Akshay Madhusudhan
Jayden Perez
Mannan Shukla

NBA Trading Card Price Predictor

Github

Problem Statement:

Our aim with this project is to predict what players in the NBA will have a breakout year and increase the value of their respective sports cards. The price of sports cards fluctuates on a few factors. Those being skill, popularity of the player and what type of card it is. The cards themselves range from basic players to having an autograph and a piece of their jersey. We believe that analyzing players' stats and role on the team will allow us to estimate the price of the card and if it will continue to increase in value or if it is time to sell the card for what it is worth.

Connection to class:

In summary our project relates to class in many ways. The way we used data scraping to gather our data, the libraries we used to create models also relate. In detail, we first scraped the data from html tables on a website, and then cleaned up the data using methods we learned in class on the homeworks. Things such as removing null rows, renaming columns to fit our needs better and removing columns we did not need. We used this to make our data frames which we stored in csv files.Throughout the project we applied fundamentals of Numpy, pandas, and dataframe manipulation which have been shown many times throughout the class. Once we had our data we had to begin pre-processing. Lastly, we used linear regression, Random Forest Regressor, and K-means to compare what different models looked like and to see if any of them could accurately predict the price of NBA cards given a player has a breakout year (An increase of stats compared to the year before).

Novelty and Importance:

While some price predictions rely purely on historical sales data and market trends, having the players current statistics and popularity in mind will allow for a fresh experience and maybe allow for a better prediction. It would also be interesting to see if injury, MVP race, or a playoff performance have an effect on the price of the card which most people would not even consider into the price of a NBA card. The importance of this would be for collectors, investors, sports betting fanatics, and fantasy sports players. The collectors and investors are already speculating potential players and if their card would be worth keeping, this model would allow for an accurate estimate of the card's potential worth. For sports betting, looking at cards of higher value and seeing the reason why could potentially sway their bets.

Importance to us:

All of us have a high interest in sports and particularly the NBA. Seeing players get better is an amazing thing, however if you collect sports cards it can be lucrative to have a good rookie card of someone who only gets better throughout their career. While we may not collect NBA cards, many people do and creating a model able to predict which card would be worth more after an upcoming season would be useful to them.

Our plan and what we accomplished:

Data:

Using data scraping to pull information off of a website for price of cards along with

player stats and projected stats for the following season. For cleaning the data we will want to deal with values based on their formatting. Websites like Basketball-Reference and PriceCharting contain a great deal of information regarding NBA trading cards and player statistics. For data storage we just saved it to a csv file and shared it via github as it was easiest and for any changes to the datasets we would update the csv everytime through the web scraping scripts we implemented. The particular sites we used were:
https://www.sportscardspro.com/console/basketball-cards-2024-panini-nba-hoops
https://fantasy.espn.com/basketball/players/projections

Data Pre-Processing:
First we made a copy of all datasets used for manipulation to ensure data preservation/integrity.

NBA Trading Card Dataset

For this dataset we began by converting the price columns (Ungraded, PSA 9, PSA 10) into valid floats and if there were any unexpected characters, replace the price value with np.nan. We then extracted the player names from the card descriptions using regex patterns to take into account card variants and other descriptors. These came in useful as we also used data extracted from card description to create more descriptor columns in a way that can be used by the model. For example, we added columns like Print_Run, Is_RC, and Is_Serial_Numbered (populated by 1s for true and 0s for false) which became key features in model prediction. We then filtered the card description for the most common variants and if no other variants were extracted through our function, then the card was set to “Base” for the variant column. After all this processing, any rows that contained null values in the Player_Extracted column were dropped as an extra precautionary measure for model training.

ESPN Stats Dataset:

First we cleaned the player names and extracted them to a different Players_Cleaned column through regex as they often had team positions and/or player season status right after the player names. We needed this column primarily to merge the stats columns with the player_extracted column in the trading card dataset. We then ensured all the values in the stats columns were converted to numeric types for model training. We then left joined the stats data on the trading card data to preserve all prices and card information.

