

INTELIGÊNCIA ARTIFICIAL – A3

*Análise, tratamento de dados e
algoritmos de Machine Learning*

Daniel Ikeda Kuniyoshi, 125111347030

Diego Fernandes Martinez, 12522193520

Nayane Pereira Mazaro, 125111365317

Pedro Shiraishi de Almeida, 125111350990

Rafael Henrique Gonçalves Soares, 125111374176

Vinicius Alves Vieira, 125111350019

ESCOLHA DA BASE

- Base de dados Carros e suas características e valor sugerido pela fabricante (MSRP)
 - *A Base de dados abrange inicialmente cerca de 12 mil itens*
 - *Entre os itens temos o modelo, marca, tipo de transmissão, categoria de mercado preço e etc.*

Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels
BMW	1 Series M	2011	premium unleaded (required)	335	6	MANUAL	rear wheel drive
BMW	1 Series	2011	premium unleaded (required)	300	6	MANUAL	rear wheel drive
BMW	1 Series	2011	premium unleaded (required)	300	6	MANUAL	rear wheel drive
BMW	1 Series	2011	premium unleaded (required)	230	6	MANUAL	rear wheel drive
BMW	1 Series	2011	premium unleaded (required)	230	6	MANUAL	rear wheel drive
BMW	1 Series	2012	premium unleaded (required)	230	6	MANUAL	rear wheel drive
BMW	1 Series	2012	premium unleaded (required)	300	6	MANUAL	rear wheel drive
BMW	1 Series	2012	premium unleaded (required)	300	6	MANUAL	rear wheel drive
BMW	1 Series	2012	premium unleaded (required)	230	6	MANUAL	rear wheel drive
BMW	1 Series	2013	premium unleaded (required)	230	6	MANUAL	rear wheel drive
BMW	1 Series	2013	premium unleaded (required)	300	6	MANUAL	rear wheel drive
BMW	1 Series	2013	premium unleaded (required)	230	6	MANUAL	rear wheel drive

OBJETIVO DO PROJETO

- Escolher uma base de dados ampla (base atual tem 12 mil itens) e fazer um estudo sobre algoritmos de ML.
- Algoritmos de Machine Learning Escolhidos
 - *Random Forest*
 - *Regressão por KNN*
- A base de dados escolhida já tinha um código previamente feito com Random Forest, porém nossa ideia era aprofundar o estudo em cima desse código previamente feito.
- O Random Forest previamente feito utilizando a base toda. Dividimos a base de dados em 2as partes e estudar separadamente.
 - *Carros populares*
 - *Carros que na categoria de mercado tem Luxo, Performance e Alta performance*

TRATAMENTO DA BASE DE DADOS


- A base de dados original tem aproximadamente 12 mil valores
- Realizamos a limpeza da base excluindo valores duplicados.
- Excluimos ou alteramos os valores nulos ou desconhecidos (“Unknow”), pois esses valores poderiam interferir nos resultados da análise e do modelo.
- Com essa limpeza inicial começamos a dividir a base de dados em carros populares e carros de luxo/performance/alta-performance.

```
[611] #Separando carros que de acordo com a categoria de mercado contenham Luxo, Performance e Alta performance  
cars_data_popular = cars_data[~cars_data['market'].str.contains("Luxury|Performance|High-Performance")]  
cars_data_luxury = cars_data[cars_data['market'].str.contains("Luxury|Performance|High-Performance")]
```

