Técnicas de análise de regressão para precificação de imóveis na cidade de São Paulo

Duílio Campos Sasdelli

Disciplina Aprendizado de Máquina
Professor Adriano Veloso, PhD.

Universidade Federal de Minas Gerais
Belo Horizonte - 2021

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Introdução

- Mercado imobiliário em 2021:
  - Já equivale a **3.5% do PIB**;
  - Volume financiado passa de **R$ 97 B**;
  - Déficit habitacional de quase **6 milhões** de moradias no país;
- Necessidade da precificação correta dos imóveis:
  - Predizer **quanto tempo** levaria para vender um imóvel anunciado;
  - Quantificar o **potencial de construção** em determinada região;
  - Identificar **anúncios com preços abaixo** do mercado;
- O presente trabalho tem como objetivo avaliar diferentes técnicas de regressão sobre dados de imóveis;

Financiamento imobiliário dispara, bate recorde e projeta alta de 34% para o ano

Mercado imobiliário dispara, bate recorde e projeta alta de 34% para o ano

Introdução


QUINTOANDAR
Quanto vale o meu imóvel?
Descubra o valor ideal para vender o seu imóvel
1 de 12
Endereço do imóvel *
Digite um local
Próximo


Descubra o preço de venda do seu imóvel
Preencha o formulário e avalie online seu imóvel
Como funciona? A ferramenta de avaliação de imóveis utiliza técnicas de estatística e inteligência artificial que analisam os dados de toda a base de imóveis anunciados pelo Portal Casa Mineira para estimar os valores do seu imóvel para venda e aluguel.
Dados do imóvel
Endereço do imóvel
Digite seu endereço
Digite logradouro e número do imóvel
Tipo
Selecione
Ano de construção
Selecione
Finalidade
Selecione

Trabalhos relacionados

- Em [1] o autor utiliza diferentes regressores para predição do preço de imóveis na cidade de **Beijing**, tais como XGBoost, Regressão Linear, Florestas Aleatórias, Ridge e Lasso, etc. Assim como o presente trabalho, o **XGBoost** apresentou os melhores resultados;
- Em [2], é feito um trabalho que analisa 12.223.582 anúncios de imóvei em todo o **Brasil** e a precificação é realizada por meio da **combinação** das técnicas de **Florestas Aleatórias** para atributos numéricos e **Redes Neurais Recorrentes** para imagens e texto. A combinação das duas técnicas trouxe bons resultados, sendo **vencedora** de uma competição do *Kaggle*;
- Em [3] e [4] os autores utilizam apenas regressores lineares (do tipo Ridge e regressão linear padrão) para precificação de imóveis nas cidades de **Fortaleza** e **Sorocaba** respectivamente. Os resultados em ambos os casos foram satisfatórios, com R^2 acima de **0.8**;

Dados

- Coletados por meio da API do **zapimoveis** em formato **JSON**;
- Optou-se por escolher apenas **apartamentos** localizados na cidade de **São Paulo;**
- Dados brutos totalizaram 631.622 anúncios;
    - Mais de 80% não possui dados de latitude, longitude e idade do imóvel;
- Foram extraídos 232 atributos, sendo:
    - 12 atributos textuais (descrição, tipo de imóvel, imobiliária, bairro, cidade, etc);
    - 15 atributos numéricos (preço de venda, número de suítes, latitude, etc);
    - 4 atributos de temporais (data de construção do imóvel, data do anúncio, etc);
    - 201 atributos com características do imóvel (varanda, piscina, etc);

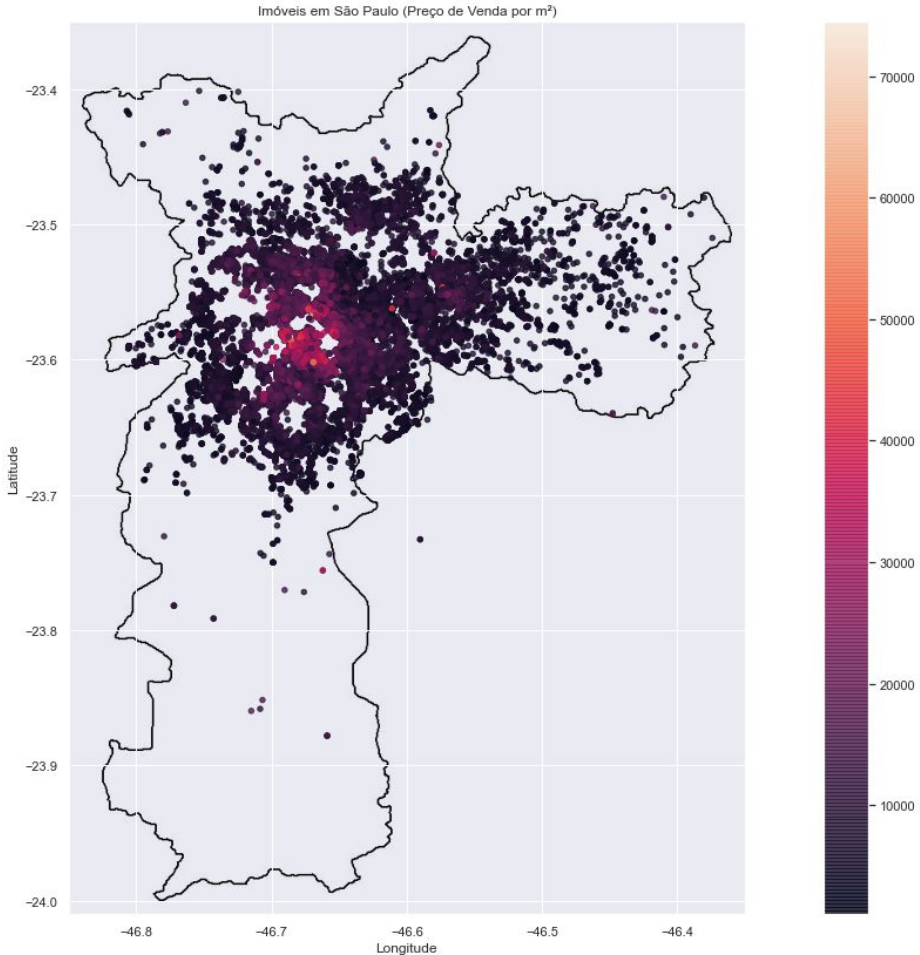
Pré-processamento

- Remoção de instâncias:
  - Sem campos “Latitude”, “Longitude”, “Idade”;
  - Com valores incorretos (erros de digitação) para determinados campos por meio do *z-score* ou valor absoluto;
- Dados categóricos:
  - Seleção dos atributos mais comuns (que existam em pelo menos 5% das instâncias):
  - *One-hot encoding*;
- Dados numéricos:
  - Substituição dos dados faltantes pela média da base de treino nos campos;
- Normalização/padronização dos dados:
  - MinMaxScaler e StandardScaler a depender da técnica utilizada;
- Geração de novos atributos polinomiais para os métodos lineares;
- Redução de dimensionalidade via PCA para o método de vizinhos mais próximos (kNN);

