UNIVERSIDADE DE SÃO PAULO
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E COMPUTAÇÃO - ICMC

Gustavo Bartholomeu Trad Souza
Nº USP – 11219216

Desafio de Regressão

São Paulo 2023

Índice

Jupyter Notebook com código:
https://colab.research.google.com/drive/1duddc9WjiDO5f762SJsJOxsBSWnyVFmx?usp=sharing

1. Descrição do problema e análise dos dados

O conjunto de dados fornecido apresenta informações sobre produtos que são geralmente vendidos em bares e restaurantes, com informações relacionadas à fabricação, composição e venda por parte do fornecedor e também informações sobre vendas realizadas em um estabelecimento russo chamado Nelson Sauvin Bar com os itens presentes no primeiro dataset. Com essas informações é possível a realização da tarefa de regressão nos dados a fim de predizer a quantidade de vendas ou o lucro que seria obtido de um determinado produto, que está presente na base de produtos mas que não era comercializado na loja, dadas algumas informações sobre esse produto.

1.1. Descrição dos dados

O dataset é composto de duas tabelas com informações sobre produtos oferecidos e o histórico de vendas de um bar russo chamado Nelson Sauvin Bar.

O arquivo Product_range.csv contém dados sobre o conjunto de todos os tipos de produtos (cerveja, salgadinhos, refrigerantes e etc.) oferecidos aos clientes pelo Nelson Sauvin Bar. Este conjunto de produtos pode ser mais ou menos especializado ou genérico e deve corresponder às expectativas do mercado-alvo da empresa. Os produtos são descritos nas seguintes 8 colunas:

- **Product_code:** Código único de identificação de cada produto.
- **Vendor_code:** Código único que identifica o vendedor do produto.
- **Name:** Nome da SKU produto.
- **Retail_price:** Preço de varejo do produto.
- **Base_unit:** Únidade base de venda do produto (litros, peças, etc...).
- **Country_of_Origin:** País de fabricação do produto.
- **Size:** Tamanho do produto.
- **ABV (Alcohol by Volume):** Porcentagem de álcool por volume do produto - relevante para as bebidas alcóolicas.

O arquivo Transactions.csv contém dados sobre a venda de cerveja artesanal do bar. Este tipo de cerveja é tipicamente fabricado em pequenos lotes, usando métodos tradicionais e ingredientes de alta qualidade. Os bares de cerveja artesanal normalmente oferecem uma ampla seleção de cervejas artesanais de cervejarias locais e regionais, bem como de outras partes do mundo. As 8 colunas do dataset são:

- **Date_and_time_of_unloading:** Data e hora da compra.
- **Product_code:** Código único de identificação de cada produto.
- **Amount:** O número de unidades vendidas.
- **Sale_amount:** Quantia total de dinheiro que a empresa ganhou com a venda.
- **Discount_amount:** Quantia de dinheiro deduzida do preço original de um produto.
- **Profit:** Diferença entre a receita que a empresa obtém e os custos associados à produção e venda de seus produtos.

- **Percentage_markup:** Valor pelo qual o custo de um produto é aumentado para determinar o preço de venda.
- **Discount_percentage:** Porcentagem de desconto concedido a um produto.

1.2. Preparação do dataset

As colunas Product_code e Name serão removidas para os algoritmos de regressão pois, além de possuírem a mesma semântica, ambas identificam univocamente cada produto, portanto não são interessantes para a tarefa de regressão.

Em seguida o dataset com as transações será preparado de forma que todas as informações acerca de um determinado produto sejam condensadas em apenas uma linha. Para isso, inicialmente, serão mantidas as colunas referentes à quantidade de produtos vendidos (**Amount**) e o lucro total obtido com as vendas de cada produto (**Profit**). Em seguida esses valores serão somados de acordo com o id do produto para que seja calculado o total de vendas, o total de dinheiro adquirido com o produto e o lucro total do produto.

1.3. Atributos quantitativos

As colunas presentes nos datasets finais com uma semântica quantitativa são: **Retail_price**, **Size**, **ABV** no dataset com as informações de cada produto e as colunas: **Amount** e **Profit** no dataset de transações.