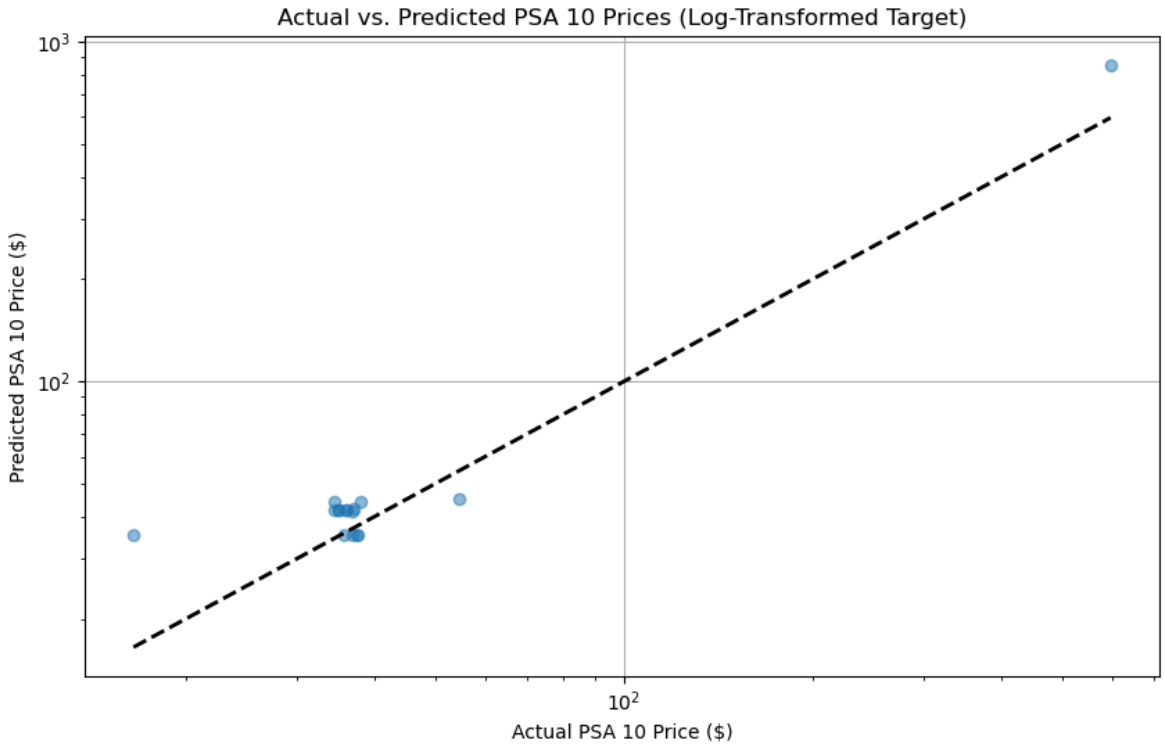
Merged Data Frame:

For this new merged data frame we had no use for two player name columns, so player_extracted was kept while player_cleaned was dropped. We then set a TARGET_PRICE global variable to allow for customizability in terms of model training. You can set this variable to PSA 10 for predicting PSA 10 prices, or PSA 9 for PSA 9 prices, or Ungraded for ungraded prices. Using one-hot encoding, we created new variant binary columns to allow for the model to be trained based on whether the card was a variant or not. To make sure all the values in the stats

columns were usable, we filled all null values with 0 and this came in useful as there were player cards with no stats since many were rookies whose projections hadn't been recorded yet. We then separated the data into features to train on and target column and used log transformation on our target column because card prices tend to have an extreme variance considering there are many low-priced cards and a few rare high-priced cards.

The train and test data were split into 40% test data and 60% training data.
We came to this decision based on previous 20/80 and 30/70 splits which performed worse when fitting and predicting with the Random Forest Regressor model (through comparison of r^2 score).