OUTLIERS

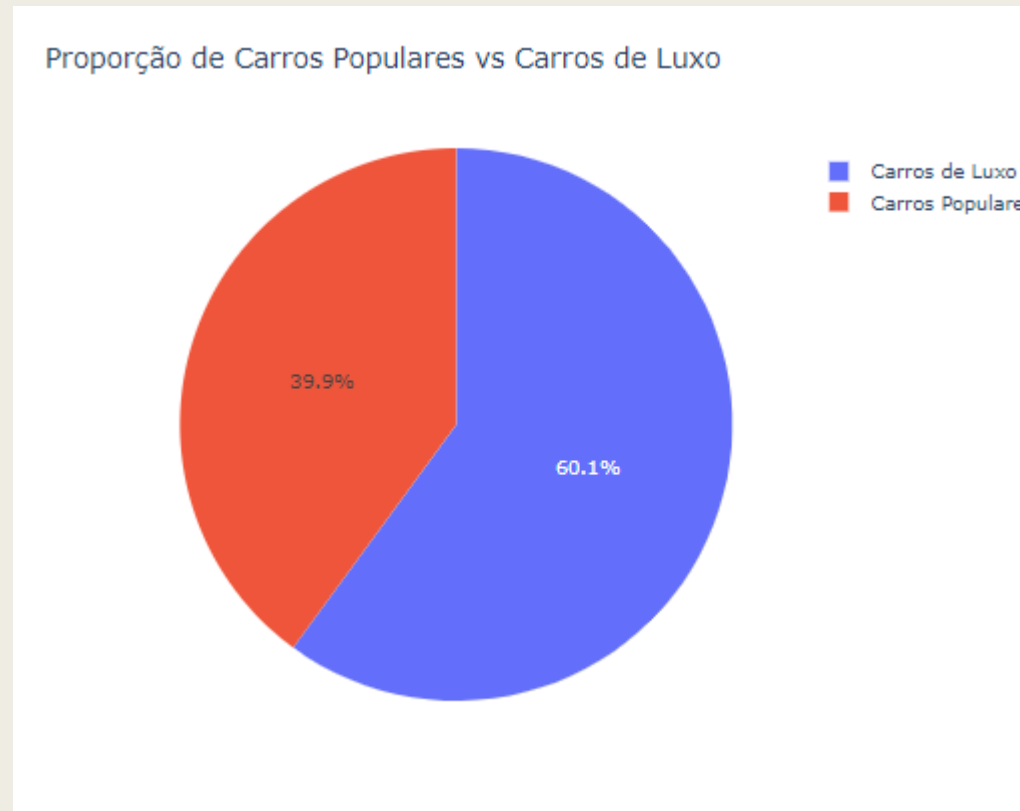
- Com a base de dados finalmente dividida, podemos fazer um tratamento adequado de outliers
 - *Um Outlier é um valor “muito fora” do padrão, ou seja, um valor que de alguma forma é inconsistente com os demais.*
- Ao verificar os outliers, realizamos a remoção deles.

```
[617] #Removendo Outliers dos carros populares
s1 = cars_data_popular.shape
clean = cars_data_popular[['hp', 'cylinders', 'highway_mpg', 'city_mpg', 'price']]
for i in clean.columns:
    qt1 = cars_data_popular[i].quantile(0.25)
    qt3 = cars_data_popular[i].quantile(0.75)
    iqr = qt3 - qt1
    lower = qt1 - (1.5 * iqr)
    upper = qt3 + (1.5 * iqr)
    min_in = cars_data_popular[cars_data_popular[i] < lower][i].index
    max_in = cars_data_popular[cars_data_popular[i] > upper][i].index
    cars_data_popular.drop(min_in, inplace=True)
    cars_data_popular.drop(max_in, inplace=True)
s2 = cars_data_popular.shape
outliers = s1[0] - s2[0]
print("Deleted outliers are: ", outliers)
```

 Deleted outliers are: 325

DISTRIBUIÇÃO DA BASE

- Por curiosidade decidimos verificar a distribuição da base após toda a limpeza.
 - Após a limpeza a base caiu de 12 mil valores para 7.5mil
 - A distribuição ficou em cerca de ~40% em carros populares e 60% em carros de luxo/performance/alta-performance.

