**Projeto**

Ao comprar algum tipo de aço para elaboração de algum projeto é essencial que suas propriedades mecânicas estejam dentro das normas estabelecidas uma vez que suas propriedades mecânicas podem prejudicar o resultado final.

Tendo em vista a confiabilidade do material estabelecido iremos executar um treinamento baseado nas seguintes propriedades mecânicas, massa específica, dureza e densidade. E o aço utilizado para análise é o SAE 1020, com as propriedades da tabela abaixo.

A ideia da análise é após fazer a conformação do aço, uma amostra ser recolhida e fazer ensaio das propriedades estabelecidas. A partir dos dados desses resultados podemos dizer se o aço é de boa ou má qualidade.

|  |  |  |
| --- | --- | --- |
| Massa Especifica | Densidade | Dureza |
| 1,0098 g/cm3 | 0.00787 | 131 |

Não foi encontrado uma base de informação de margem de erro do que seria aceitável para cada propriedade, dai, foi estabelecido um valor de 1% para mais e para menos no qual esse valor seria aceitável.

Além disso foi preciso criar uma base de dados, em razão de dados não serem encontrados com facilidade na internet. Portanto a base de dados ficou de acordo com a imagem abaixo.

A screenshot of a cell phone

Description automatically generated

Os dados gerados foram gerados da seguinte forma utilizando mysql

|  |
| --- |
| **Gerando dados aceitáveis com razão de 1%**  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.95)+7.95,2),FLOOR(RAND()\*(132-129+1)+129) ,TRUNCATE(RAND()\*(0.10098-0.103)+0.103,5),TRUE );  **Gerando dados não aceitáveis com razão de 2%**  **DENSIDADE**  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.96-8)+8,2),FLOOR(RAND()\*(132-129+1)+129) ,TRUNCATE(RAND()\*(0.10098-0.103)+0.103,5),false );  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.7)+7.7,2),FLOOR(RAND()\*(132-129+1)+129) ,TRUNCATE(RAND()\*(0.10098-0.103)+0.103,5),false );  **DUREZA**  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.95)+7.95,2),FLOOR(RAND()\*(133-140+1)+140) ,TRUNCATE(RAND()\*(0.10098-0.103)+0.103,5),FALSE );  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.95)+7.95,2),FLOOR(RAND()\*(128-120+1)+120) ,TRUNCATE(RAND()\*(0.10098-0.103)+0.103,5),FALSE );  **MASSA**  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.95)+7.95,2),FLOOR(RAND()\*(132-129+1)+129) ,TRUNCATE(RAND()\*(0.10098-0.1)+0.1,5),FALSE );  INSERT INTO aco (densidade, dureza, massa, fabricacao)VALUES (TRUNCATE(RAND()\*(7.79-7.95)+7.95,2),FLOOR(RAND()\*(132-129+1)+129) ,TRUNCATE(RAND()\*(0.11-0.104)+0.104,5),FALSE ); |

Obs: esses códigos foram repetidos inúmeras vezes para criar uma base de dados

**Resultados**

Para fazer a análise foram utilizados os seguintes métodos, Tree J48, KNN, Naive Bayes e Regressão Logística.

Após fazer a análise (Análise por Python) todos os resultados deram matriz de confusão de 100%. Foi feita a análise utilizando pelo programa WEKA que por algum motivo deu resultados diferente, tendo KNN com resultado de 97% e Tree J48 de 91%.

Um dos motivos pelo qual o resultado em Python possa ter dados todos 100% é pelo fato da normalização dos resultados, pegando apenas os resultados mais expressivos. O que facilitou na hora do treinamento da máquina, haja vista que a base de dados segue uma forma logica linear, o que torna de mais fácil entendimento para máquina.

|  |  |
| --- | --- |
| 23 | 0 |
| 0 | 36 |

Matriz confusão KNN

|  |  |
| --- | --- |
| 23 | 0 |
| 0 | 46 |

Matriz confusão TreeJ48

O KNN foi o método mas efetivo pelo fato dele trabalhar por itens de aproximação, utilizando funções de distancia o qual chega a valores mais aproximados. Separando esses grupos por similaridade assim fica mais fácil explora-las em diversos grupos