Below is a graph representing the accuracy of the Random Forest Regressor model on **PSA 10** pricing:



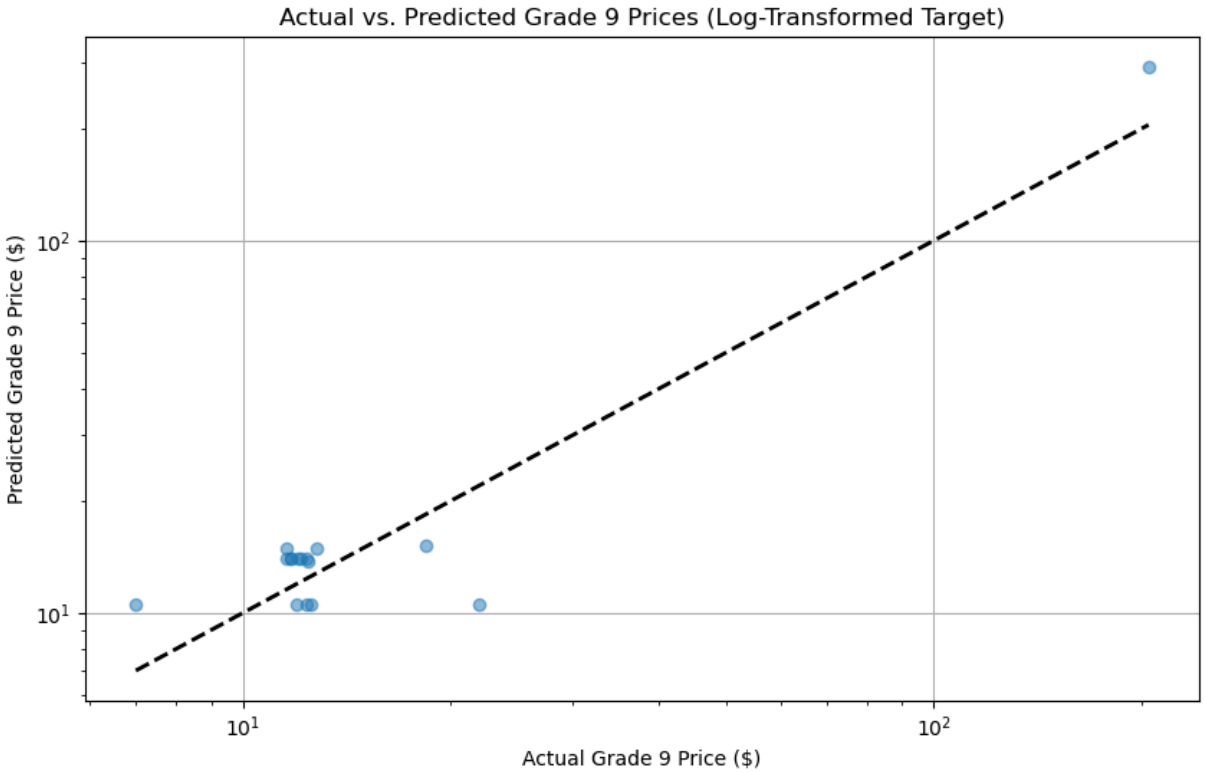
Model Evaluation...
Target Variable: PSA 10
Mean Absolute Error (MAE): $21.73
Root Mean Squared Error (RMSE): $64.24
R-squared (R^2): 0.7767

	feature	importance
0	Print_Run	0.863208
1	Is_Serial_Numbered	0.049907
2	Is_RC	0.028707
3	STL	0.020284
4	3PM	0.015429
5	AST	0.010361
6	REB	0.009952
7	BLK	0.000613
8	FT%	0.000499
9	FG%	0.000473
10	MIN	0.000402
11	PTS	0.000165
12	Var_Base	0.000000

Sample Predictions:

