Treinamento, validação e teste.

1-Pelo o'que o treinamento é responsável?
O treinamento é responsável pelo aprendizado do algoritmo, por isso atribuímos atribuímos uma grande quantidade de dados para essa fase você viu também que a fase de teste servia para testar o aprendizado do algoritmo, para isso na segunda divisão de dados era atribuída, nessa aula iremos estudar os conceitos de treinamento validação e teste para que o aprendizado de algoritmo expresse realmente a realidade e tenha uma utilidade é necessário que:

1.1- Para que o aprendizado do algoritmo expresse realmente a realidade o'que é preciso?
É necessário treinamento a validação dos conhecimentos adquiridos e do teste já vimos em aulas anteriores que a primeira divisão do “data sets” é feita em dados para treinamento e dados para teste o modelo preditivo gerado pelo algoritmo receberá essa aprendizagem dos dados de treinamento e o mesmo modelo será testado pelos dados de teste, então é medida sua acurácia que define o ninho de exatidão nos resultados obtidos uma divisão comum dos testes cabe 70% a 75% para dados de treinamento e 25 a 30% para dados de teste.

1.2-A divisão comum?
Essa divisão é feita na construção do modelo preditivo e é essencial que esta etapa seja cumprida, se ela for feita de maneira errada pode gerar-se problemas no modelo para evitar problemas nas amostras dos testes dos dados deve se separar os dados forma aleatória ,ou seja, imagine que você possui uma base de dados com 1000 registros de carros com suas respectivas informações incluindo essa marca mas esta base de dados está sendo ordenada em ordem alfabética pela marca.

1.3-A divisão de conjuntos.

Você faz a divisão no conjunto pegando os primeiros 700 registros para treinamento e os outros 300 restantes para teste você consegue enxergar um exemplo que tipo de problema essa divisão poderia gerar? O modelo preditivo queria aprender a classificação somente os carros cuja marca comece com a letra A até a letra R por exemplo, sendo que os carros que começam com a letra S, T,U ou V e as mais restantes podem ter características bem diferentes dos primeiros carros que entraram para o treinamento.

1.4-Observações do modelo.
É importante garantir que o modelo aprenda com todos os tipos de dados possíveis, por isso utilizamos aleatoriedade na hora da divisão do nosso “data sets” existem funções para isso na linguagem python que aprenderemos ainda neste curso, temos dois problemas que podem ocorrer na má divisão dos dados são dois extremos

1.5-Problemas que acontecem na má divisão dos dados.
Temos dois problemas que podem ocorrer na má divisão dos dados são dois extremos:

Overfitting: Temos um problema disso sobre sobreajuste, significa que o nosso modelo teve um excelente aprendizado nos dados de treinamento porém ele não soube aplicar esse aprendizado em dados novos como os dados de teste, nesse caso o algoritmo só decorou que deveria ser feito com os dados de treino e tentou aplicar a mesma técnica nos dados novos que são totalmente diferentes dos primeiros, quando isso ocorre dizemos que um modelo não tem capacidade de generalização.

Possíveis Soluções:

Primeiramente simplificar o modelo escolhendo um tipo de algoritmo mais simples ou então coletar mais dados se seu problema é de sobreajuste, a sua base de dados provavelmente é muito pequena, o problema de Overfitting geralmente acontece quando temos uma base de dados pequena.

Para detectar esse problema e para variar o quanto é preciso modelo é utilizado a validação cruzada que em inglês.

Na validação cruzada K-FOLD o conjunto de
dados é dividido em subconjuntos K, que também podemos chamar de fold para cada subconjunto aplicaremos treinamento e teste em dados diferentes veja o exemplo da imagem:

Iteration 1.	**Test.**	**Train.**	**Train.**	**Train.**	**Train.**
Iteration 2.	**Train.**	**Test.**	**Train.**	**Train.**	**Train.**
Iteration 3.	**Train.**	**Train.**	**Test.**	**Train.**	**Train.**
Iteration 4.	**Train.**	**Train.**	**Train.**	**Test.**	**Train.**
Iteration 5.	**Train.**	**Train.**	**Train.**	**Train.**	**Test.**

Temos cinco interações que são subconjuntos e cada um possui uma divisão diferente dos dados de teste e treinamento, os resultados serão diferentes em cada interação por fim a validação cruzada nos retorna para cada subconjunto, o score uma pontuação indicando uma performance de cada um a partir dessa pontuação você pode comparar modelos e identificar se o modelo se ajusta ou não aos seus dados.

Underfitting: Quando temos um problema de super ajuste significa que o desempenho do modelo já falhou no próprio treinamento ou seja ele não é capaz de identificar padrões nos dados de treinamento logo os dados de teste serão também comprometidos nesse caso o modelo não deverá ser aplicado.

Possíveis soluções: Escolher um algoritmo mais poderoso para aprender a relação entre os nossos dados, outra solução seria verificar se os atributos na base de dados estão representando bem o que você deseja classificar ou prever.

Bom pessoal, então essa foi a aula de hoje, não se esqueçam de concluir a aula de hoje resolvendo os exercícios disponibilizados aí no link, então é isso eu desejo uma boa tarde a todos e até a próxima.