1.3.1. Substituição dos valores faltantes

O preenchimento dos dados faltantes no caso dos atributos quantitativos será realizado através do cálculo do valor médio dos valores contidos no atributo. A quantidade de valores faltantes em cada coluna é:

Retail_price	**436**
Size	**688**
ABV	**692**

No dataset de transações as entradas com valores faltantes também serão substituídas pela média da coluna, dessa forma será possível realizar a soma dos valores totais de cada produto. A quantidade de valores faltantes em cada coluna é:

Amount	**0**
Profit	**14**

1.3.2. Normalização dos dados

Será aplicada a normalização min-max nas colunas quantitativas da tabela de produtos de forma que os valores ficarão entre 0 e 1. Na tabela de transações será

realizada a soma dos valores quantitativos, portanto não será realizada a normalização desses valores.

1.4. Atributos categóricos

Apenas o dataset de produtos possui colunas categóricas. Os atributos categóricos nesse dataset são: **Base_unit**, **Vendor_code** e **Country_of_Origin**.

1.4.1. Substituição dos valores faltantes

Não será realizada a substituição dos valores faltantes para os atributos categóricos visto que a substituição pela moda pode desbalancear o dataset, portanto as linhas com valores faltantes serão removidas do dataset, evitando um aumento no desbalanceamento. A quantidade de valores faltantes em cada atributo é:

Base_unit 404
Country_of_Origin 1109
Vendor_code 1026

1.4.2. Codificação dos dado

As colunas categóricas possuem as seguintes quantidades de valores únicos (classes) em cada:

Base_unit : 3
Country_of_Origin : 27
Vendor_code : 260

Após o preenchimento dos dados faltantes é realizada a conversão dos atributos categóricos de classes de forma que estes possam assumir valores numéricos (strings ou inteiros com semântica de classe não podem ser passados para os algoritmos de aprendizado), para isso será criada uma coluna para cada classe presente nas colunas categóricas e essa nova coluna irá receber o valor 1 caso a entrada possuísse aquela classe na coluna original e 0 caso contrário. O dataset final de produtos possui 294 colunas e 4021 linhas.

1.5. Correlação entre os dados

Foi definida uma correlação máxima de 0.7, a partir da qual um dos elementos da correlação será removido. O dataset de transações não terá sua correlação medida pois os valores que ela apresenta serão os valores alvo da regressão e calculados separadamente, com o objetivo de testar qual deles pode ser melhor descrito pelas informações do produto. As colunas que serão removidas por possuirem uma alta correlação com outra coluna são:

Pieces, Netherlands, Arpus, Axiom, Bevog, Brasserie des Franches-Montagnes, BrewDog, Chang, CoolHead Brew, Dogma, Hitachino NEST, Lervig, Naparbier, Netherlands, Pohjala, Shimeji Beer, Snacks, Tempest Brewing, The Garage Brewery, To Ol, White Hag.

1.6. Dataset final

Para a criação dos datasets finais, inicialmente haverá a separação do dataset de produtos entre os produtos presentes no dataset de transações e os produtos que não estão. Essa separação permite que o algoritmo seja treinado e avaliado com os produtos oferecidos pelo bar e em seguida com o outro dataset é possível utilizar o algoritmo treinado para prever os lucros do bar com produtos que não são comercializados por ele.

Há 813 produtos que não são vendidos pelo **Nelson Sauvin Bar** e que poderiam ser vendidos caso se mostrassem vantajosos e 3208 produtos vendidos pelo bar e que os dados podem ser usados para a extração de informações que possam auxiliar na previsão. Para isso inicialmente o dataset **Product_range.csv** será separado em um dataset dos produtos contidos no **Transactions.csv** e outro com as informações dos produtos que não estão contidos nas transações. Em seguida o dataset de treino e o de transações será unido pela coluna de **Product_code**.

Dessa forma o dataset final possui 3208 linhas, cada uma contendo informações de um produto distinto e 274 colunas, sendo 2 delas as colunas alvo que deverão ser preditas pelos algoritmos de regressão.

2. Descrição das técnicas utilizadas

Os algoritmos de regressão deverão receber como entrada os dados dos produtos, sendo esses distribuídos em 272 colunas, e deverão prever um valor de saída. Para cada técnica serão utilizados 2 valores diferentes para Y, a quantidade de unidades vendidas e olucro total obtido com essas vendas.

2.1. Regressão Múltipla

O primeiro conjunto de métodos a ser utilizado é a regressão múltipla. Esse método consiste em utilizar várias variáveis simultaneamente para encontrar a saída desejada, e esses coeficientes são ajustados a fim de minimizar a soma dos quadrados residuais entre os valores alvos e os valores preditos. Serão utilizadas duas variantes desse método, uma delas sendo a regressão Linear e a outra a regressão Ridge com diferentes valores para o parâmetro alfa.