	Card	Actual_PSA 10	Predicted_PSA 10	MIN	FG%	FT%	3PM	REB	AST	STL	BLK	PTS
24	Ja Morant #288	34.99	41.764701	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
20	Zion Williamson #290	36.15	41.764701	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
19	Matas Buzelis #241\n[RC]	35.77	35.464796	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
32	Julius Erving #300	35.86	41.764701	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
5	Luka Doncic #281	34.57	44.329436	37.0	0.486	0.772	3.7	9.1	9.3	1.4	0.5	33.1
16	Reed Sheppard #291\n[RC]	36.87	35.464796	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
44	Kyrie Irving [Premium Gold Prizm] #108\n/10	597.09	852.400004	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
34	Donovan Clingan #293	34.99	41.764701	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
47	Luka Doncic #115	38.09	44.329436	37.0	0.486	0.772	3.7	9.1	9.3	1.4	0.5	33.1
7	Dalton Knecht #247\n[RC]	16.50	35.460490	21.6	0.422	0.821	1.6	3.8	1.6	1.0	0.5	11.3
31	Zach Edey #239\n[RC]	37.57	35.464796	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
13	Shai Gilgeous-Alexander #283	37.05	42.271698	34.8	0.532	0.875	1.4	5.6	6.8	2.0	0.9	32.0
17	Matas Buzelis #295\n[RC]	37.33	35.464796	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
38	Anthony Edwards #286	34.57	41.764701	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
41	Victor Wembanyama [Silver Prizm Premium] #165	54.52	44.892770	33.4	0.491	0.820	2.1	12.0	4.4	1.8	4.3	25.4
10	LeBron James #18	36.93	41.572758	35.2	0.533	0.752	2.1	7.4	8.0	1.2	0.6	26.0

PSA 9 Predictions:



Model Evaluation...
Target Variable: Grade 9
Mean Absolute Error (MAE): $8.22
Root Mean Squared Error (RMSE): $22.36
R-squared (R^2): 0.7689

Top Feature Importances:

	feature	importance
0	Print_Run	0.738800
1	Is_Serial_Numbered	0.168806
2	Is_RC	0.036069
3	REB	0.019792

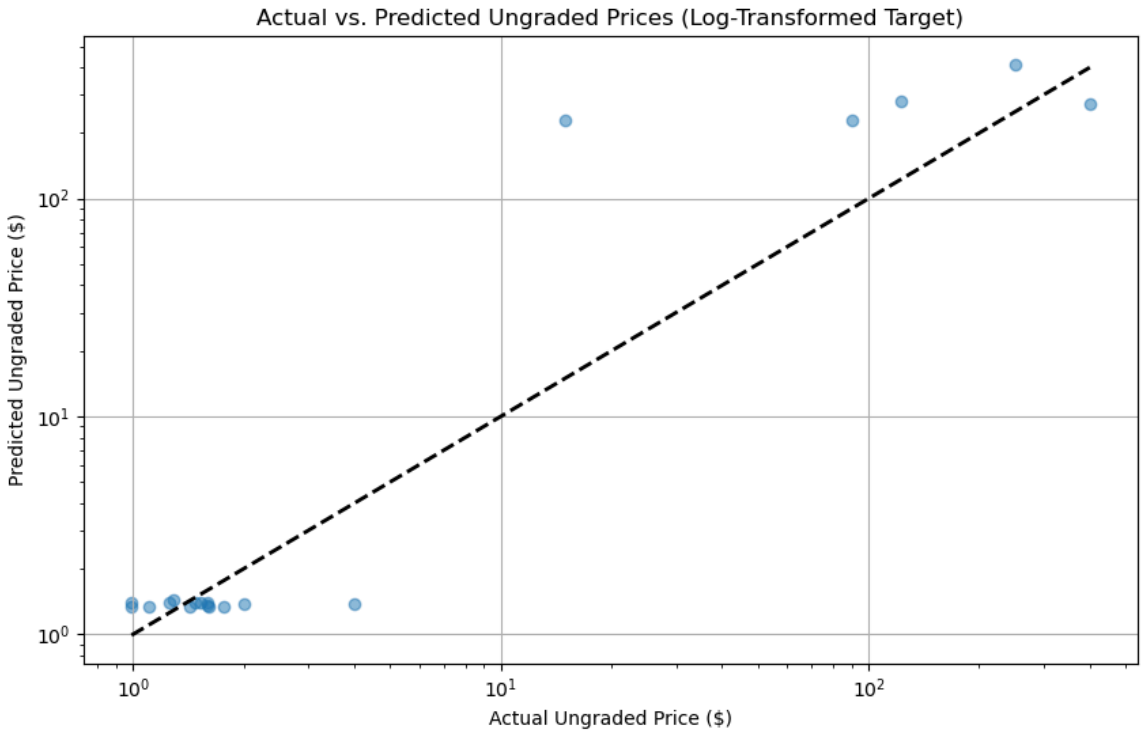
4	3PM	0.014146
5	STL	0.010321
6	AST	0.010280
7	BLK	0.000536
8	MIN	0.000401
9	FT%	0.000307
10	FG%	0.000307
11	PTS	0.000089
12	Var_Base	0.000000