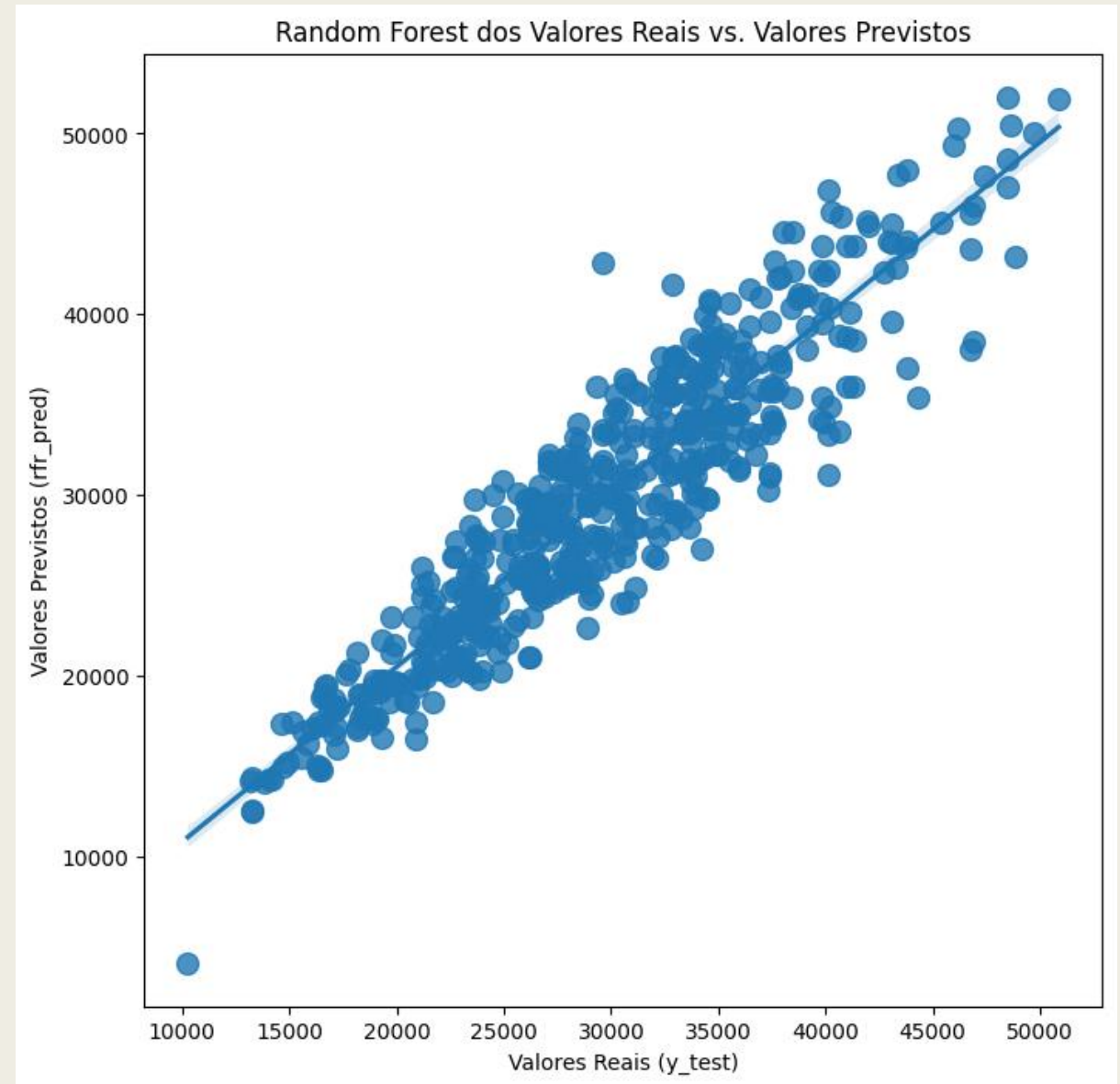


MODELOS- Random Forest (rfr)