$$z = \frac{x - \mu}{\sigma}$$

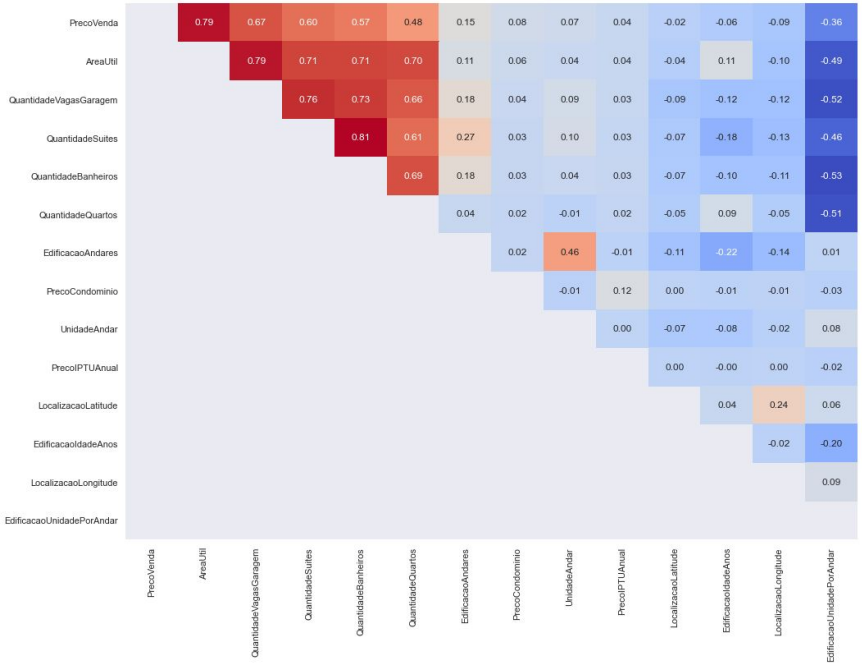
Análise descritiva

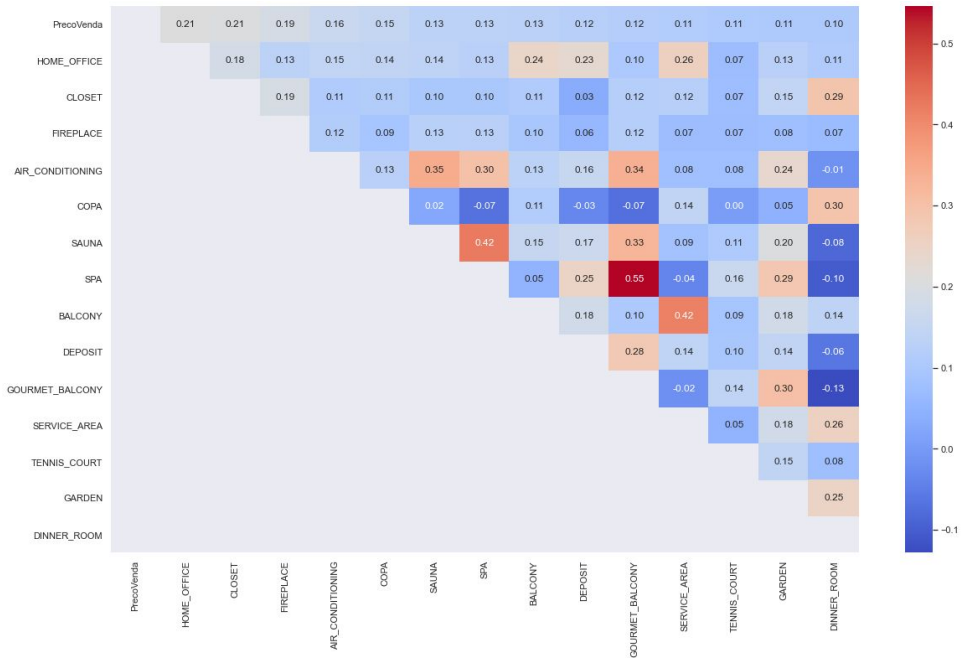
- Após a remoção de instâncias via pré-processamento:
  - 55104 instâncias;
  - 59 atributos:
    - 10 atributos numéricos: Área, Latitude, Longitude, Suites, Quartos, Banheiros, Vagas, Número de andares, Unidades por andar, Idade em anos;
    - 49 atributos *booleanos* (0 ou 1): Piscina, Salão de Festas, Playground, Academia, Churrasqueira, Elevador, etc;
  - Criação do atributo preço de venda por área para a análise descritiva;



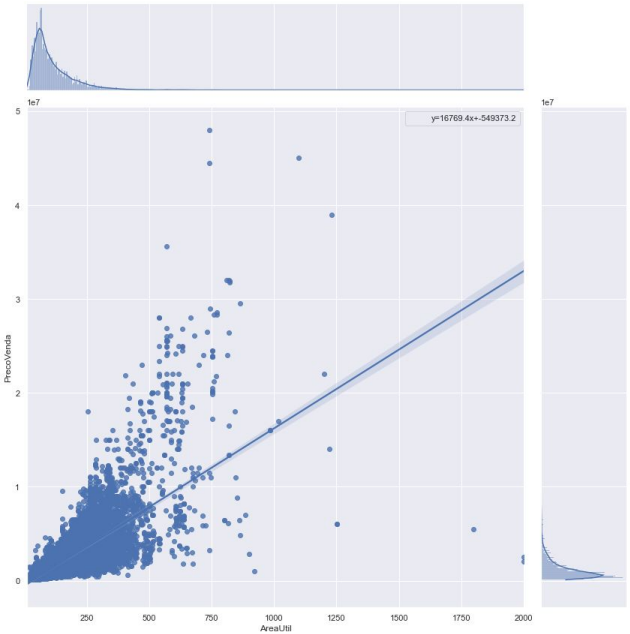
Análise descritiva

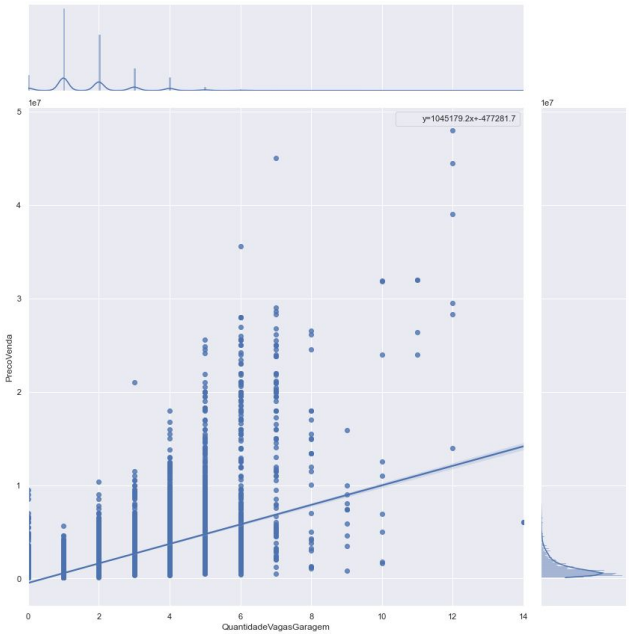
$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X) \cdot \operatorname{var}(Y)}}$$

