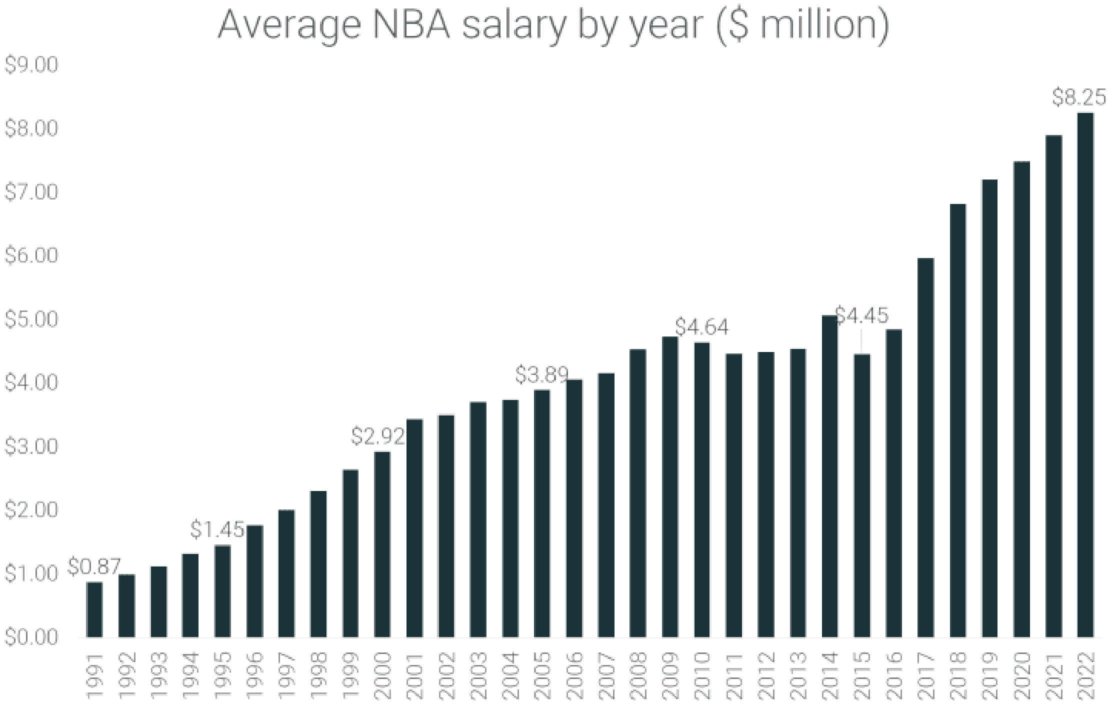
There is a short list of possible cities for an NBA expansion team, such as Nashville, Kansas City, Seatle, Louisville, and Las Vegas. Our case study will propose an approach for building an NBA expansion team in Las Vegas. Las Vegas is a large entertainment sector and a lucrative tourism destination; however, its growing sports fan base and population are the key drivers of what makes this an ideal location for a new NBA team. A Washington Post article details LeBron James' desire to own a team in this entertainment and gambling destination. After playing a preseason game at T-Mobile Arena, James acknowledged the wonderful fan support in the city, and he has the billion-dollar net worth to bring a new NBA team to life (Golliver, 2022).

Furthermore, Las Vegas has had an increasing presence in the sports industry as it is home to NFL, NHL, and WNBA teams. By sharing the T-Mobile Arena with the Vegas Golden Knights, the new NBA team would immediately have a home and excited fan base. Las Vegas was 40th on Nielsen's TV market sizes list out of 115 based "Big Four" professional sport teams and 80 college programs. ("Major Pro Sports…", 2018). This number is solely determined by the number of homes in the city, and it does not take into consideration that Las Vegas is an entertainment destination. The new team will also be able to attract tourists and build a fanbase with existing and new residents moving to the growing city. For our research, we are focusing on six key components: player acquisition, salary, performance, sponsorship, attendance, and customer relationship management. To help predict salaries for the upcoming season, we applied a deep learning model looking at past performance and past salaries.