- O Random forest é basicamente um modelo que busca um dado aleatório de uma base inicial e através dele cria uma árvore, e repete o processo, construindo árvores de dados.
- Quando se trata de regressão ele pega um novo dado e faz uma média com relação às previsões das árvores criadas, essa média é o resultado da regressão.
- Quando se trata de classificação, a cada novo dado, a árvore faz uma previsão da classe do dado (a que possível árvore ele faz parte), a classe mais “votada” entre as arvorés é o resultado final da classificação.
- O Random Forest consegue lidar bem com uma alta gama de dados, mas ao mesmo tempo pode ter um custo computacional maior (mais memória e poder computacional) devido ao seu formado de criar múltiplas “árvores”, impactando também no tempo de previsão (também devido ao número de dados/árvores”

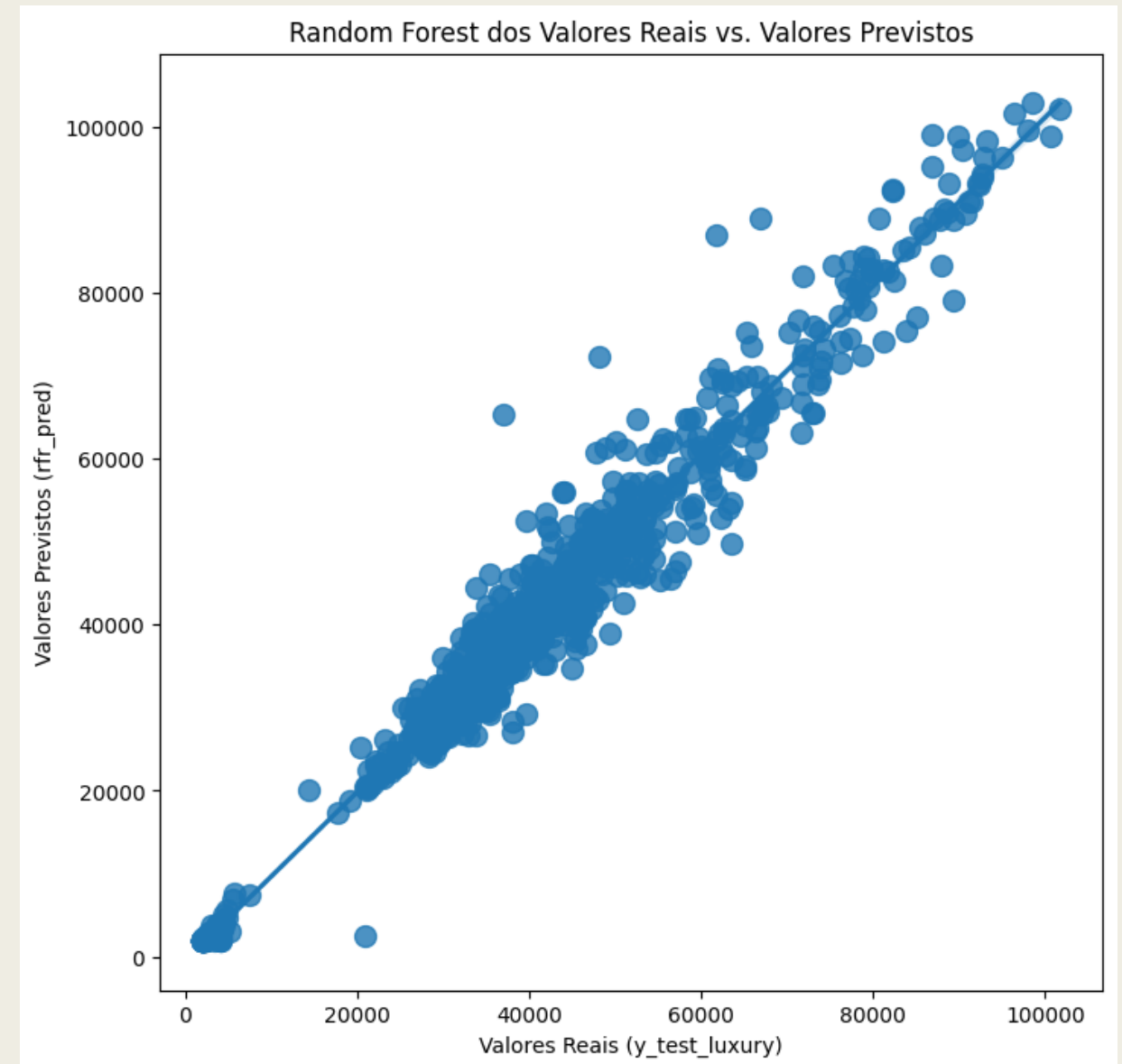
Random Forest – Carros Populares

- A densidade do algoritmo dos carros populares performou com uma precisão (R2 Score) de 0.85, o que significa que teve uma capacidade de explicar 85% da variabilidade de dados da resposta de acordo com os dados de entrada.
 - *É como falar sobre a “Precisão” do algoritmo em relação a entrada e saída*
- A linha representa a tendência linear das previsões do modelos, ou seja, valores reais x valores previstos. Quanto mais próximos os pontos estão das linhas, mais performático é o resultado do algoritmo



Random Forest – Luxo/Performance

- A densidade do algoritmo dos carros de luxo performou com uma precisão (R2 Score) de 0.96, o que significa que teve uma capacidade de explicar 96% da variabilidade de dados da resposta de acordo com os dados de entrada.
 - *Ele performou melhor que os carros populares, provavelmente porque teve uma quantidade de dados maior que os valores*
- Observa-se que a densidade dos pontos ao redor da linha de tendência é maior que os populares, o que evidencia um R2 maior.