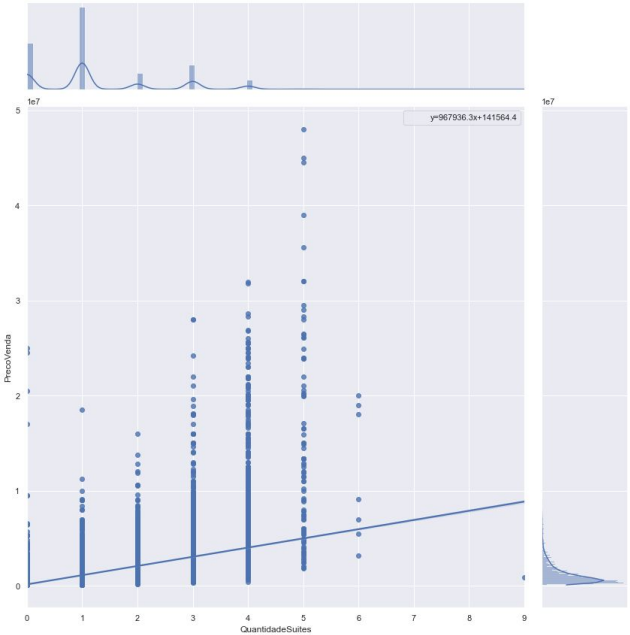




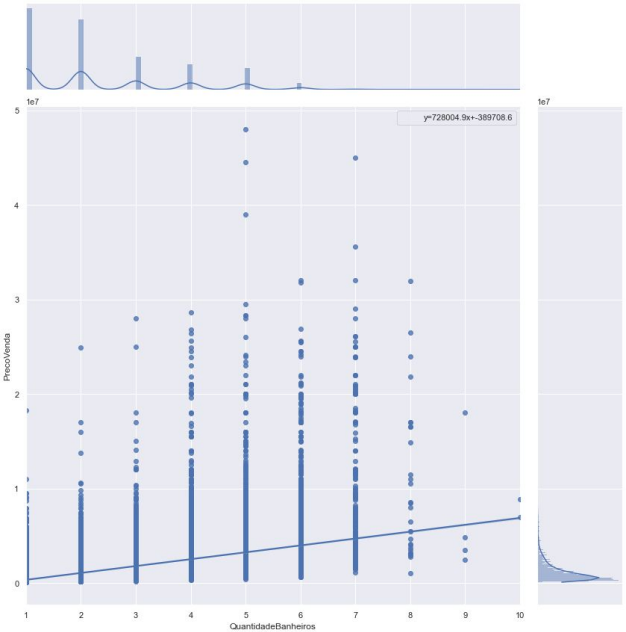
Análise descritiva

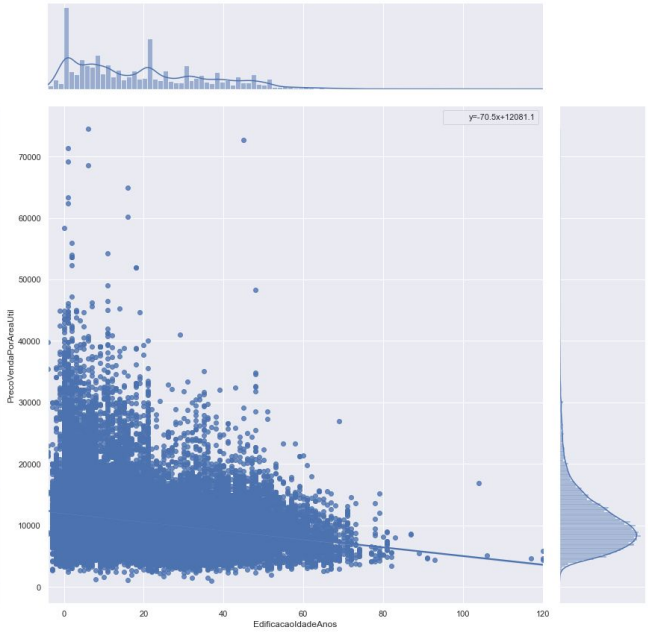


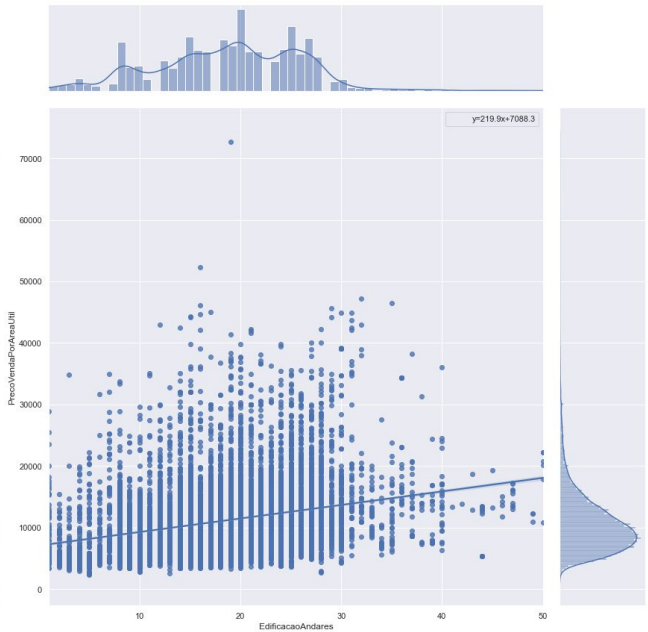




Análise descritiva







Regressores utilizados

- Foram testados 5 diferentes regressores, 4 disponíveis na biblioteca *scikit-learn* e 1 da *xboost*;
  - Linear: **LinearRegression;**
  - Vizinhos mais próximos: **KNeighborsRegressor;**
  - Árvore de decisão: **DecisionTreeRegressor;**
  - Ensemble: **RandomForestRegressor;**
  - Ensemble - Boosting: **XBoostRegressor;**
- Também foram testados outros regressores, os quais foram descartados devido ao baixo desempenho:
  - Máquina de vetor de suporte: **SVR**;
  - Redes neurais: **MLPRegressor;**

Métricas de avaliação

- Coeficiente de determinação R^2:
  - Mede o grau de variação da variável dependente pelas variáveis independentes;
  - Valores geralmente entre 0 e 1;
  - Possui limitações:
    - Tende a **aumentar** com o aumento de parâmetros;
    - **Não indica** se o **regressor correto** foi utilizado;

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

- Tempo de treinamento:
  - Avaliar o desempenho dos regressores;
- Outras métricas foram consideradas, tais como a própria SS_{RES}, R^2 ajustado, mas foram descartadas;
- Validação cruzada usando 5 *folds* (20% para treino e 80% para teste);
- Uso de busca em grade (*GridSearchCV*) para escolha de parâmetros;

Regressão Linear

- Utilizado como *baseline*;
- Foram utilizados dois regressores:
  - Um com as características originais com viés (60 parâmetros);
  - Outro com características transformadas polinomialmente com viés (grau 2) (1830 parâmetros);
- Possui limitações:
  - Ainda que se crie novas características, as mesmas são limitadas a combinações lineares entre si;
  - Não é capaz de identificar bem como as variáveis independentes se relacionam entre si para predizer a variável dependente;
  - É sensível a *outliers* por se basear na minimização da soma de quadrados residuais;

Regressão Linear - Resultados

Resultados para validação cruzada					
Par	R^2 médio de teste		R^2 médio de treino		Tempo médio de treino
	Média	Var	Média	Var	
poli=1	0.676327	0.009123	0.677829	0.002231	**0.109237**
poli=2	**0.791624**	0.012143	0.829615	0.002453	109.878032

Coeficientes para Regressão Linear (dados completos) com polinômio = 1	
Coeficiente	Parâmetro
3.423118e+07	AreaUtil
-2.952655e+06	QuantidadeQuartos
2.657833e+06	QuantidadeVagasGaragem
-1.856407e+06	EdificacaoIdadeAnos
-1.069231e+06	EdificacaoUnidadePorAndar
5.711066e+05	EdificacaoAndares
3.267713e+05	LocalizacaoLatitude
-2.824112e+05	QuantidadeBanheiros
2.401055e+05	ELECTRIC_GENERATOR
2.118210e+05	SPA

Coeficientes para Regressão Linear (dados completos) com polinômio = 2	
Coeficiente	Parâmetro
7.951645e+07	AreaUtil LocalizacaoLatitude
7.375151e+07	AreaUtil QuantidadeVagasGaragem
6.686354e+07	AreaUtil EdificacaoAndares
-5.045498e+07	AreaUtil EdificacaoIdadeAnos
-3.199369e+07	AreaUtil^2
-3.195397e+07	AreaUtil EdificacaoUnidadePorAndar
-2.494221e+07	AreaUtil
-2.196161e+07	AreaUtil LocalizacaoLongitude
-2.034322e+07	AreaUtil DISABLED_ACCESS
1.901264e+07	AreaUtil QuantidadeSuites

