· Simple Linear Regression

· Multiple Linear Regression

· Polynomial Regression

· Logistic Regression

· Decision trees Regression

- Random Forest Regression

· Ridge Regression

· Lasso Regression

· Artificial Neural Networks for Regression analysis

**- Regressão Linear Simples**

Permite a geração de uma reta y = b0 + b1\*x, onde y é o valor a ser previsto e x é o dado a ser considerado. A reta atravessa os valores correlacionados

Como a base possui mais de uma Array, o método não é aplicável.

**- Regressão Linear Múltipla**

Extensão da regressão linear simples, permite a geração da fórmula y = b0 + b1\*x1 + b2\*x2 + ... + bn\*xn, a qual permite a extração de n dados de base para a previsão.

Para os dados de Hossain:

Fórmula obtida: y = (-1.0534540) + (0.0288384\*x1) + (-0.00109075\*x2) + (0.00144061x3) + (7.34752x4) + (-0.958468\*x5).

X1 = Inclination Angle

X2 = Speed S rpm

X3 = Feed fy mm/min

X4 = Feed fx mm

X5 = Depth of Cut t mm

Score = 0,7064% Training, 0,4510% Test

Mean Absolute Error = 0,8270

Mean Squared Error = 1,0607

Root Mean Squared Error = 1,0299

**- Regressão Linear Polinomial**

Permite a geração da fórmula y = b0 + b1\*x1 + b2\*x12 + ... + bn\*x12 . Pode haver mais de um valor x. No caso da regressão polinomial, não é gerada uma reta, mas sim uma curva, para que as previsões estejam mais próximas dos resultados.

Para os dados de Hossain:

Degree: 2

**- Regressão com Árvores de Decisão**

Com o gráfico dividido, determina diversas condições para que a saída seja estabelecida

**-Regressão com Random Forest**

Decisões randômicas, com aprendizagem em conjunto. Obtém a média de todas as decisões

Para os dados de Hossain:

N\_estimators = 10

**- Maquinas de Vetores de Suporte**

Procura o melhor hiperplano. Assim, obtem uma superfície com a ajuda de vetores de suporte

**- Regressão utilizando DNN**

Permite o estabelecimento de pesos para cada valor de entrada, que com o acréscimo do valor, aciona neurônios posteriores, até que um valor final seja atingido.

Configurações:

DNN (5-3-1)

Ativação: relu-relu-linear

Loss: mean\_squared\_error

Optimizer: adam

Metrics: mean\_squared\_error

Batch\_size: 2

Epochs: 500

Tabela dos valores estatísticos utilizando os dados de Hossain

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Método** | **MAE** | **MSE** | **RMSE** | **Score** | **Reprodutivel** |
| Linear Múltipla | Treino: 0,6279  Teste: 0,8271 | Treino: 0,6096  Teste: 1,0607 | Treino: 0,7808  Teste: 1,0299 | Treino: 0,7064  Teste: 0,4510 | Sim |
| Polinomial | Treino: 0,5229  Teste: 0,7889 | Treino: 0,3977  Teste: 1,0083 | Treino: 0,6306  Teste: 1,0041 | Treino: 0,8084  Teste: 0,4782 | Sim |
| Árvore de Decisão | Treino: 0,0000  Teste: 0,7756 | Treino: 0,0000  Teste: 1,0214 | Treino: 0,0000  Teste: 1,0214 | Treino: 1,0000  Teste: 0,4714 | Pouco |
| Floresta Randômica | Treino: 0,1981  Teste: 0,8325 | Treino: 0,0840  Teste: 0,9446 | Treino: 0,2899  Teste: 0,9719 | Treino: 0,9595  Teste: 0,5111 | Não |
| SVR | Treino: 0,3634  Teste: 0,7048 | Treino: 0,3224  Teste: 0,7736 | Treino: 0,5678  Teste: 0,8795 | Treino: 0,8331  Teste: 0,5996 | Sim |
| DNN | Treino: 0,3110  Teste: 0,8112 | Treino: 0,1809  Teste: 1,0175 | Treino: 0,4254  Teste: 1,0087 |  | Não |