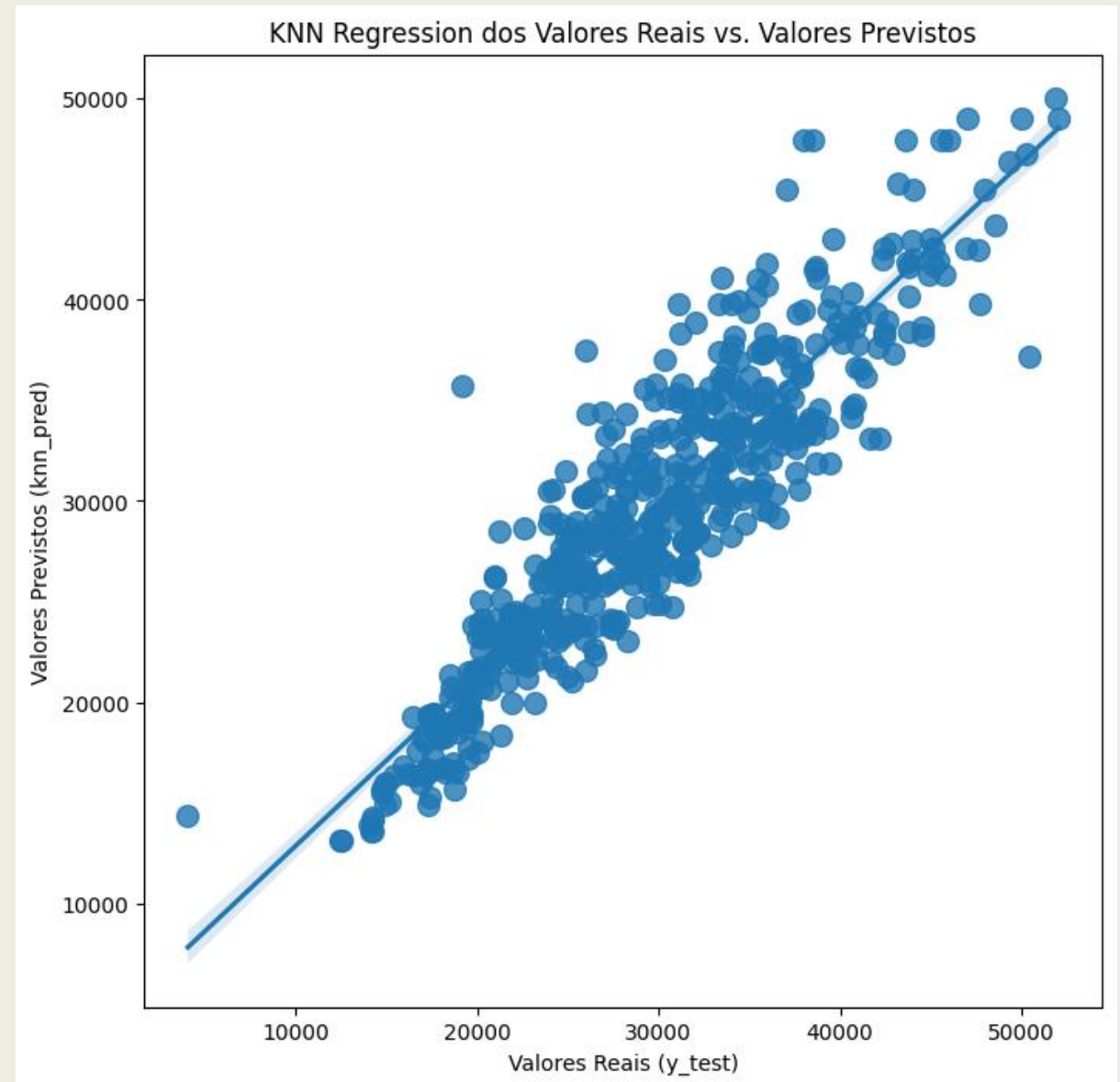


MODELOS- Regressão por KNN

- O KNN é um algoritmo simples, ele faz uso da estratégia de verificação de “Vizinhos próximos”, por isso seu nome K-Nearest Neighbors (KNN).
- O seu funcionamento é simples, na predição e classificação, quando um valor novo é adicionado ele verifica a distância dos seus vizinhos mais próximos e classifica de acordo com eles, quando se trata de regressão ele utiliza a média dos seus vizinhos.
- A KNN também é treinada com uma base de dados já definida, com uma entrada e uma saída determinada, ou seja, um algoritmo supervisionado