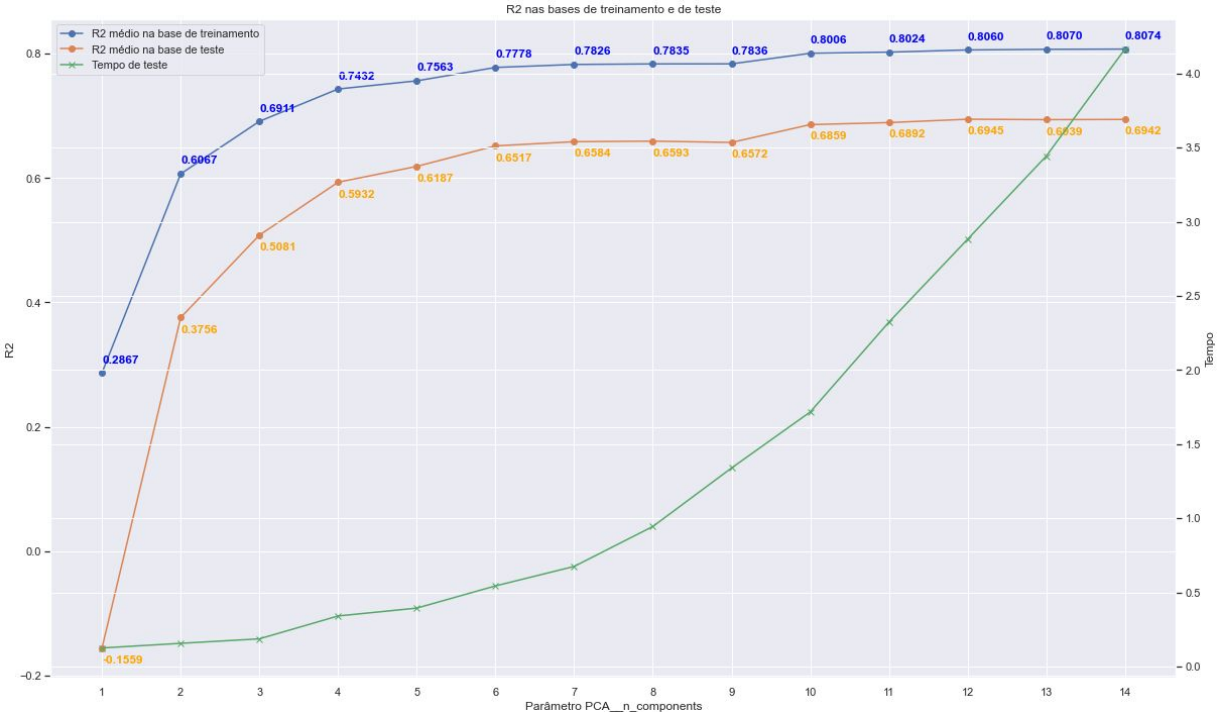
Vizinhos mais próximos (kNN)

- Regressor baseado em vizinhos mais próximos (kNN);
- Técnica bastante **sensível à quantidade e distribuição** de características;
  - Necessário realizar a redução de dimensionalidade com Análise de Componentes Principais (PCA);
  - Também necessário **padronização** (*StandardScaler*) por meio do *z-score*;
- Parâmetros avaliados:
  - **pca_n_components:** número de componentes principais utilizados (1 a 14);
  - **n_neighbors:** número de vizinhos a ser considerado para cálculo da regressão (1 a 14);
- Trata-se de um regressor não-linear muito simples e não recomendado para dados com muitas dimensões;

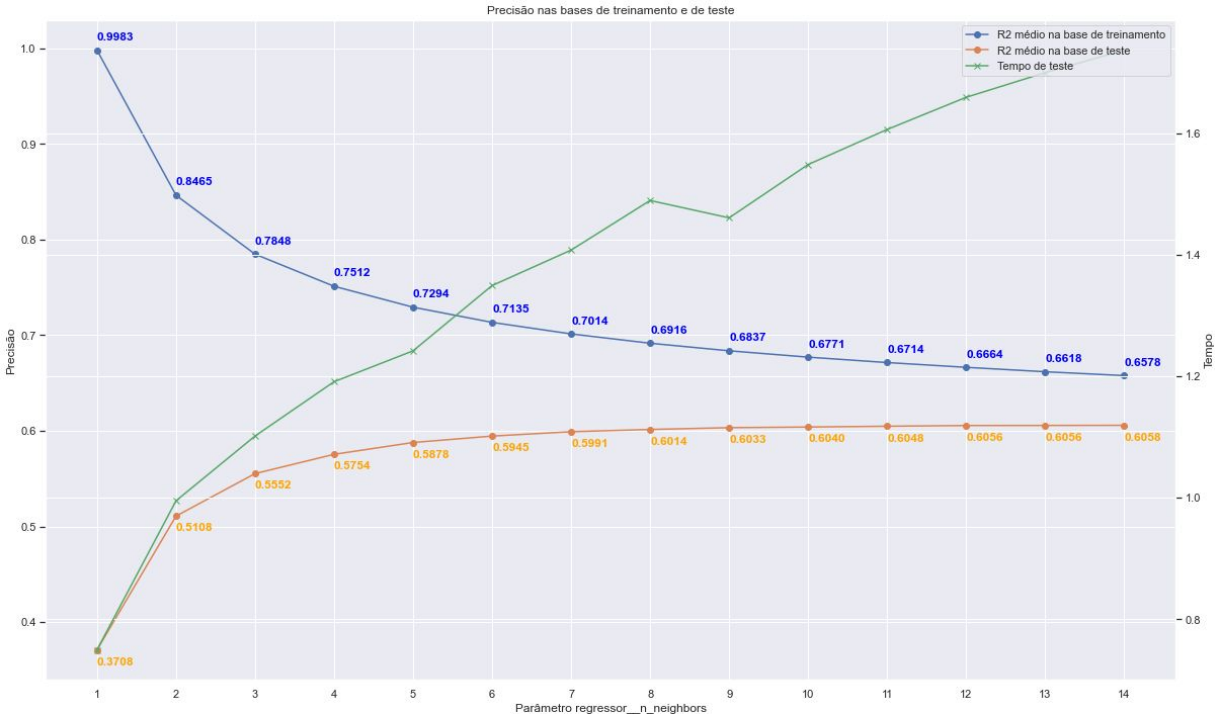
Vizinhos mais próximos (kNN) - Resultados

Melhores resultados para validação cruzada					
Parâmetros	**R^2 médio de teste**		**R^2 médio de treino**		**Tempo médio de treino**
	Média	**Var**	**Média**	**Var**	
PCA_n_components=14 n_neighbors=7	**0.712835**	0.008641	**0.788452**	0.002627	4.303055
PCA_n_components=12 n_neighbors=9	0.712813	0.007082	0.771283	0.003331	**3.103700**
PCA_n_components=13 n_neighbors=7	0.712693	0.007731	0.787499	0.003437	3.469988
PCA_n_components=13 n_neighbors=8	0.712290	0.007275	0.779714	0.003249	3.550433
PCA_n_components=14 n_neighbors=8	0.712287	0.006848	0.778828	0.002487	4.510920