Literature Review

Navigating and understanding NBA teams' financials and player contracts is a hard task, and there are times when you need a glossary of terms. For example, a team can reach four types of salary ceilings – cap, luxury cap, first apron, and second apron. Restrictions and tax sanctions are implemented to keep NBA teams within an appropriate amount; however, every team went over the cap in the 2023-2024 season. Additionally, there is a clear correlation between spending and winning, with twelve of the top fifteen spending teams making playoffs in 2023-2024. Team valuations are increasing, as so are player contracts, making managing salaries difficult for teams to balance. For example, supermax deals are designed to help retain loyal top talent by allowing teams to re-sign a player who has played at least seven years with that team, for a maximum of five years for an amount up to 35% of the salary cap subject to an 8% annual increase. (Sportrac) There are certain conditions that need to be met for a player to qualify, such as making all NBA-team or becoming MVP. Jaylen Brown just signed a supermax deal worth $286,230,000 over 5 years, and Luka Doncic is poised to be the first player to reach $70/million a year when his opportunity for a supermax contract comes in the 2025 offseason. (*Luka Doncic 2025*, 2024) Figure 1 shows the rising average NBA salary for the past 30 years.

Figure 1



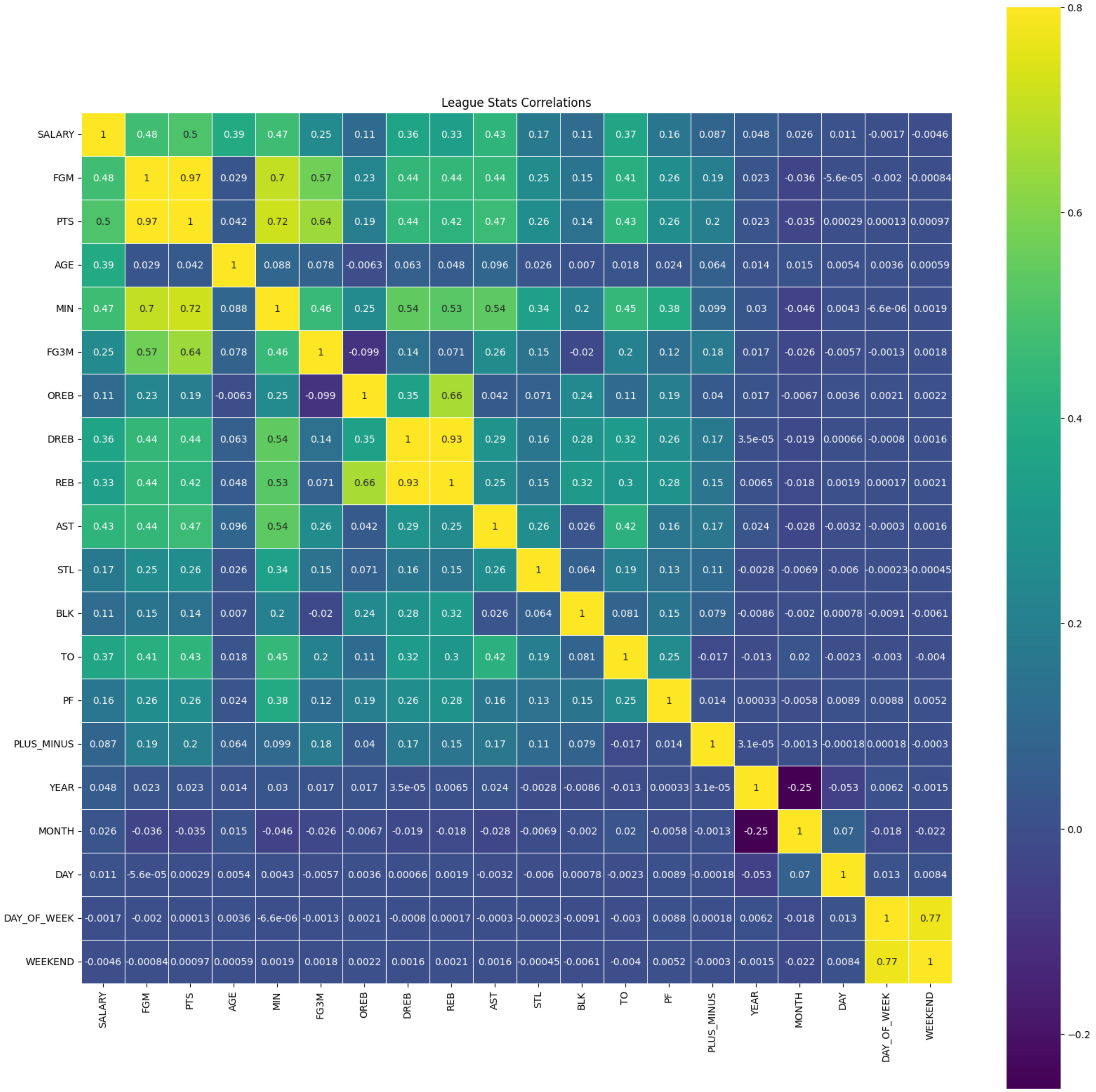
NBA Salaries Analysis (1991-2022). https://runrepeat.com/salary-analysis-in-the-nba-1991-2019