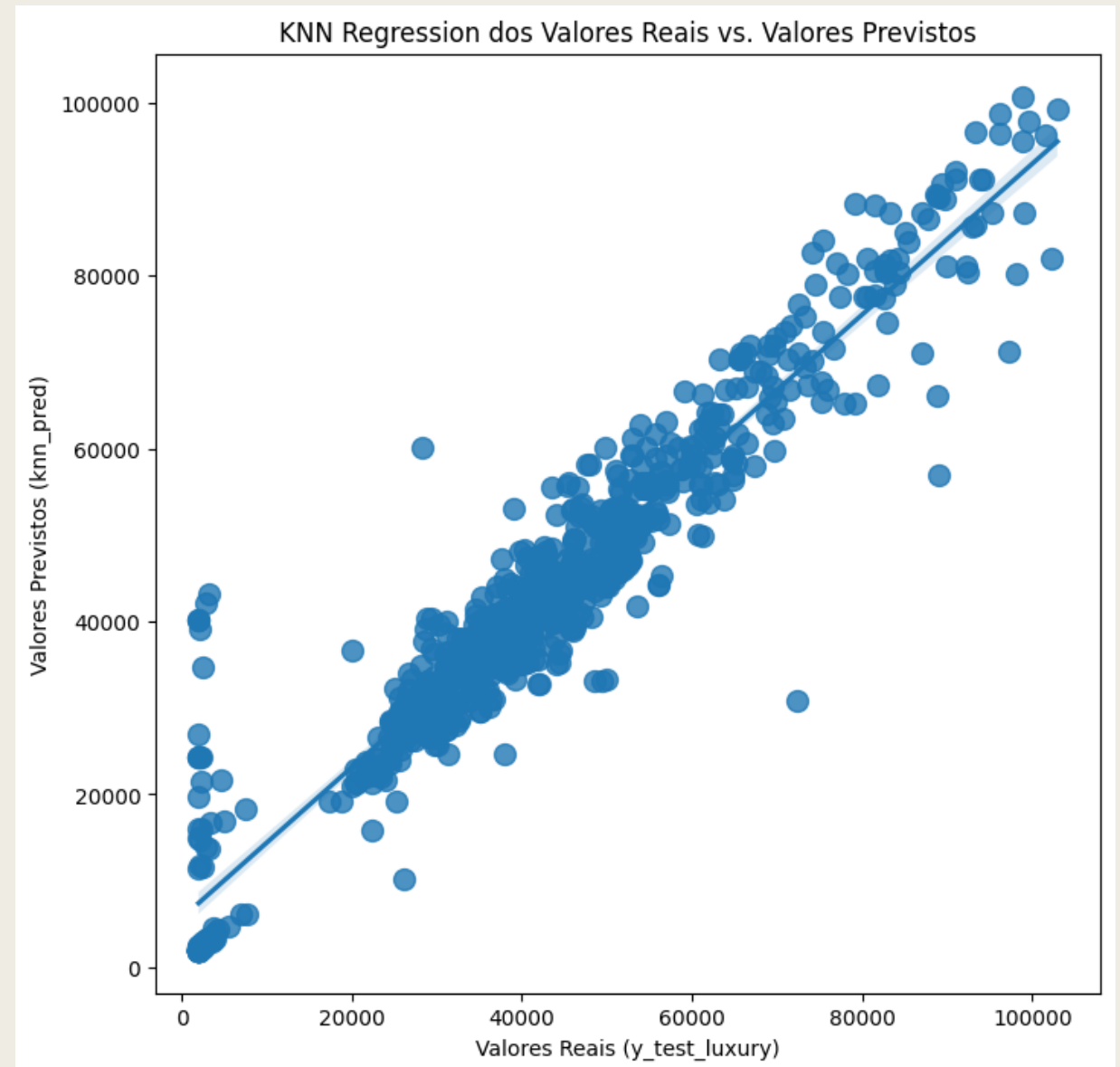
KNN – Carros Populares

- A densidade do algoritmo com carros populares performou com uma precisão (R2 Score) de 0.82, o que significa que teve uma capacidade de explicar 82% da variabilidade de dados da resposta de acordo com os dados de entrada.
 - *Ele performou melhor que os carros populares, provavelmente porque teve uma quantidade de dados maior que os valores*



KNN – Luxo/Performance

- A densidade do algoritmo dos carros de luxo performou com uma precisão (R2 Score) de 0.90, o que significa que teve uma capacidade de explicar 90% da variabilidade de dados da resposta de acordo com os dados de entrada.
 - *Ele performou melhor que os carros populares, provavelmente porque teve uma quantidade de dados maior que os valores*
- Observa-se que a densidade dos pontos ao redor da linha de tendência é maior que os populares, o que evidencia um R2 maior.



CONCLUSÕES

- Observamos que se compararmos os algoritmos Random Forest e KNN com suas respectivas bases (populares e de luxo/performance), o random forest tem um score R^2 melhor que o KNN, isso se deve ao fato que o algoritmo de rfr lida melhor com uma maior densidade de dados.
- Foi observado também em testes que quanto maior o número de vizinhos no KNN, menor é o score, provavelmente pela característica de que o KNN usa de referência os valores mais próximos e não lida tão bem com uma grande densidade de valores
- Mesmo com a base de carros de luxo tendo uma densidade de dados quase o dobro dos carros populares.
- Através dos histogramas também percebemos que a faixa de preço de carros populares é até 50k enquanto os de luxo sua maioria se concentra em 20k e 65k e ultrapassando esses valores