Vizinhos mais próximos (kNN) - Resultados



Vizinhos mais próximos (kNN) - Resultados



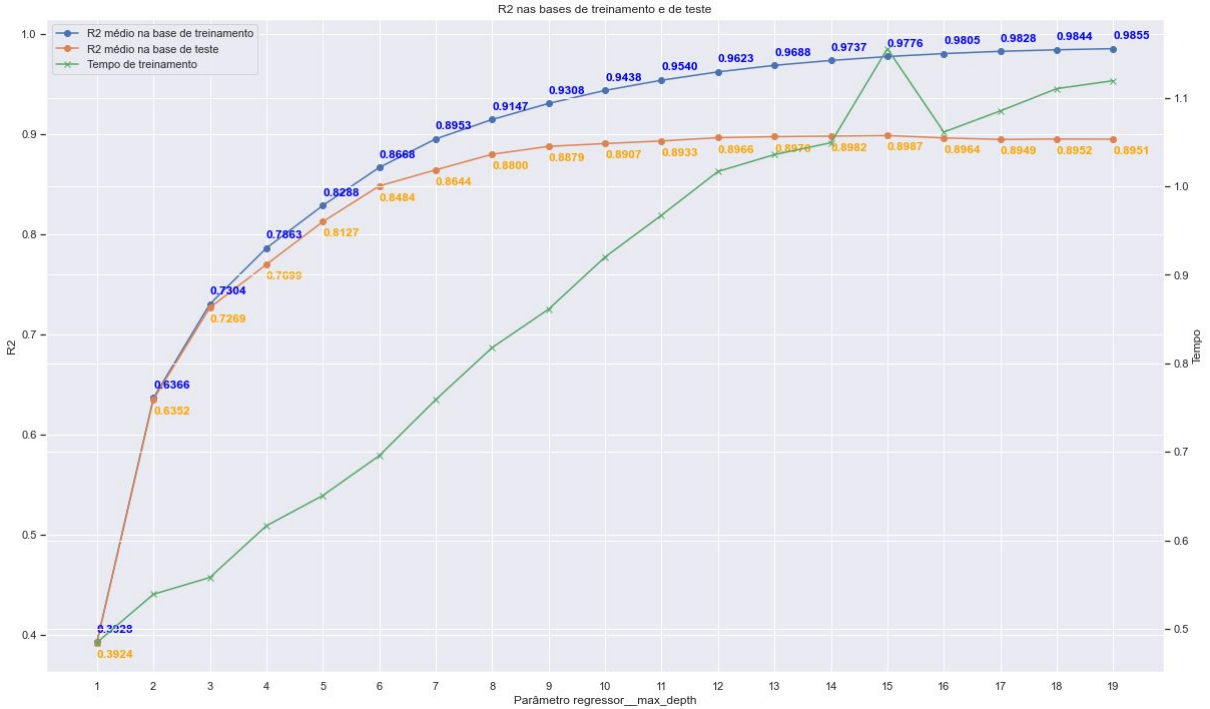
Árvore de Decisão

- Regressor baseado em **árvores de decisão**, conforme visto em sala de aula;
- Parâmetros avaliados:
  - **max_depth:** tamanho máximo da árvore (1 a 19);
  - **min_samples_split:** número mínimo de amostras para dividir um nó interno (2 a 14);
- Trata-se de um **regressor não-linear**, ou seja, capaz de identificar **relacionados não lineares** entre as variáveis independentes;
- É sujeito a sobreajuste se os parâmetros forem utilizados incorretamente;

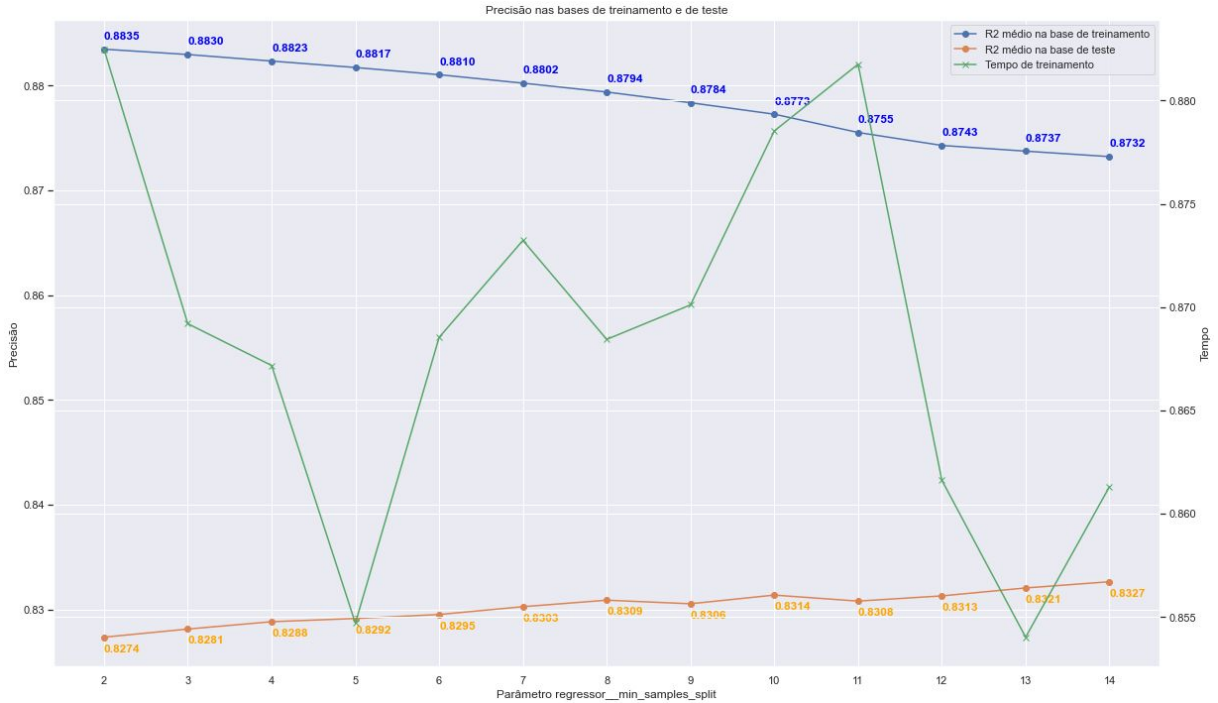
Árvore de Decisão - Resultados

Melhores resultados para validação cruzada					
Parâmetros	R^2 médio de teste		R^2 médio de treino		Tempo médio de treino
	Média	Var	Média	Var	
max_depth = 14 min_samples_split = 14	**0.903530**	0.012969	0.964727	0.002290	**1.023064**
max_depth = 15 min_samples_split = 14	0.903239	0.013318	0.968029	0.002220	1.068107
max_depth = 15 min_samples_split = 13	0.902250	0.013779	0.968990	0.002296	1.050258
max_depth = 14 min_samples_split = 10	0.901667	0.014598	0.971173	0.001771	1.037601
max_depth = 15 min_samples_split = 10	0.901463	0.014864	**0.974790**	0.001533	1.312572

Árvore de Decisão - Resultados



Árvore de Decisão - Resultados



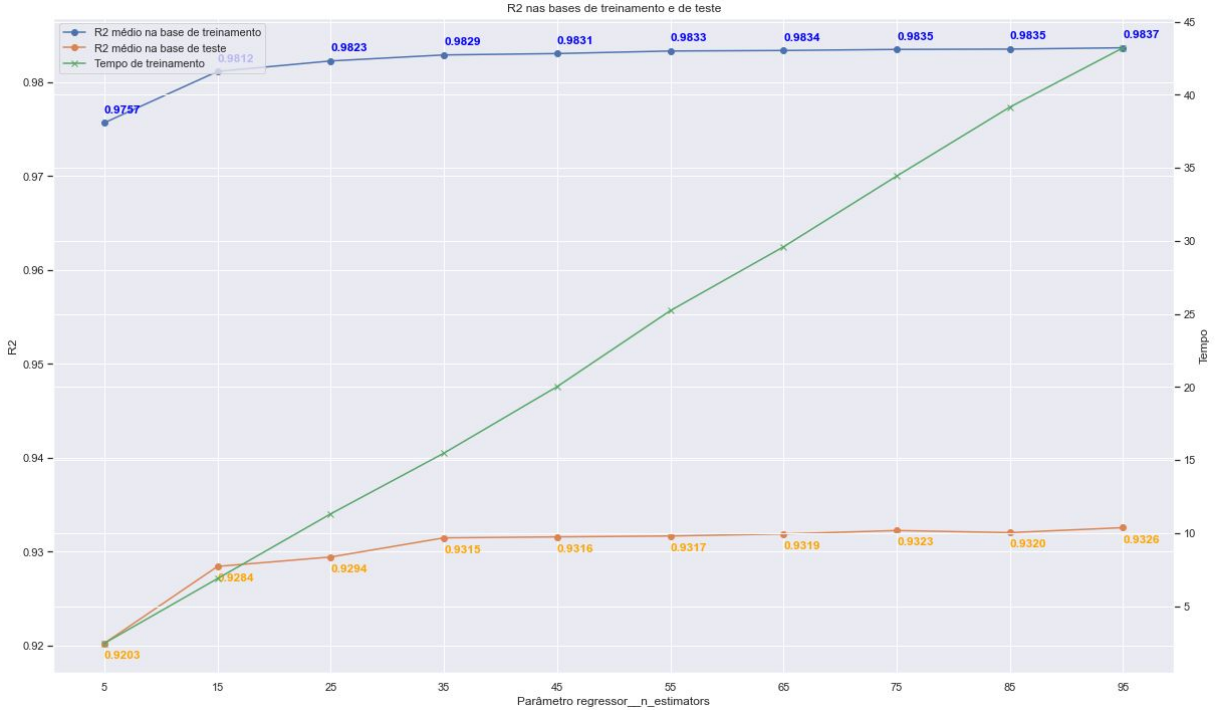
Florestas Aleatórias

- Regressor baseado em Florestas Aleatórias, conforme visto em sala de aula;
- Parâmetros utilizados:
  - **n_estimators:** número de estimadores (árvores de decisão) utilizados (5 a 95, intervalo de 20);
  - **max_depth:** tamanho máximo das árvores, tal como para árvore de decisão (10 a 28, intervalo de 2);
- Trata-se de um regressor *ensemble* que utiliza diferentes estimadores (não-correlacionados) mais simples e realiza uma votação ao final;
- É menos sujeito a *overfitting* por criar árvores não correlacionadas, diferentes entre si:
  - ***bootstrap aggregating***: subconjuntos aleatórios de amostras para treino de cada árvore;
  - ***bagging*:** subcaracterística aleatórios são utilizadas para treino de cada árvore;