Forecasting salaries are also challenging due to the different types of player contracts, market forces, player health, aging factors, and player marketability. There are multiple approaches to projecting and predicting salaries. With their paper, "Classification of NBA Salaries through Player Statistics," the Berkeley Sports Analytics Group used K-means classification methods to predict free agency salaries with performance metrics. Some researchers used machine learning and regression models to predict NBA salaries. (Wu et al.) One author found that Random Forest and Gradient Boosting techniques performed better than Support Vector Machine (SVM), Elastic Net, Adaboost, Light Gradient Boosting Machine (LGBM) when predicting free agents' salaries and discovered that the leading factors where minutes played, points, previous season salary, and advances player statistics. (Pastorello, 2023) For the purpose of our research, we evaluated all players from the previous five seasons and not just free agents. Hoping that this will improve our model's performance in a short time of preparation.

Methods

Many experiments were conducted using the Google Colab platform and python programming language; however, six were chosen for this evaluation. Using an external machine and not a personal one allowed for quicker running times. Natural language toolkit (NLKT), TensorFlow, Keras, ScktitLearn, Numpy, Pandas, Seaborn, and Matplotlib libraries were used to conduct and display various executions of data throughout the experiments. Each experiment started with importing important libraries and software. For data collection, we obtained and cleaned NBA box score data from each game in the 2019-2020 season through the 2022-2023 season. Our data was retrieved from Kaggle and Sportrac, and Basketball-Reference websites, and consisted of 85,129 rows. Earlier versions of the model did not have enough data to produce a significant model. Therefore, we had changed our collection from averages of players each season, to player's performance each game. An EDA was conducted to examine the data before we ran the models. Key performance metrics used for the analysis were points, minutes played, assists, offensive rebounds, defensive rebounds, total rebounds, turnovers, steals, blocks, field goals made, 3PT field goals made, personal fouls, and an advance metric plus minus. Unsurprisingly, all performance indicators have a positive relationship with minutes played because the more you play, the more these will add up, even the negative indicators, such as fouls and turnovers. Similarly, these had positive correlations with Salary. We later added in date information, such as year, month, and day, in attempts to make the model stronger. Correlations between all features are highlighted in a matrix below in figure 2. The date related correlations are not as strong in the matrix as compared to performance, but you will see the overall relative importance to salary in figure 3.

