**DOCUMENTAÇÃO DO PROJETO**

**1.OBJETIVO**

O objetivo desse trabalho foi aprender e treinar habilidades na área de Aprendizado de Máquina. O conjunto de dados utilizado foi o *House Prices*(fonte: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>). Trata-se de uma análise preditiva para prever o preço de vendas de imóveis (em dólares) utilizando uma base histórica de vendas comregistros de 1460 imóveis na cidade de Ames (Iowa, USA) e seus 79atributos.Através desse exercício pude desenvolver habilidades na área de *Analytics* *Advanced*, como seleção de atributos, *encoders* e aplicação de técnicas de regressão.

**2.ABORDAGEM PARA SOLUÇÃO DO PROBLEMA**

A análise foi construída em 2 diferentes arquivos: I) Pré-processamento dos dados e II) Construção dos modelos de Machine Learning:

*1) Pré-processamento dos dados:* processo de ingestão e transformação de dados.

Neste arquivo inspecionei os dados a procura de valores nulos, faltantes ou duplicados. Também explorei as relações entre as variáveis e realizei uma análise para detecção de *outliers* para excluir valores discrepantes que pudessem envisesar alguns atributos e comprometer os resultados dos modelos. Além disso apliquei transformações nos atributos categóricos para se tornarem numéricos e se adequarem ao formato exigido pelos algoritmos. Os dados também foram normalizados, com o objetivo de minimizar diferenças na escala das diferentes variáveis. E por fim, com o objetivo de reduzir a dimensionalidade dos dados, realizei uma análise de seleção de *features* eliminando aquelas que menos contribuíam para explicar a variação no atributo de interesse a ser predito (preço dos imóveis). No fim o conjunto de dados que seguiu para os modelos de machine learning ficou com a dimensão de 1204 registros e 42 atributos.

*II) Previsão com machine learning:* construção dos modelos estatísticos.

Na fase de aprendizado dos dados, testei quatro diferentes algoritmos de regressão:

* Linear Regression
* KNN Regressor
* Tree Decision Regressor
* Random Forest Regressor

Também construí um modelo *baseline* adicional baseado na média dos preços dos imóveis para simular o cenário mais simples que eu imaginei e confrontar com o desempenho dos 4 modelos mais complexos citados acima. para analisar o desempenho dos modelos utilizei as 4 principais métricas de problemas de regressão:

* Erro absoluto médio (MAE)
* Erro percentual médio (MAPE)
* Raiz do erro quadrático médio (RMSE)
* Coeficiente de determinação (R²).