Predição da quantidade de produtos vendidos

Linear - MSE: 551114194489539251051924815872.00

R Score: -256610586171966209990328 32.00

Ridge alpha = 0 - MSE: 13045315284013933247623158 1958144.00

R Score: -607417851926233806647014 1952.00

Ridge alpha = 0.5 - MSE: 17742.22

R Score: 0.17

Ridge alpha = 1 - MSE: 17700.70

R Score: 0.18

Ridge alpha = 5 - MSE: 17865.31

R Score: 0.17

Ridge alpha = 15 - MSE: 18394.20

R Score: 0.14

Predição dos lucros obtidos por produto

Linear - MSE: 707629827836205516243507185975296 00.00

R Score: -17070260912969966893504921 6.00

Ridge alpha = 0 - MSE: 145309395407988907281472578071465820 16.00

R Score: -35053204304644475947898109952.00

Ridge alpha = 0.5 - MSE: 344215657.39

R Score: 0.17

Ridge alpha = 1 - MSE: 338054158.58

R Score: 0.18

Ridge alpha = 5 - MSE: 329628755.56

R Score: 0.20

Ridge alpha = 15 - MSE: 340567426.81

R Score: 0.18

2.2. Máquina de Vetor de Suporte para Regressão (SVR)

Ao invés de ajustar as margens para separar as classes, a idéia do SVR é ajustar a maior quantidade de exemplos entre as margens, sendo controlada por um hiperparâmetro epsilon.

Predição da quantidade de produtos vendidos

SVM - alpha = 0 - MSE: 21667.95

R Score: -0.01

SVM - alpha = 0.5 - MSE: 21668.30

R Score: -0.01

SVM - alpha = 1 - MSE: 21664.83

R Score: -0.01

SVM - alpha = 5 - MSE: 21601.70

R Score: -0.01

SVM - alpha = 20 - MSE: 21280.73

R Score: 0.01

Predição dos lucros obtidos por produto

SVM - alpha = 0 - MSE: 449950498.39

R Score: -0.09

SVM - alpha = 0.5 - MSE: 449927520.40

R Score: -0.09

SVM - alpha = 1 - MSE: 449992667.86

R Score: -0.09

SVM - alpha = 5 - MSE: 449987651.79

R Score: -0.09

SVM - alpha = 20 - MSE: 450053891.02

R Score: -0.09

2.3. Árvores de Regressão

Assim como no método para classificação, a Árvore de Decisão para Regressão objetiva predizer um valor. Contudo, neste caso, é uma variável quantitativa.

Predição da quantidade de produtos vendidos

Decision Tree - max_depth = 2 - MSE: 20340.67

R Score: 0.05

Decision Tree - max_depth = 5 - MSE: 18903.81

R Score: 0.12

Decision Tree - max_depth = 10 - MSE: 13895.18

R Score: 0.35

Decision Tree - max_depth = 20 - MSE: 16908.14

R Score: 0.21

Decision Tree - max_depth = 50 - MSE: 25089.78

R Score: -0.17

Decision Tree - max_depth = 100 - MSE: 24934.60

R Score: -0.16

Predição dos lucros obtidos por produto

Decision Tree - max_depth = 2 - MSE: 453816443.46

R Score: -0.09

Decision Tree - max_depth = 5 - MSE: 299264228.31

R Score: 0.28

Decision Tree - max_depth = 10 - MSE: 500244661.25

R Score: -0.21

Decision Tree - max_depth = 20 - MSE: 557664168.60

R Score: -0.35

Decision Tree - max_depth = 50 - MSE: 649100889.69

R Score: -0.57

Decision Tree - max_depth = 100 - MSE: 584755689.11

R Score: -0.41

2.4. Ensembles

2.4.1. Bagging (Bootstrap AGGregatION)

Utiliza **múltiplas versões do conjunto de treinamento**, cada um sendo criado selecionando $n' < n$ exemplos a partir de D **com reposição**. Cada um dos subconjuntos formados é utilizado para **treinar um modelo paralelo** e, ao final, **o resultado final é composto pela média dos valores**.

Geralmente, o modelo é exatamente o mesmo (árvores de decisão, SVMs, ou outro qualquer).