Figure 2



	SALARY	FGM	PTS	AGE	MIN	FG3M	OREB	DREB	REB	AST	STL	BLK	TO	PF	PLUS_MINUS	YEAR	MONTH	DAY	DAY_OF_WEEK	WEEKEND
SALARY	1	0.48	0.5	0.39	0.47	0.25	0.11	0.36	0.33	0.43	0.17	0.11	0.37	0.16	0.087	0.048	0.026	0.011	-0.0017	-0.0046
FGM	0.48	1	0.97	0.029	0.7	0.57	0.23	0.44	0.44	0.44	0.25	0.15	0.41	0.26	0.19	0.023	-0.036	-5.6e-05	-0.002	-0.00084
PTS	0.5	0.97	1	0.042	0.72	0.64	0.19	0.44	0.42	0.47	0.26	0.14	0.43	0.26	0.2	0.023	-0.035	0.00029	0.00013	0.00097
AGE	0.39	0.029	0.042	1	0.088	0.078	-0.0063	0.063	0.048	0.096	0.026	0.007	0.018	0.024	0.064	0.014	0.015	0.0054	0.0036	0.00059
MIN	0.47	0.7	0.72	0.088	1	0.46	0.25	0.54	0.53	0.54	0.34	0.2	0.45	0.38	0.099	0.03	-0.046	0.0043	-6.6e-06	0.0019
FG3M	0.25	0.57	0.64	0.078	0.46	1	-0.099	0.14	0.071	0.26	0.15	-0.02	0.2	0.12	0.18	0.017	-0.026	-0.0057	-0.0013	0.0018
OREB	0.11	0.23	0.19	-0.0063	0.25	-0.099	1	0.35	0.66	0.042	0.071	0.24	0.11	0.19	0.04	0.017	-0.0067	0.0036	0.0021	0.0022
DREB	0.36	0.44	0.44	0.063	0.54	0.14	0.35	1	0.93	0.29	0.16	0.28	0.32	0.26	0.17	3.5e-05	-0.019	0.00066	-0.0008	0.0016
REB	0.33	0.44	0.42	0.048	0.53	0.071	0.66	0.93	1	0.25	0.15	0.32	0.3	0.28	0.15	0.0065	-0.018	0.0019	0.00017	0.0021
AST	0.43	0.44	0.47	0.096	0.54	0.26	0.042	0.29	0.25	1	0.26	0.026	0.42	0.16	0.17	0.024	-0.028	-0.0032	-0.0003	0.0016
STL	0.17	0.25	0.26	0.026	0.34	0.15	0.071	0.16	0.15	0.26	1	0.064	0.19	0.13	0.11	-0.0028	-0.0069	-0.006	-0.00023	-0.00045
BLK	0.11	0.15	0.14	0.007	0.2	-0.02	0.24	0.28	0.32	0.026	0.064	1	0.081	0.15	0.079	-0.0086	-0.002	0.00078	-0.0091	-0.0061
TO	0.37	0.41	0.43	0.018	0.45	0.2	0.11	0.32	0.3	0.42	0.19	0.081	1	0.25	-0.017	-0.013	0.02	-0.0023	-0.003	-0.004
PF	0.16	0.26	0.26	0.024	0.38	0.12	0.19	0.26	0.28	0.16	0.13	0.15	0.25	1	0.014	0.00033	-0.0058	0.0089	0.0088	0.0052
PLUS_MINUS	0.087	0.19	0.2	0.064	0.099	0.18	0.04	0.17	0.15	0.17	0.11	0.079	-0.017	0.014	1	3.1e-05	-0.0013	-0.00018	0.00018	-0.0003
YEAR	0.048	0.023	0.023	0.014	0.03	0.017	0.017	3.5e-05	0.0065	0.024	-0.0028	-0.0086	-0.013	0.00033	3.1e-05	1	-0.25	-0.053	0.0062	-0.0015
MONTH	0.026	-0.036	-0.035	0.015	-0.046	-0.026	-0.0067	-0.019	-0.018	-0.028	-0.0069	-0.002	0.02	-0.0058	-0.0013	-0.25	1	0.07	-0.018	-0.022
DAY	0.011	-5.6e-05	0.00029	0.0054	0.0043	-0.0057	0.0036	0.00066	0.0019	-0.0032	-0.006	0.00078	-0.0023	0.0089	-0.00018	-0.053	0.07	1	0.013	0.0084
DAY_OF_WEEK	-0.0017	-0.002	0.00013	0.0036	-6.6e-06	-0.0013	0.0021	-0.0008	0.00017	-0.0003	-0.00023	-0.0091	-0.003	0.0088	0.00018	0.0062	-0.018	0.013	1	0.77
WEEKEND	-0.0046	-0.00084	0.00097	0.00059	0.0019	0.0018	0.0022	0.0016	0.0021	0.0016	-0.00045	-0.0061	-0.004	0.0052	-0.0003	-0.0015	-0.022	0.0084	0.77	1