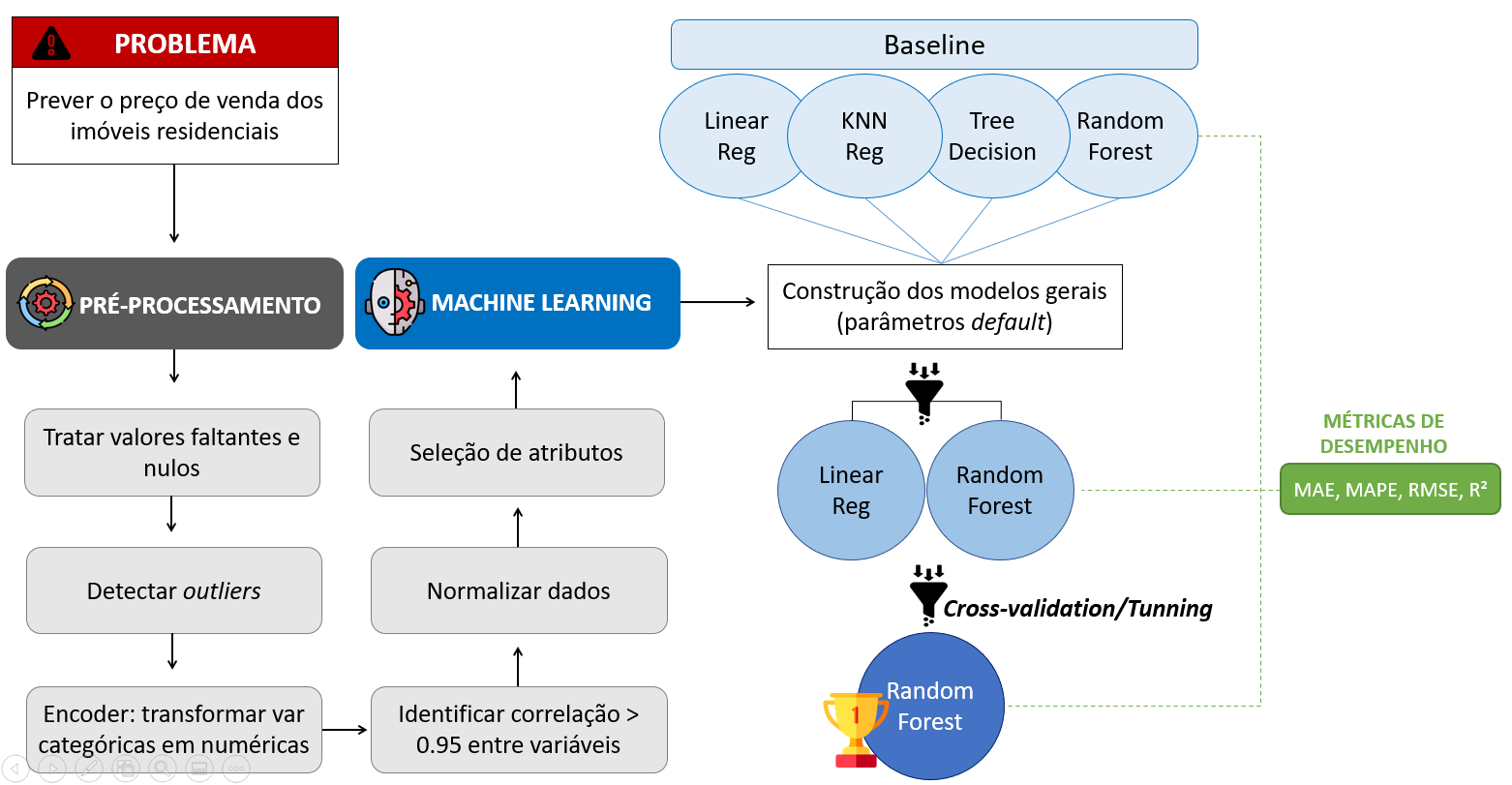
A descrição detalhada e a expressão matemática das métricas podem ser encontradas aqui: <https://scikit-learn.org/stable/modules/model_evaluation.html>**.**

O primeiro passo dessa análise foi construir modelos gerais usando os valores padrão dos hiperparâmetros dos algoritmos e checar se eles superavam o modelo *baseline*. Se sim, valeria a pena seguir e tentar melhorá-los posteriormente. Nessa etapa inicial realizei apenas um treino para aprendizado do modelo.

Na fase seguinte selecionei os 2 modelos que tiveram melhor desempenho nos testes da fase anterior com treino único. E então apliquei a técnica de validação cruzada, com múltiplos treinos e testes. O objetivo dessa implementação é diminuir os efeitos da aleatoriedade sobre a escolha dos dados utilizados durante a fase de aprendizado do modelo. Através dela garantimos que o modelo será treinado e testado com todos os registros pelo menos uma vez, e aumentamos as chances dele realmente aprender os padrões que explicam a variação do preço dos imóveis, ao invés de apenas "decorar" os valores. Também reduzimos as chances de ocorrer *overfitting* (sobre-ajuste e ótima performance do modelo na fase de treino, mas performance ruim na fase de teste e em produção).

Além da validação cruzada, nesse segunda etapa realizei o *tunning* dos dois melhores modelos através do ajuste dos hiperparâmetros, na tentativa de obter resultados ainda mais apurados. Assim pude comparar qual dos dois gerava as menores métricas de erro e obtive o meu modelo candidato final.