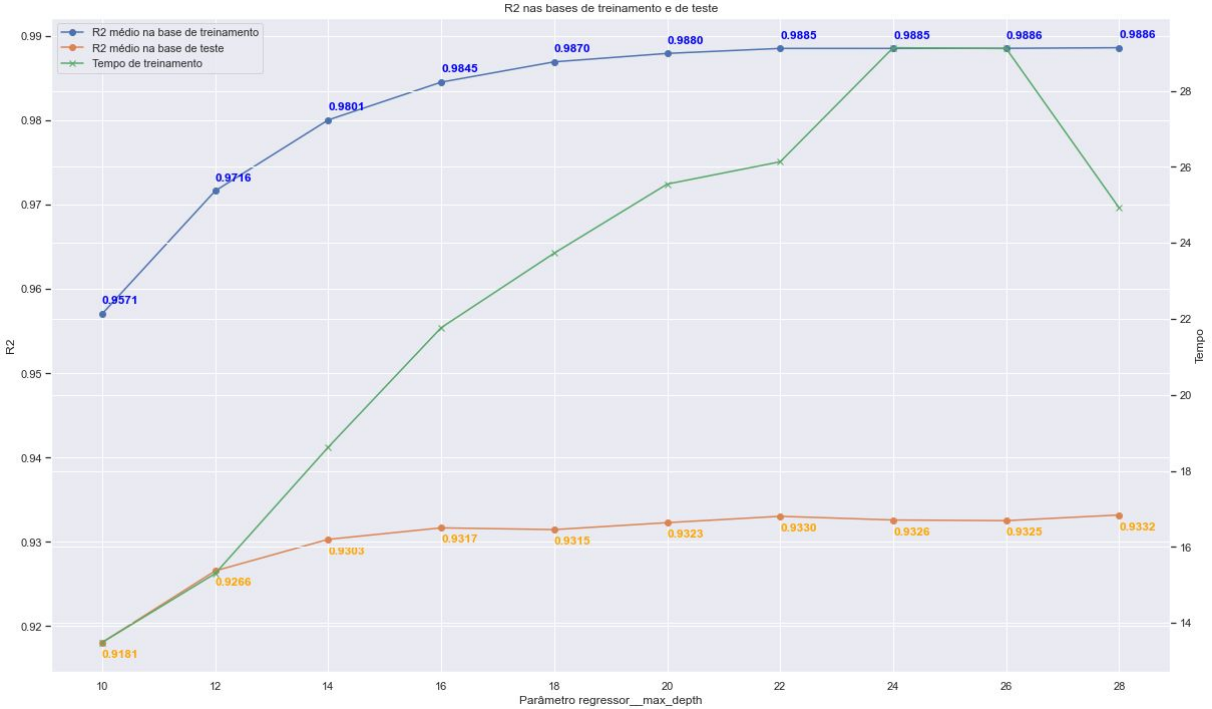
Florestas Aleatórias - Resultados

Melhores resultados para validação cruzada					
Parâmetros	R^2 médio de teste		R^2 médio de treino		Tempo médio de treino
	Média	Var	Média	Var	
max_depth = 22 n_estimators = 95	**0.935864**	0.008605	0.989764	0.000465	50.239506
max_depth = 26 n_estimators = 35	0.935770	0.008009	0.989308	0.000891	19.968931
max_depth = 24 n_estimators = 65	0.935515	0.008881	0.989773	0.000511	40.671070
max_depth = 26 n_estimators = 95	0.935311	0.008725	**0.990118**	0.000609	54.463199
max_depth = 28 n_estimators = 35	0.935223	0.008965	0.989248	0.000358	**17.560837**

Florestas Aleatórias - Resultados



Florestas Aleatórias - Resultados



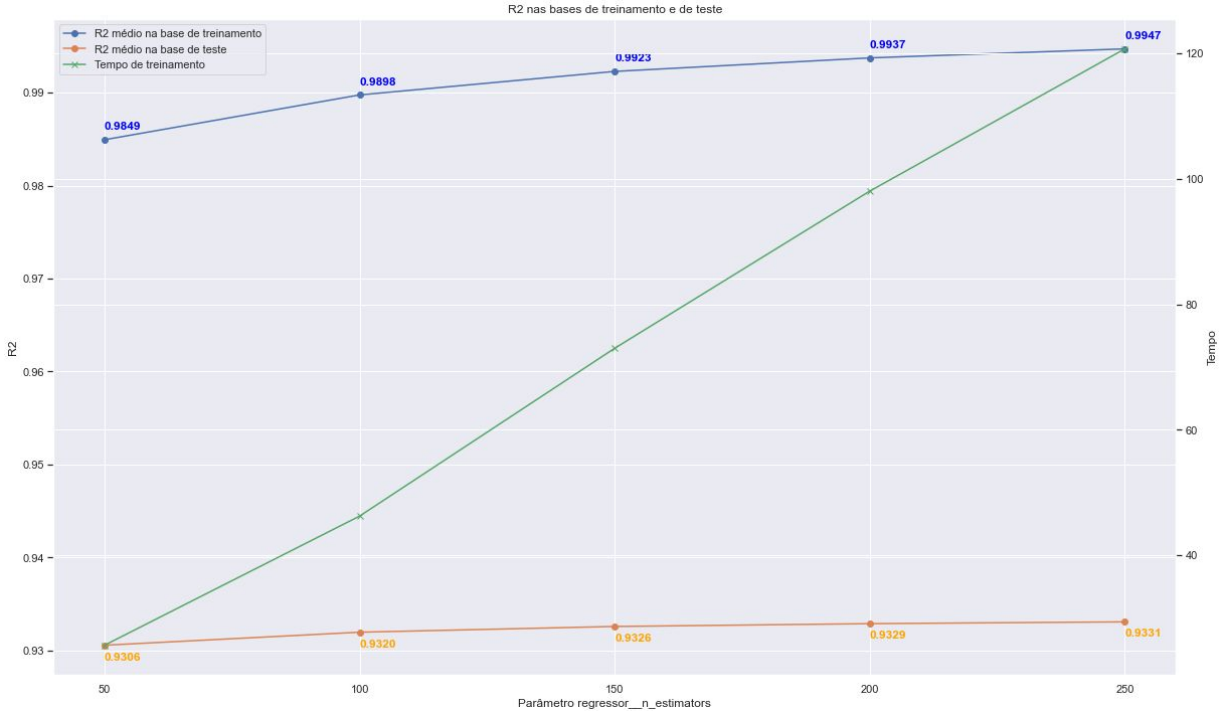
XGBoost

- O **eXtreme Gradient Boosting (XGBoost)** pertence à classe de algoritmos de *boosting* de gradiente com árvores de decisão, "semelhante" ao **AdaBoost** visto em sala de aula;
- Possui algumas otimizações para paralelismo e melhor uso do *hardware*, tais como:
  - Amostras são ordenadas uma única vez e armazenadas em um **formato de coluna comprimida (CSC)**;
  - Realiza **cortes na árvore** se para remover ramos com baixa probabilidade;
  - Lida bem com **dados esparsos** ao criar uma direção padrão para dados faltantes (ou iguais a zero);
  - Usa a **cache** de forma eficiente ao armazenar cálculos em uma estrutura de bloco;
  - Aplica **regularizações** (*Lasso* e *Ridge*) penalizando árvores complexas;
- Parâmetros utilizados:
  - **n_estimators:** número de estimadores (árvores de decisão) utilizados (50 a 250, intervalo de 50);
  - **max_depth:** tamanho máximo das árvores, tal como para árvore de decisão (5 a 25, intervalo de 5);
  - **min_child_weight:** semelhante ao **min_samples_split** mas leva em conta o peso das amostras (4 a 13, intervalo de 2)
- Trata-se de uma técnica que consiste no estado da arte em uma ampla gama de problemas de classificação e regressão