Predição da quantidade de produtos vendidos

Bagging - n_estimators = 5 - MSE: 21711.18

R Score: -0.01

Bagging - n_estimators = 10 - MSE: 21716.21

R Score: -0.01

Bagging - n_estimators = 50 - MSE: 21736.39

R Score: -0.01

Bagging - n_estimators = 100 - MSE: 21733.73

R Score: -0.01

Predição dos lucros obtidos por produto

Bagging - n_estimators = 5 - MSE: 440719561.27

R Score: -0.06

Bagging - n_estimators = 10 - MSE: 441054502.42

R Score: -0.06

Bagging - n_estimators = 50 - MSE: 441085953.71

R Score: -0.06

Bagging - n_estimators = 100 - MSE: 441099991.25

R Score: -0.06

2.4.2. Random Forest

Random Forest é o método de Bagging mais conhecido e utilizado, sendo baseado em Árvores de Decisão, mas que possui algumas particularidades:

- A execução do bootstrap busca criar subconjuntos descorrelacionados entre si;
- Calcula a média dos resultados.

Predição da quantidade de produtos vendidos

Random Forest - n_estimators = 2, max_depth = 2 - MSE: 16930.85

R Score: 0.21

Random Forest - n_estimators = 2, max_depth = 10 - MSE: 19398.93

R Score: 0.10

Random Forest - n_estimators = 2, max_depth = 50 - MSE: 26155.54

R Score: -0.22

Random Forest - n_estimators = 10, max_depth = 2 - MSE: 15782.39

R Score: 0.27

Random Forest - n_estimators = 10, max_depth = 10 - MSE: 13364.15

R Score: 0.38

Random Forest - n_estimators = 10, max_depth = 50 - MSE: 11903.75

R Score: 0.45

Random Forest - n_estimators = 50, max_depth = 2 - MSE: 16882.14

R Score: 0.21

Random Forest - n_estimators = 50, max_depth = 10 - MSE: 12043.62

R Score: 0.44

Random Forest - n_estimators = 50, max_depth = 50 - MSE: 15558.20

R Score: 0.28

Predição dos lucros obtidos por produto

Random Forest - n_estimators = 2, max_depth = 2 - MSE: 361153178.38

R Score: 0.13

Random Forest - n_estimators = 2, max_depth = 10 - MSE: 1251947275.99

R Score: -2.02

Random Forest - n_estimators = 2, max_depth = 50 - MSE: 1077772435.90

R Score: -1.60

Random Forest - n_estimators = 10, max_depth = 2 - MSE: 369742758.60

R Score: 0.11

Random Forest - n_estimators = 10, max_depth = 10 - MSE: 478314714.89

R Score: -0.15

Random Forest - n_estimators = 10, max_depth = 50 - MSE: 397019189.89

R Score: 0.04

Random Forest - n_estimators = 50, max_depth = 2 - MSE: 341175591.64

R Score: 0.18

Random Forest - n_estimators = 50, max_depth = 10 - MSE: 336207546.35

R Score: 0.19

Random Forest - n_estimators = 50, max_depth = 50 - MSE: 354410041.88

R Score: 0.15

2.4.3. AdaBoost (Adaptive Boosting)

A ideia de Boosting é criar um regressor base e ir adicionando novos regressores para fortalecer este primeiro, aumentando sua performance.

Predição da quantidade de produtos vendidos

AdaBoost - n_estimators = 2 - MSE: 20784.48

R Score: 0.03

AdaBoost - n_estimators = 10 - MSE: 19171.31

R Score: 0.11

AdaBoost - n_estimators = 50 - MSE: 24796.51

R Score: -0.15

AdaBoost - n_estimators = 100 - MSE: 19274.41

R Score: 0.10

Predição dos lucros obtidos por produto

AdaBoost - n_estimators = 2 - MSE: 460783510.14

R Score: -0.11

AdaBoost - n_estimators = 10 - MSE: 596593474.17

R Score: -0.44

AdaBoost - n_estimators = 50 - MSE: 4346514337.58

R Score: -9.49

AdaBoost - n_estimators = 100 - MSE: 753218013.30

R Score: -0.82

2.5. Multi-layer Perceptron (MLP)

É uma técnica que consiste em ampliar sequencialmente conjuntos de Perceptrons a fim de criar um aproximador de funções não linear. Através de operações com parâmetros ajustáveis chamados pesos e bias os dados de entrada são transformados em uma saída que é uma predição do atributo alvo, em seguida o erro entre o valor real e a predição é calculado e à partir desse erro é calculado o gradiente negativo (pois o objetivo é reduzir o erro) para cada parâmetro ajustável e os ajustes são aplicados.