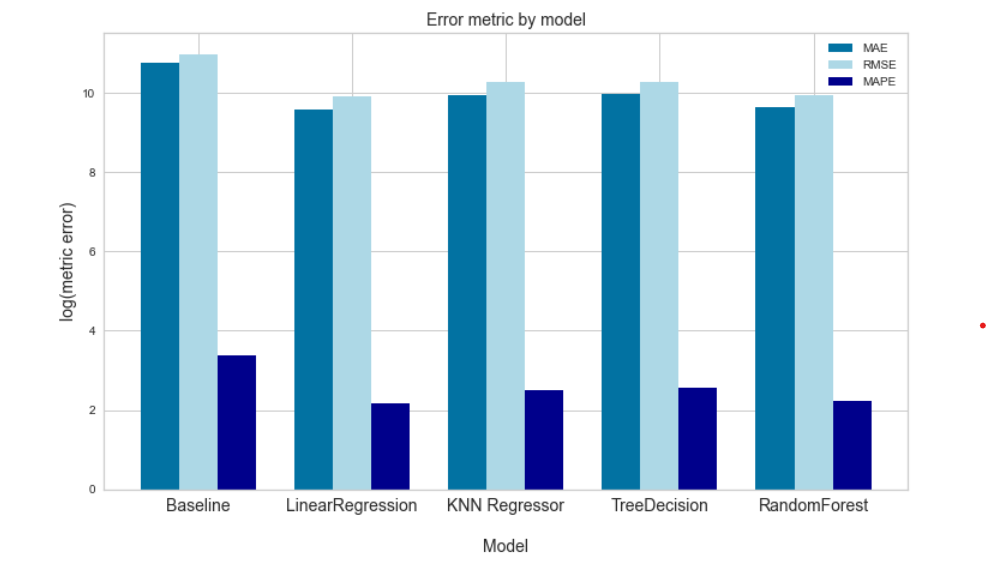
Abaixo um fluxograma ilustrando os processos realizados na abordagem do problema:



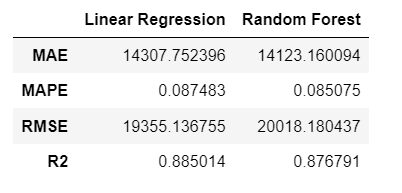
**3.RESULTADOS**

O modelo campeão que gerou as previsões de preços de imóveis com menores erros foi o Random Forest, embora o modelo Linear Regression também tenha performado bem, com valores ligeiramente superiores de MAPE e MAE. A **figura 1** mostra o desempenho de todos os modelos candidatos antes da implementação da validação cruzada, quando realizamos o teste depois de um treino único do algoritmo. Nela podemos ver que a Regressão Linear e o Random Forest tiveram os melhores desempenho, com a Regressão Linear performando ligeiramente melhor. Porém, quando implementei a validação cruzada com múltiplos treinos e testes (kfold = 10) e fiz o *tunning* dos modelos para esses dois candidatos, o Random Forest se saiu melhor para a maioria das métricas (**Tabela 2**).

**Figura1.** Métrica de erros para todos os modelos testados na primeira fase de modelagem (treino e teste únicos) e antes do ajuste dos hiperparâmetros (*tunning* dos modelos).



**Tabela2.** Métrica de erros para os 2 melhores modelos após a validação cruzada (múltiplos treinos e testes) e o *tunning*.



Como podemos ver acima, o MAPE (erro percentual médio) do algoritmo campeão foi de apenas 8.5%. Ou seja, quando o modelo previu os preços dos imóveis que tinham um valor de venda já conhecido, ele errou cerca de 8.5% (em média) para mais ou para menos em relação ao valor real desses imóveis. Transferindo a ideia para o cenário real de previsão de valores, isso significa que quando ele prever um valor de venda de U$100.000 para um imóvel - por exemplo - o valor real de venda de imóveis com as mesmas características desse deve estar entre U$92.000 e U$108.000. Olhando as métricas que estão na escala real da variável de interesse (dólares), temos um MAE (erro absoluto médio) de ~U$14320, enquanto o RMSE (raiz do erro quadrático) foi de ~U$19414.

**4.CONCLUSÃO E POTENCIAIS MELHORIAS**

O nosso melhor modelo foi o Random Forest capaz de prever o preço de novos imóveis com um erro médio de 8.5%. Com o objetivo de aprimorar a análise eu gostaria de aprender e aplicar outros tipos de algoritmos, como regressão Lasso/Ridge e modelos que usam a ideia de *gradient boosting* (como o *XGBoost*), na tentativa de conseguir erros menores que aqueles obtidos aqui. Também seria interessante aprender a implementação de técnicas de redução de dimensionalidade dos dados, como a PCA (Principal Component Analysis), para ver se temos algum ganho na performance.

Também gostaria de experimentar outros métodos de detecção de outliers para checar se os resultados divergem.

**5.FONTE DE DADOS**

Base e descrição dos dados no Kaggle: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

**6.REFERÊNCIAS**

1.Peng, Chao-Ying Joanne. Dong, Yiran. Principled missing data methods for researchers. Fonte: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3701793/>.

2.Curso Machine Learning e Data Science com Python de A a Z. Plataforma Udemy. <https://ibm-learning.udemy.com/course/machine-learning-e-data-science-com-python-y/>

3.Alto, Valentina. Detecting outliers with PyOD. <https://towardsdatascience.com/detecting-and-modeling-outliers-with-pyod-d40590a96488>