Sample Predictions:

	Card	Actual_Grade 9	Predicted_Grade 9	MIN	FG%	FT%	3PM	REB	AST	STL	BLK	PTS
24	Ja Morant #288	11.71	14.059006	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
20	Zion Williamson #290	12.11	14.059006	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
19	Matas Buzelis #241\n[RC]	11.98	10.523574	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
32	Julius Erving #300	12.01	14.059006	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
5	Luka Doncic #281	11.57	14.959812	37.0	0.486	0.772	3.7	9.1	9.3	1.4	0.5	33.1
16	Reed Sheppard #291\n[RC]	12.36	10.523574	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
44	Kyrie Irving [Premium Gold Prizm] #108\n/10	204.57	292.841970	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
34	Donovan Clingan #293	11.71	14.059006	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
47	Luka Doncic #115	12.77	14.959812	37.0	0.486	0.772	3.7	9.1	9.3	1.4	0.5	33.1
7	Dalton Knecht #247\n[RC]	7.00	10.523574	21.6	0.422	0.821	1.6	3.8	1.6	1.0	0.5	11.3
31	Zach Edey #239\n[RC]	22.00	10.523574	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
13	Shai Gilgeous-Alexander #283	12.42	13.770653	34.8	0.532	0.875	1.4	5.6	6.8	2.0	0.9	32.0
17	Matas Buzelis #295\n[RC]	12.52	10.523574	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
38	Anthony Edwards #286	11.57	14.059006	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
41	Victor Wembanyama [Silver Prizm Premium] #165	18.42	15.156062	33.4	0.491	0.820	2.1	12.0	4.4	1.8	4.3	25.4
10	LeBron James #18	12.38	13.996551	35.2	0.533	0.752	2.1	7.4	8.0	1.2	0.6	26.0

Linear Regression Results:
MAE: $21.86
RMSE: $77.27
R^2: -1.7594

Ungraded Predictions:



Model Evaluation...
Target Variable: Ungraded
Mean Absolute Error (MAE): $42.34
Root Mean Squared Error (RMSE): $83.41
R-squared (R^2): 0.3497

Top Feature Importances:

	feature	importance
0	Is_Serial_Numbered	0.505439
1	Print_Run	0.483825
2	Is_RC	0.005446
3	PTS	0.004873
4	MIN	0.000100
5	STL	0.000086
6	AST	0.000078
7	BLK	0.000049
8	3PM	0.000046
9	REB	0.000026
10	FG%	0.000022
11	FT%	0.000010
12	Var_Base	0.000000

Sample Predictions:

	Card	Actual_Ungraded	Predicted_Ungraded	MIN	FG%	FT%	3PM	REB	AST	STL	BLK	PTS
27	Stephon Castle [Silver] #234\n/199	15.00	230.096548	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
40	Jaylen Wells #269\n[RC]	0.99	1.393153	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
26	Nikola Jokic #284	1.42	1.334392	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
44	Kyrie Irving [Premium Gold Prizm] #108\n/10	122.75	279.292794	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
24	Ja Morant #288	0.99	1.334392	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
37	Victor Wembanyama [Lazer Blue Prizm Premium] #...	90.00	227.463232	33.4	0.491	0.820	2.1	12.0	4.4	1.8	4.3	25.4
12	Austin Reaves [Gold Vinyl Prizm Premium] #138\n/1	250.00	412.740114	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
19	Matas Buzelis #241\n[RC]	1.25	1.393153	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
4	Victor Wembanyama #165	2.00	1.375382	33.4	0.491	0.820	2.1	12.0	4.4	1.8	4.3	25.4
25	Chet Holmgren #289	1.77	1.334392	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
8	Dalton Knecht [Gold Artist Proof] #294\n/10	400.00	273.368687	21.6	0.422	0.821	1.6	3.8	1.6	1.0	0.5	11.3
3	Dalton Knecht #294\n[RC]	1.29	1.443468	21.6	0.422	0.821	1.6	3.8	1.6	1.0	0.5	11.3
6	Larry Bird #299	1.60	1.334392	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
41	Victor Wembanyama [Silver Prizm Premium] #165	3.99	1.375382	33.4	0.491	0.820	2.1	12.0	4.4	1.8	4.3	25.4
34	Donovan Clingan #293	1.10	1.334392	0.0	0.000	0.000	0.0	0.0	0.0	0.0	0.0	0.0
13	Shai Gilgeous-Alexander #283	1.59	1.358022	34.8	0.532	0.875	1.4	5.6	6.8	2.0	0.9	32.0