Predição da quantidade de produtos vendidos

MLP - hidden_units = 100, activation = logistic - MSE: 16997.36

R Score: 0.21

MLP - hidden_units = 100, activation = relu - MSE: 17672.86

R Score: 0.18

MLP - hidden_units = (100, 50), activation = logistic - MSE: 17236.98

R Score: 0.20

MLP - hidden_units = (100, 50), activation = relu - MSE: 17320.01

R Score: 0.19

MLP - hidden_units = (100, 100), activation = logistic - MSE: 16487.45

R Score: 0.23

MLP - hidden_units = (100, 100), activation = relu - MSE: 17254.12

R Score: 0.20

MLP - hidden_units = (50, 50, 50), activation = logistic - MSE: 17018.22

R Score: 0.21

MLP - hidden_units = (50, 50, 50), activation = relu - MSE: 17127.15

R Score: 0.20

Predição dos lucros obtidos por produto

MLP - hidden_units = 100, activation = logistic - MSE: 458414612.79

R Score: -0.11

MLP - hidden_units = 100, activation = relu - MSE: 356137724.20

R Score: 0.14

MLP - hidden_units = (100, 50), activation = logistic - MSE: 468338628.25

R Score: -0.13

MLP - hidden_units = (100, 50), activation = relu - MSE: 356217975.08

R Score: 0.14

MLP - hidden_units = (100, 100), activation = logistic - MSE: 459231943.26

R Score: -0.11

MLP - hidden_units = (100, 100), activation = relu - MSE: 365281621.41

R Score: 0.12

MLP - hidden_units = (50, 50, 50), activation = logistic - MSE: 467733558.02

R Score: -0.13

MLP - hidden_units = (50, 50, 50), activation = relu - MSE: 389236238.90

R Score: 0.06

3. Interpretação dos resultados obtidos

Dentre os modelos experimentados, aquele que obteve os melhores resultados para a predição da quantidade de vendas de um determinado produto foi o Random Forest - n_estimators = 10, max_depth = 50. Esse algoritmo obteve um erro quadrático médio de 11903.75 e um R2 Score de 0.44.

E o que obteve os melhores resultados na predição do lucro total obtido com um determinado produto foi o Decision Tree - max_depth = 5. Esse algoritmo obteve um erro quadrático médio de 299264228.30 e um R2 Score de 0.27.

Esse resultado demonstra que o algoritmo a ser escolhido é bastante influenciado pela variável que está sendo predita, uma vez que com os mesmos dados de entrada, diferentes algoritmos se ajustaram melhor a diferentes tarefas. Ambas as técnicas são baseadas em Árvores de decisão, porém para a previsão da quantidade de produtos vendidos a técnica de Ensembles se mostrou mais adequada, enquanto que na predição do lucro obtido com cada produto a Árvore de decisão simples com profundidade máxima de 5 se sobressaiu. Portanto para a realização da previsão nos produtos que não são comercializados pelo bar, será utilizada a Random Forest com 10 estimators e profundidade 50 para a previsão da quantidade de produtos que serão vendidos e a árvore de decisão com profundidade máxima igual a 5 para a previsão do lucro total obtido em cada produto.

Os 3 produtos com a possibilidade de gerarem maior lucro e que não são vendidos pelo bar são:

Nome	Product_code	Profit
Weihenstephaner 1516	3737	305553.01
Weihenstephaner Pils 5,1%	2961	238048.64
Duchesse Cherry Barrel	4827	147667.33

E os 3 produtos com a possibilidade de terem um maior número de vendas e que não são vendidos pelo bar são:

Nome	Product_code	Amount
Weihenstephaner 1516	3737	755.8
Weihenstephaner Pils 5,1%	2961	677.6
Jaws Saison Raspberry Edition / 7.7% / 0.5 but	1749	620.9

É possível perceber que os 2 produtos com possibilidade de serem os mais rentáveis também são os que tem mais possibilidade de serem os mais vendidos. Isso mostra que os algoritmos obtiveram resultados semelhantes, uma vez que há a correlação entre quantidade e lucro, que condizem com a realidade do problema.