Figure 3

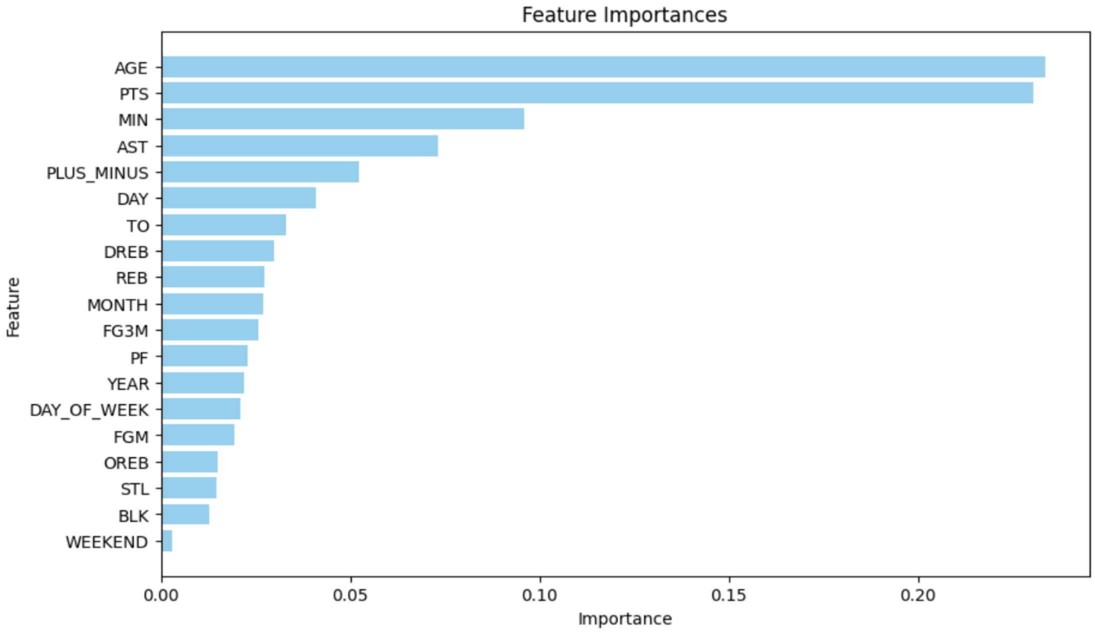
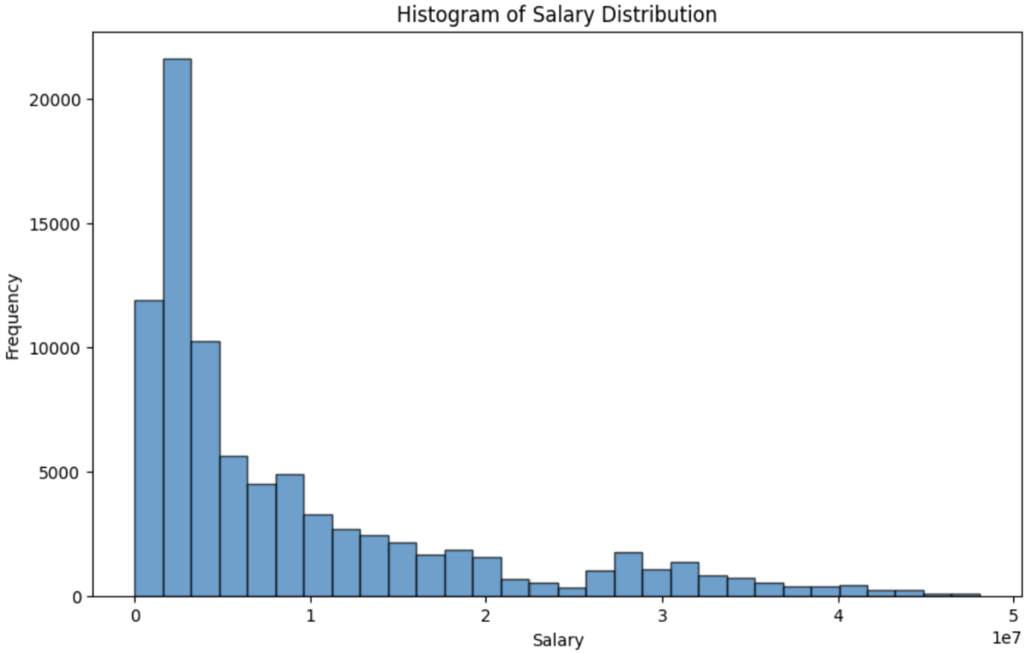


Figure 4 shows the salary distribution, which is rightly skewed, with 75% of the salaries below $12,420,000. The highest salary from this dataset is $48,070,014.

Figure 4



We then ran a random forest model and k-means to determine the natural clusters from the data. We used these to set our bin boundaries based on salaries. Then we split the data into testing, training, and validation sets. We applied the model with GRU layers and trained the model on the training set.

The initial model used 7 GRU layers (5 layers and 2 dense) with units 512/256/128/64/32/64, activation ReLU for all GRU layers and Softmax for the output layer, epoch size of 200, and batch size of 64. It also used regularization methods early stopping, batch normalization, and L2 on the output layer. Early experiments applied techniques such as bidirectional and unidirectional, batch sizes of 32, various number of layers and units, and optimizer Adam, but these attempts were unsuccessful at significantly raising the performance. Experiments 2-6, highlighted in this paper, saw changes in the type of model, performance variables selected, and type of regularization. The models then were compiled with the metric of accuracy, the loss function of sparse categorical cross entropy, and set with the optimizer of Adam. The last section of the experiments included the evaluation of the model performance. Training and validation accuracy and loss were calculated and plotted on a graph for a visual representation, and confusion matrices were created to help display the true positives for the correct identifications of classifications.

Experiment #1 and #2

For experiment 1 we used all features to determine the initial clusters. Due to overfitting, in experiment 2 we removed all the date related features (except year) as we thought this information may have been bringing too much noise to the data set. Overall, this assumption was not accurate for our model at this stage, so we returned to all features in experiments 3-6. The contrasting salary bins and PCA for Experiment 1 and 3-6 and experiment 2 is seen below in figures 5-8.