As we can see from the sample predictions and test data in general, ungraded predictions seemed to have a worse performance than Grade 9 and PSA 10 specifically because of how much variance there was in the data. A lot of ungraded cards can go for around 1 dollar, while a select few go for hundreds of dollars. Card variants in tandem with player popularity and performance for the specific print run of that card played a huge role in ungraded pricing.

Linear Regression Results:
MAE: $32.78
RMSE: $93.08
R^2: 0.1902

Results:

PSA10

Random Forest Regressor: The results of PSA 10 were the best of any we did with a MAE of $21.73 it means on average our prediction was off by $21.73 which due to there being many cards worth a lot of money in the sample was pretty promising to see. THe RMSE of $64.24 supports this by suggesting the larger errors were much larger than the cheaper cards. This was likely caused by the few cards in the $1000s of dollars skewing the result. The R^2 of .7767 suggests that the approximately 77.67% of variance in PSA10 card prices is explained by the model.

Linear Regression: As we can tell through the r^2 score, the linear regression method seemed to be performing way worse than just predicting through the mean, which meant it was not a good method for modeling this data. Notably, we would be better guessing the price of the card rather than using this model.

PSA9

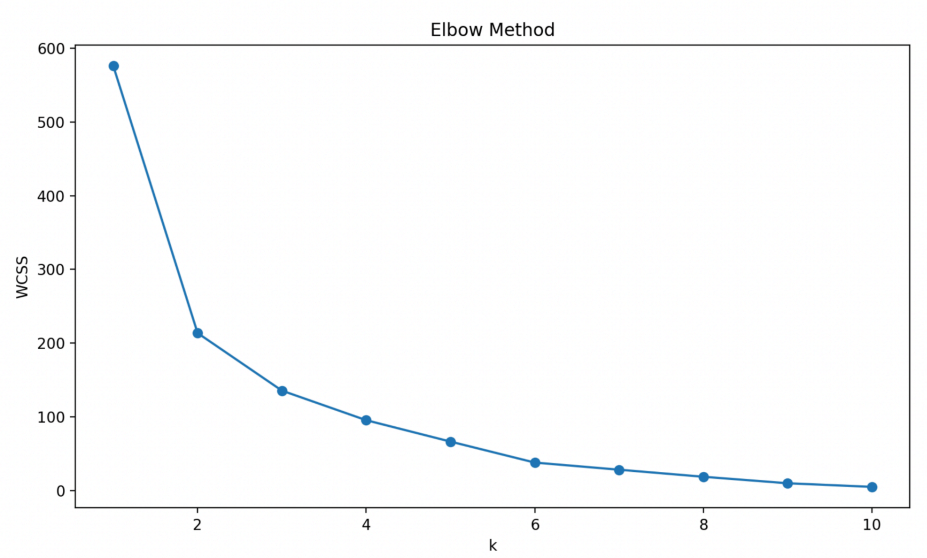
Random Forest Regressor: The results of PSA9 was the 2nd best model with a MAE of $8.22 it means on average our prediction was off by $8.22 meaning we have a very strong accuracy for PSA9 cards. The RMSE of $22.36 is also good showing there may be some error in the very high value cards. The R^2 of .7689 suggests that the approximately 76.89% of variance in PSA9 card prices is explained by the model.