4. Conclusão

A utilização dos algoritmos de regressão permitiu prever os produtos mais rentáveis para um estabelecimento a partir de dados já existentes gerados pelo próprio estabelecimento. É interessante observar que, apesar de serem treinados com os mesmos dados, os valores de saída foram melhores descritos por algoritmos diferentes, a pesar de ambos serem baseados no mesmo princípio (árvores de decisão), o que mostra a importância dos valores alvo no desempenho do algoritmo, uma vez que valores com semânticas e comportamentos diferentes são melhor descritos por técnicas diferentes.

Com uma quantidade maior de dados seria possível realizar um treinamento mais eficiente que poderia trazer ainda mais futuros lucros para o estabelecimento.

5. Apêndice (descrição das técnicas de Classificação apresentadas pelos colegas durante os seminários)

5.1. Estimando a mudança na vantagem do time da casa no futebol durante a pandemia de covid-19 usando regressão de poisson bivariada.

Objetivo: Devido à pandemia os estádio se encontravam vazios, portanto foi realizado um estudo para comparar o efeito da presença da torcida no desempenho do time da casa. Foram coletados dados de jogos que ocorreram com estádios vazios e estádios cheios e houve uma comparação entre ambos, analisando a quantidade de gols e cartões amarelos.

Algoritmos utilizados:

- Regressão de Poisson Bivariada
- Regressão Linear

Métricas de avaliação:

- valor de R para checar se o modelo convergiu

Resultados:

- Em 11 campeonatos, a vantagem do time da casa diminuiu tanto para gols quanto para cartões amarelos.
- Em 4 campeonatos, a vantagem diminuiu apenas em relação aos cartões amarelos.
- Nos 4 países em que foram analisados mais de um campeonato, apenas a Espanha teve resultados semelhantes em ambas as competições

5.2. Análise preditiva da pressão sanguínea durante a Hemodiálise.

Objetivo: Uma complicação comum durante a hemodiálise é a hipotensão intradialítica (HID), caracterizada pela redução do volume de sangue que circula no corpo. Medir a HID é uma rotina comum na prática clínica. No entanto, essa tarefa é desafiadora em pacientes submetidos à hemodiálise devido às variações significativas na pressão arterial pré, intra e pós-diálise. Nesse sentido, o artigo utiliza de algoritmos de aprendizado de máquina para prever e perfilar a pressão sanguínea durante a hemodiálise.

Modelos utilizados:

- Extreme gradient boosting (XGBOOST)
- Random Forest
- Support vector regression (SVR)
- Regressão Linear

- Regressão LAASO.

Métricas de avaliação:

- MAE - Mean Absolute Error
- RMSE - Root Mean Squared Error
- R QUADRADO

Resultados:

- O algoritmo de XGBOOST sofreu um overfitting.
- O algoritmo de Random Forest apresentou os melhores resultados, com o menor MAE e menor R2.
- Os modelos lineares sofreram menos overfitting.
- O algoritmo de LAASO obteve o pior desempenho dentre os algoritmos de regressão linear.
- O método de Ensemble teve a maior capacidade preditiva.

Conclusão:

- A base de dados pode ter pouca variabilidade, uma vez que foi coletada em somente um hospital. Uma base de dados mais completa pode gerar melhores resultados.
- A quantidade de dados sendo medidos também poderia ser aumentada, com mais informações sobre o estado e os sintomas do paciente durante o procedimento de hemodiálise é possível que os algoritmos apresentam melhores resultados.

5.3. Regressão Linear Modal

Objetivo: O objetivo de uma regressão linear é encontrar um valor do y predito que minimize a função de perda através do método dos mínimos quadráticos. Portanto, dado um conjunto X, a regressão linear fornece a média da variável resposta Y, dessa forma foi proposta uma maneira de modelar a moda condicional da variável resposta Y dado o conjunto X, ao invés da média condicional como é comum.

Modelos utilizados para comparação:

- Regressão tradicional (média)
- Regressão mediana
- Estimativa-MM
- MODLR

Conclusão:

- O MODLR é robusto para outliers e também é robusto para

distribuições de erros condicionais de cauda longa.

* O MODLR pode fornecer intervalos de previsão (confiança)

mais curtos do que outras abordagens de regressão linear

para um nível de confiança.

* A ideia da MODLR pode ser generalizada para outros modelos como regressão não-linear, regressão não paramétrica e regressão linear com coeficientes parciais variáveis.