Figure 5 - all features

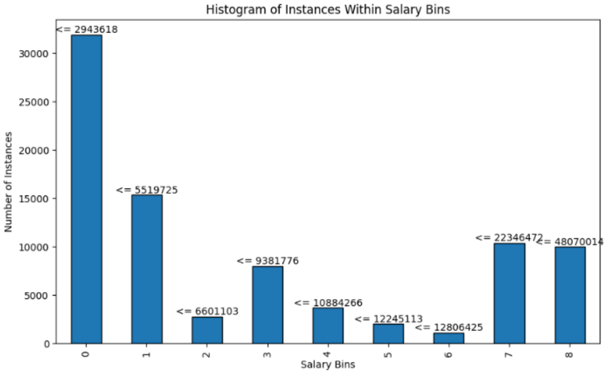


Figure 6 - without month, day, or weekend

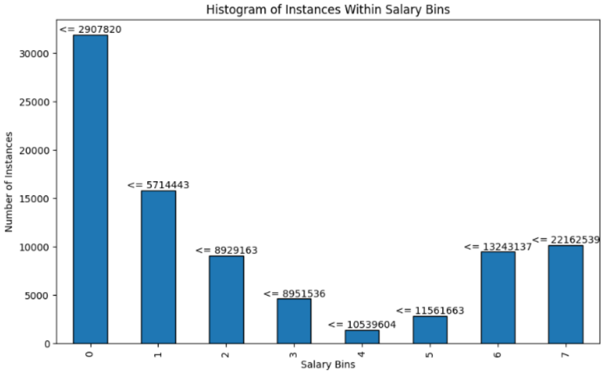


Figure 7 – all features

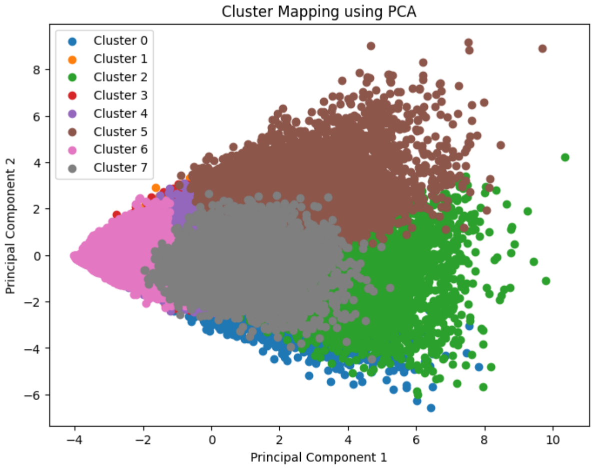
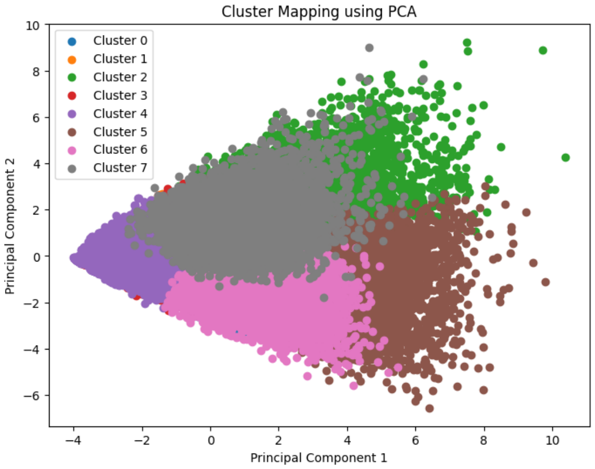


Figure 8 – without month, day, or weekend



Experiment one had a training accuracy of .6825 and loss of .8646, but a testing accuracy of .5396 and testing loss of 1.2755. Experiment two had a testing accuracy of .5187 and loss of 1.2814, suggesting that removing the date features did not improve the model. Both models had similar validation accuracy ~.52.

Experiment #3

For this experiment, we added back all the features and added dropout to address the overfitting issue. We removed batch normalization and added dropout .2 to every layer and it increased the time of the model significantly from 13.2 minutes to 42.7 minutes. However, this did not improve the accuracy of the model and it went down to training accuracy of .5604 and testing accuracy of .5239. The training loss increased as well to 1.2256, but the testing loss decreased to 1.331.

Experiment #4

Experiment 4 brought back batch normalization, removed dropout, and added L2 regularization in all layers, not just the first dense layer. This was our longest running model yet with 68.1 minutes and was the least performing model of the six with a training accuracy of .5082, and testing accuracy of .5019. It seemed to have helped the overfitting issue well, but the overall performance of the model decreased. Experiment 1 and Experiment 4 training and validation plots are compared below to show how it helped the overfitting.