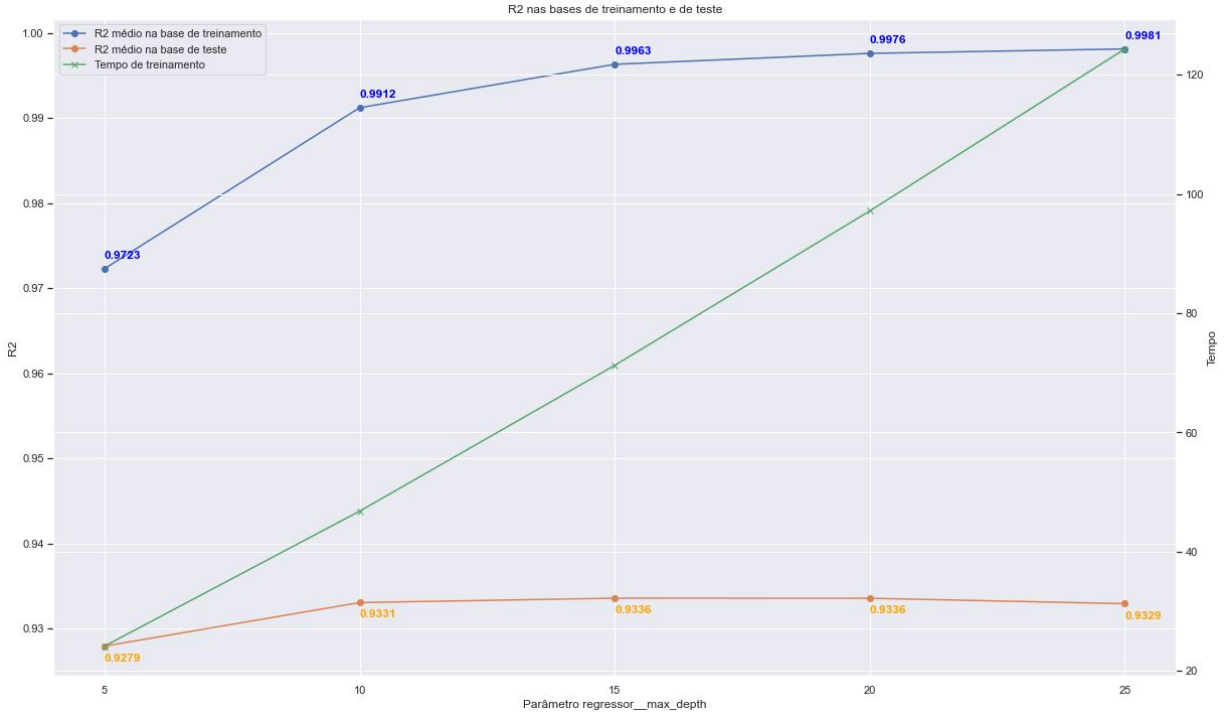
XGBoost - Resultados

Melhores resultados para validação cruzada					
Parâmetros	R^2 médio de teste		R^2 médio de treino		Tempo médio de treino
	Média	Var	Média	Var	
max_depth = 10 n_estimators = 250 min_child_weight = 7	**0.936270**	0.009296	**0.996822**	0.000703	83.276821
max_depth = 10 n_estimators = 200 min_child_weight = 7	0.936133	0.009323	0.995709	0.000921	64.164569
max_depth = 10 n_estimators = 150 min_child_weight = 7	0.935925	0.009315	0.994189	0.001047	**48.323372**
max_depth = 10 n_estimators = 250 min_child_weight = 9	0.935851	0.011346	0.995690	0.000602	80.770186
max_depth = 10 n_estimators = 200 min_child_weight = 9	0.935777	0.011226	0.994419	0.000785	62.585785