Linear Regression: Same as PSA10 has a horrible R^2 being -1.75 meaning it has no good method for modeling the data and therefore should be ignored.

Ungraded

Random Forest Regressor: The results for Ungraded were much worse in comparison to PSA9 and PSA10 with an MAE of $42.34. This shows that the model was of on average $42.34 and for cards that tend to be very cheap this is not a good prediction also shown by the R^2 of .3497.

Linear Regression: Surprisingly this was the best linear regression type model we had with an MAE of $32.78 and R^2 of .1902.



We also tested K-Means clustering as a potential modeling method, to determine the appropriate number of clusters for our k-means analysis, we plotted the within-cluster sum of squares (WCSS) for values of k from one through ten and looked for the point at which adding additional clusters yielded diminishing returns. The resulting elbow plot showed a sharp decline in WCSS as k increased from one to three, indicating that each new cluster up to k = 3 captured substantial additional structure in the data. Beyond three clusters the curve began to flatten, with only marginal decreases in WCSS. This inflection point at k = 3 suggested that three clusters strike the best balance between model simplicity and explanatory power, so we selected k = 3 for our final analysis.

Cluster Centers

MIN	FG%	FT%	3PM	REB	AST
1.6	0.0312593	0.0608148	0.118519	0.281481	0.118519
1.77636e-15	-2.77556e-17	2.77556e-17	1.11022e-16	0	2.22045e-16
33.5091	0.487727	0.815	2.66364	8.8	6.10909

Cluster Centers

Var_Artist Proof	Var_Base	Var_Gold Artist Proof	Var_Gold Vinyl Prizm Premium	Var_Lazer Blue Prizm Premium
-1.38778e-17	0.962963	-6.93889e-18	-2.08167e-17	6.93889e-18
0.3	-1.11022e-16	0.1	0.2	3.46945e-18
2.08167e-17	0.545455	0.0909091	6.93889e-18	0.0909091

Cluster Centers

STL	BLK	PTS	Is_RC	Print_Run	Is_Serial
0.0740741	0.037037	0.837037	0.555556	5000	0
0	-5.55112e-17	0	0	28.3	1
1.49091	1.91818	26.9455	0	3189.45	0.363636

Cluster Centers

Var_Nebula Prizm Premium	Var_Premium	Var_Premium Gold Prizm	Var_Red Explosion	Var_Silver /199	Var_Silver Prizm Premium
6.93889e-18	0.037037	-1.38778e-17	6.93889e-18	6.93889e-18	6.93889e-18
0.1	-3.46945e-18	0.2	0	0.1	0
3.46945e-18	-3.46945e-18	0.0909091	0.0909091	-3.46945e-18	0.0909091

Cluster Summary

Cluster	count	avg_price	avg_pts	avg_reb	avg_printrun	pct_rookie
0	27	35.3933	0.837037	0.281481	5000	0.555556
1	10	1044.94	0	0	28.3	0
2	11	243.853	26.9455	8.8	3189.45	0

Silhouette score for k=3: 0.443

The three rows in our Cluster Centers table each represent the average, un-scaled feature values for one of the three clusters. In other words, for Cluster 0 the average card sees only about 1.6 minutes of playing time (MIN), essentially no measurable field-goal percentage contribution (FG%), and virtually no premium variant flags which entails the base/rookie mass‑market group. Cluster 1, by contrast, has zero in all performance stats (because these cards belonged to players without recorded projections in the merged dataset with a mix of rookies) yet an average print run of only about 28 and an average PSA-10 price over $1,000; this entails the scarce premium variant group. Finally, Cluster 2 is composed of veterans and breakout stars, averaging nearly 27 points and 8.8 rebounds per game, trading at an average PSA-10 price of about $244 on a print run of roughly 3,200 copies. Taken together, these results tell us that the cards most likely to increase in value are those in Cluster 1—extremely limited prints of premium variants—and to a lesser extent the high-performance, uncommon cards of veterans in Cluster 2. Our silhouette score essentially means that this clustering method's performance was neither particularly good nor poor. Definitely better than linear regression, but worse than random forest regressor.