Figure 9

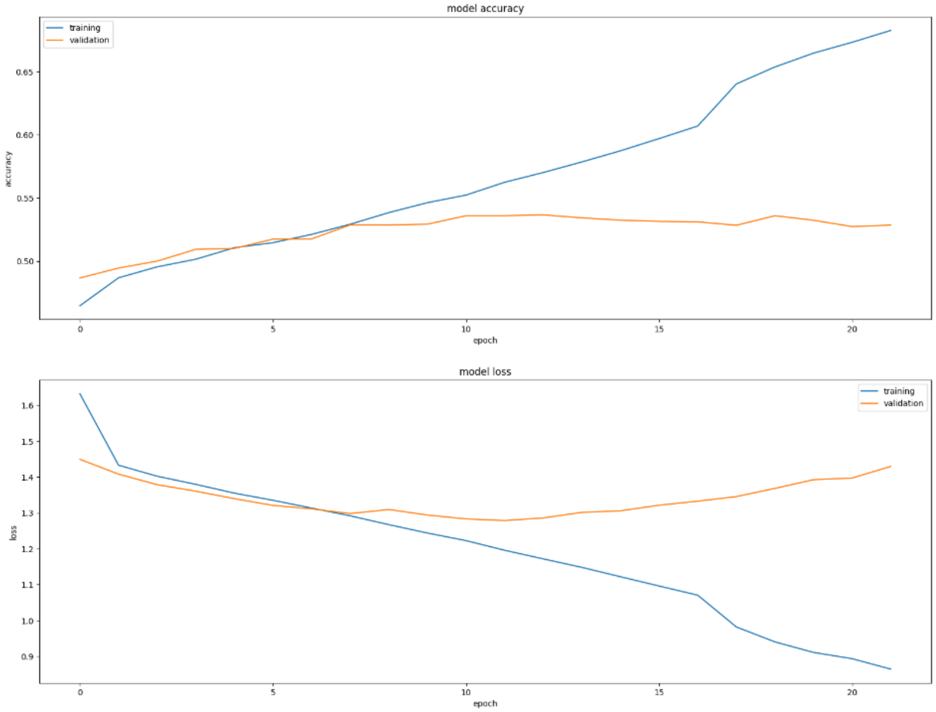
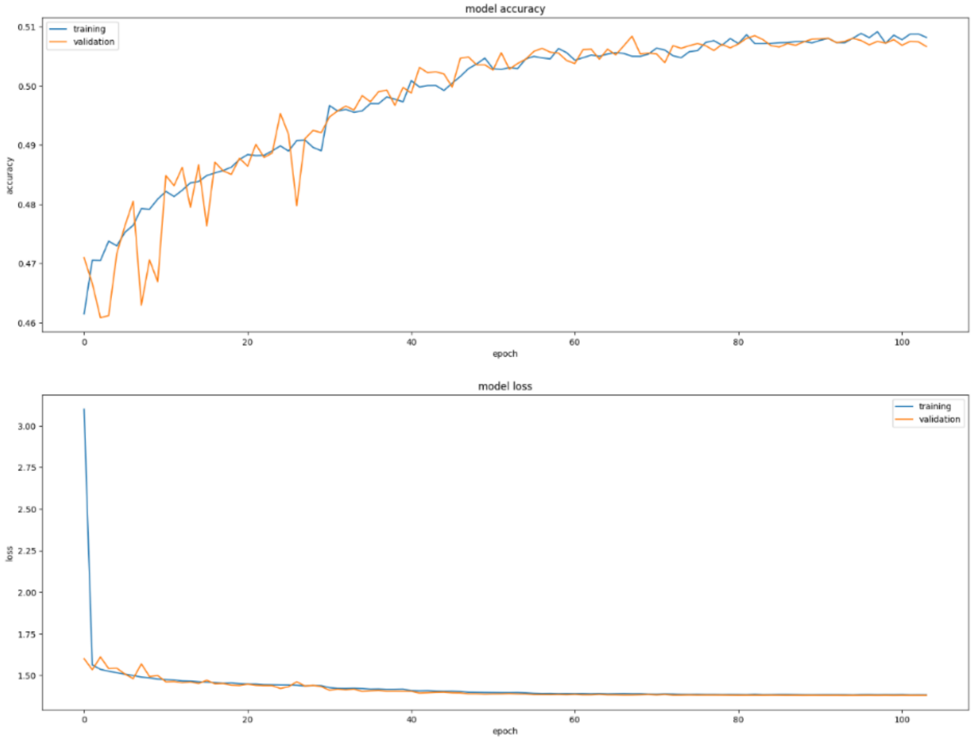


Figure 10



Experiment # 5

We reverted to experiment 1 parameters but used a LSTM model instead of GRU. This produced a similar pattern to experiment one with high training accuracy and low training validation loss, but the third-best testing accuracy. It also has a high validation loss with 1.4353, the highest of the group. The training accuracy was the highest of the experiments with .6962 but testing accuracy was slightly worse than experiment one with .5322. The minutes it took to run the model was 17.4.