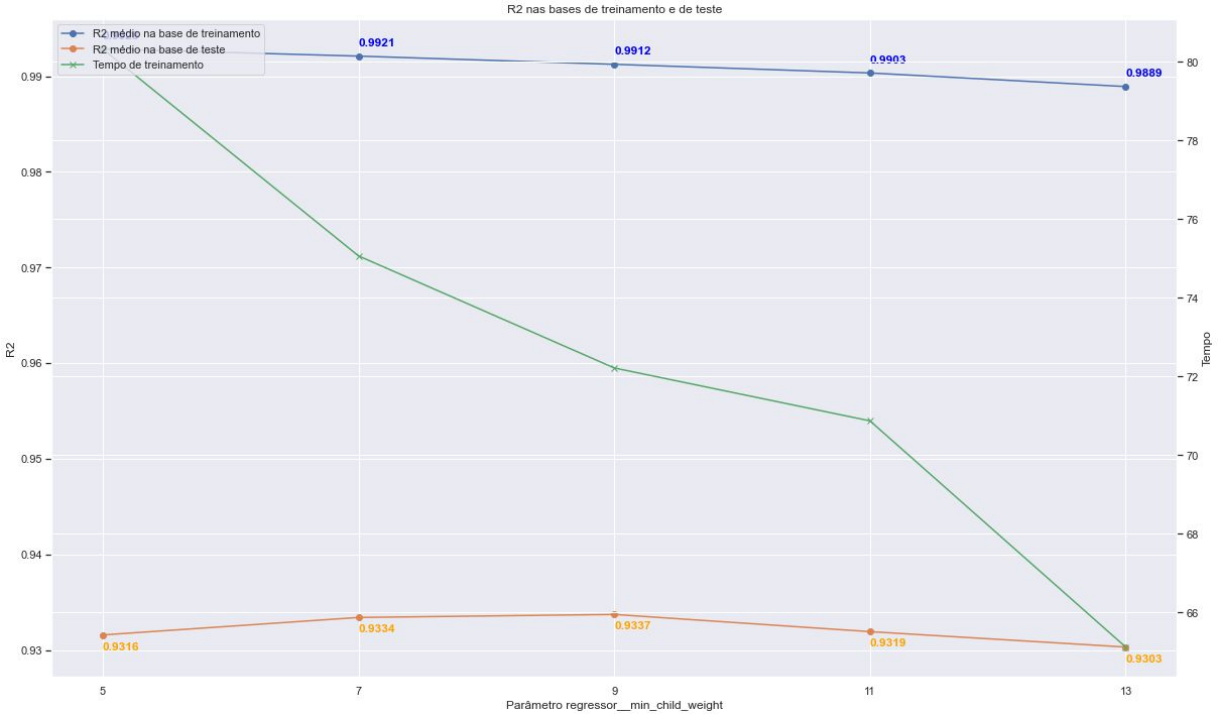
XGBoost - Resultados



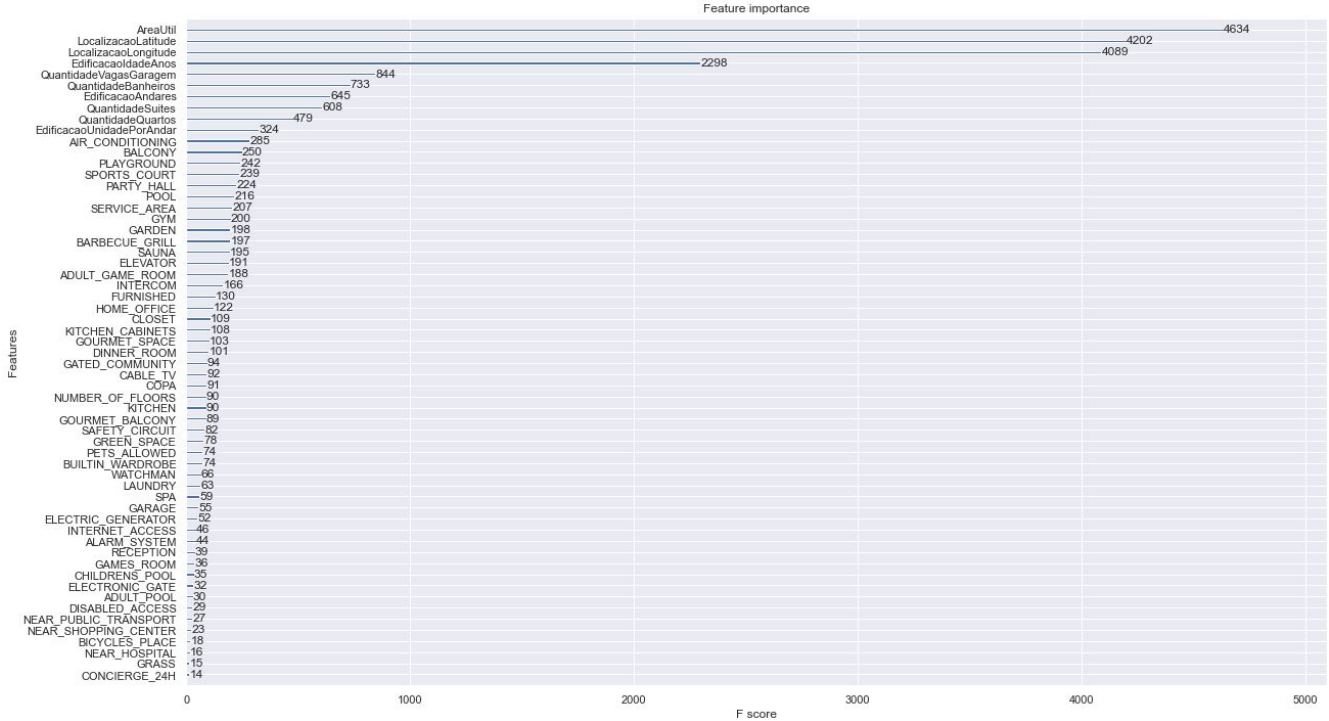
XGBoost - Resultados



XGBoost - Resultados



XGBoost - Resultados



Resumo dos resultados

Melhores resultados para cada regressor na validação cruzada						
Regressor	Parâmetros	R^2 médio de teste		R^2 médio de treino		Tempo médio de treino
		Média	Var	Média	Var	
XGBoostRegressor	max_depth = 10 n_estimators = 250 min_child_weight = 7	**0.936270**	0.009296	**0.996822**	0.000703	83.276821
RandomForestRegressor	max_depth = 22 n_estimators = 95	0.935864	0.008605	0.989764	0.000465	50.239506
DecisionTreeRegressor	max_depth = 14 min_samples_split = 14	0.903530	0.012969	0.964727	0.002290	1.023064
LinearRegression (baseline)	poli=2	0.791624	0.012143	0.829615	0.002453	**109.878032**
KNeighborsRegressor	pca_n_components=14 n_neighbors=7	0.712835	0.008641	0.788452	0.002627	4.303055
LinearRegression (baseline)	poli=1	**0.676327**	0.009123	**0.677829**	0.002231	**0.109237**

Discussão sobre os resultados

- Os regressores lineares utilizados como ***baseline*** não são capazes de capturar a **natureza não-linear do problema**:
    - Por exemplo, bairros próximos entre si podem apresentar uma alta discrepância nos valores de venda por m^2 ; (combinação entre os atributos longitude e latitude);
- O regressor por **kNN** não apresentou resultados satisfatórios possivelmente devido à **diminuição de dimensionalidade;**
- A regressão por **árvore de decisão** apresentou um **bom custo benefício** entre desempenho e resultado via **R^2**;
- Para resultados melhores foi necessário utilizar regressores ***ensemble***, seja via ***bagging*** **(Florestas Aleatórias)**, seja via ***boosting*** **(XGBoost);**

Trabalhos futuros

- Avaliação de **outros parâmetros** das técnicas já utilizadas de modo a aprimorar tanto o resultado quanto o desempenho das mesmas;
- Tal como em [2], utilização de técnicas de **redes neurais que avaliem também as imagens** e a **descrição textual** do imóvel:
  - As imagens permitem verificar aspectos de **decoração**, renovação e conservação;
  - A descrição textual permite uma visão mais detalhada das características do imóvel e região. Por exemplo: **pontos de interesse próximos** e renovações realizadas;
- Obtenção de dados de **transações efetivamente realizadas** de modo a correlacionar o preço do imóvel com o tempo médio de venda (o tempo que anúncio ficou ativo);

Referências

- [1] YAN, Ziyue. ZONG, Lu Zong, Lu. Spatial Prediction of Housing Prices in Beijing Using Machine Learning Algorithms. Disponível em: https://dl.acm.org/doi/10.1145/3409501.3409543. Acesso em 15/08/2021;
- [2] AFONSO, Bruno. Et. Al. Housing Prices Prediction with a Deep Learning and Random Forest Ensemble. Disponível em: https://sol.sbc.org.br/index.php/eniac/article/download/9300/9202/ . Acesso em 15/08/2021;
- [3] NUNES. David Brandão. Et. Al. Modelo de regressão linear múltipla para avaliação do valor de mercado de apartamentos residenciais em Fortaleza, CE. Disponível em: https://www.scielo.br/j/ac/a/bPkWZ5CznsJXYNHBWK78R8F/?lang=pt. Acesso em 15/08/2021;
- [3] PEREIRA, Júlio César. Et. Al. Construção de um modelo para o preço de venda de casas residenciais na cidade de Sorocaba-SP. Disponível em: Acesso em 15/08/2021. Disponível em: https://revista.feb.unesp.br/index.php/gepros/article/download/861/469. Acesso em 15/08/2021;
- [4] Portal ZapImóveis. Disponível em: https://www.zapimoveis.com.br/venda/. Acesso em 15/08/2021;
- [5] Documentação da biblioteca pandas. Disponível em: https://pandas.pydata.org/. Acesso em 15/08/2021;
- [6] Documentação da biblioteca numpy. Disponível em: https://numpy.org/. Acesso em 15/08/2021;
- [7] Documentação da biblioteca scitkit-learn. Disponível em: https://scikit-learn.org/stable/index.html. Acesso em 15/08/2021;
- [8] Documentação da biblioteca XGBoost. Disponível em: https://xgboost.readthedocs.io/en/latest/index.html. Acesso em 15/08/2021;
- [9] Documentação da biblioteca SciPy. Disponível em: https://docs.scipy.org/doc/scipy/index.html. Acesso em 15/08/2021;
- [10] Crédito imobiliário com recurso de poupança cresce 124% e tem novo recorde em 2021. Portal Valor Investe: https://valorinveste.globo.com/produtos/credito/noticia/2021/07/22/credito-imobiliario-com-recurso-de-poupanca-cresce-124percent-e-tem-novo-recorde-em-2021.ghtml. Acesso em 15/08/2021;
- [11] Dados revisados do déficit habitacional e inadequação de moradias nortearão políticas públicas. Portal do Ministério do Desenvolvimento Regional: https://www.gov.br/mdr/pt-br/noticias/dados-revisados-do-deficit-habitacional-e-inadequacao-de-moradias-nortearao-politicas-publicas. Acesso em 15/08/2021;
- [12] Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". Acesso em 15/08/2021;