Experiment #6

Since the previous experiment had the highest training accuracy yet, we tried to implement more regularization to see if this would help the overfitting and bump the testing accuracy up above what experiment one had produced. We added dropout to just the first layer and it was unsuccessful at improving the accuracy. Training accuracy went down to .6044 and training loss went up to 1.1001. Testing accuracy increased from experiment 5 to .5246 but still technically below experiment 1. The model did, however, have the best validation loss of 1.2988. Additionally, we applied dropout to just the first layer and the model run time increased to 27.8 but was not as long as experiments 3 and 4.

Conclusion

Deciding what models work best will be determined by the original goal of the research; however, for this study we are evaluating our models based on accuracy. Although better than guessing, these models performed mediocrely, and further tweaking and experimenting is needed to produce higher performing models. This could look like using more data, different contributing variables, different models, and testing out different parts of the architecture like layers, units, regularization, epochs, and batch size. In addition, the original clusters could be formed using other techniques. Models 1 and 5 showed signs of promise with training accuracy ~68 and 69%, but overfitting was a major issue.

Since we are seeking the model with the best testing accuracy, model one is the suggested model to use, with the highest testing accuracy of .5396 and the second lowest testing loss of 1.2755. It was built with seven GRU layers (2 dense layers) with units 512/256/128/64/32/64, activation ReLU for all GRU layers, Softmax for the output layer, epoch size of 200, and batch size of 64. It also used regularization methods early stopping, batch normalization, and L2 on the output layer. Enhancements and regularization techniques should help address the overfitting issue; however, further research and more experiments are recommended to help develop more successful models.

Experiment Performance Summary Table

	Training Data		Validation Data		Testing Data		Processing Time
Parameters	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	in Minutes
Model: GRU Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 Regularization: Early Stopping, Batch Normalization, L2 in dense layer Features: All	0.6825	0.8646	0.5285	1.4294	0.5396	1.2755	13.3
Model: GRU Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 Regularization: Early Stopping, Batch Normalization, L2 in dense layer **Features: Removed Day, Weekend, Month**	0.6177	1.0241	0.5241	1.3458	0.5187	1.2814	12.0
Model: GRU Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 **Regularization: Early Stopping, Dropout, L2 in dense layer** Features: All	0.5604	1.2256	0.5260	1.3290	0.5239	1.3331	42.7
Model: GRU Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 **Regularization: Early Stopping, Batch Normalization, L2 in all layers** Features: All	0.5082	1.3832	0.5066	1.3804	0.5019	1.3892	68.1
Model: LSTM Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 Regularization: Early Stopping, Batch Normalization, L2 in dense layer Features: All	0.6962	0.8298	0.5266	1.4353	0.5322	1.2767	17.365
Model: LSTM Layers: 7 (5 layers and 2 dense layers) Layer units: 512/256/128/64/32/64 Activation: ReLU, Softmax Optimizer: Adam Epochs: 200 Batch size: 64 **Regularization: Early Stopping, Batch Normalization, L2 in dense layer, Dropout 0.1 in first layer** Features: All	0.6044	1.1001	0.5263	1.2988	0.5346	1.2726	27.845

References

Golliver, B. (2022). Lakers' LeBron James wants to own NBA expansion team in Las Vegas. *The Washington Post (Washington, D.C. 1974. Online)*.

Luka Doncic is poised for richest NBA contract ever in 2025. (2024, April 1). BasketNews.com. Retrieved May 30, 2024, from https://basketnews.com/news-204096-luka-doncic-poised-for-richest-nba-contract-ever-in-2025.html.

"Major pro Sports Teams Ranked by Market Size." *Sports Media Watch*, 26 Oct. 2018, www.sportsmediawatch.com/nba-market-size-nfl-mlb-nhl-nielsen-ratings/.

Pastorello, G. (2023, August 24). *Predicting NBA Salaries with Machine Learning | Gabriel Pastorello*. Towards Data Science. Retrieved May 30, 2024, from https://towardsdatascience.com/predicting-nba-salaries-with-machine-learning-ed68b6f75566

Curcic, Dimitrije. "NBA Salaries Analysis (1991-2022). https://runrepeat.com/salary-analysis-in-the-nba-1991-2019

Sporttrac.com

Wu, William, et al. "Classification of NBA Salaries through Player Statistics." *Sports Analytics Group at Berkeley*, chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://sportsanalytics.studentorg.berkeley.edu/projects/nba-salaries-stats.pdf.