Relatório 12 - Predição e a Base de Aprendizado de Máquina

Guilherme Loan Schneider

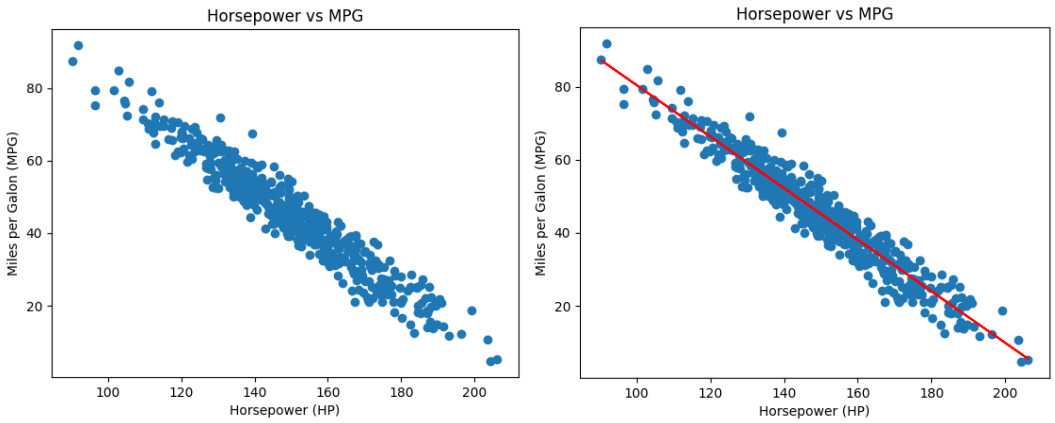
Descrição da atividade

A partir das aulas definidas no card “Predição e a Base de Aprendizado de Máquina”, foi possível compreender questões como modelos preditivos, que treinam em cima de dados e após isso são utilizados para fazer previsões em dados futuros, e muitas outras questões relacionadas a Machine Learning.

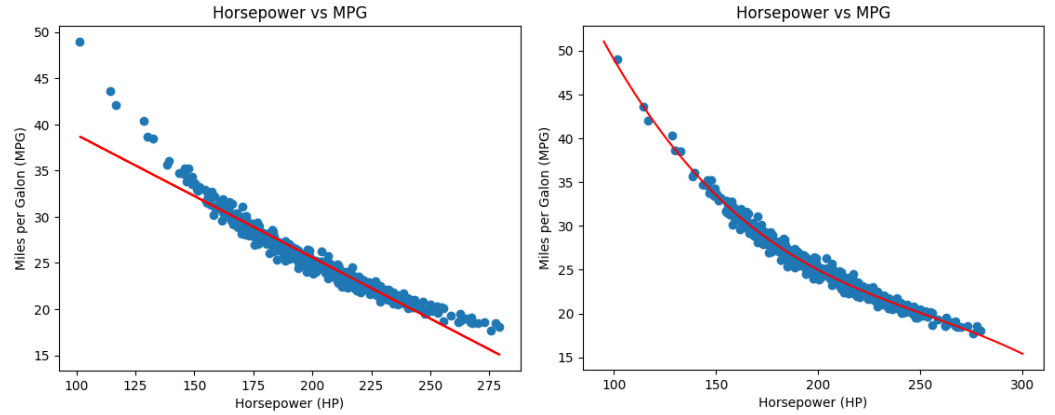
Predictive Models

O primeiro modelo preditivo é a Regressão Linear, que consiste em compreender correlações entre dados, e a partir dela definir uma linha que se assemelha àquele padrão de dados. Um bom score desse tipo de regressão é aquele que chega muito próximo de 1, preservando a variância dos valores originais.

No gráfico abaixo, é criada uma correlação entre o consumo de um veículo e a sua potência em cavalos.

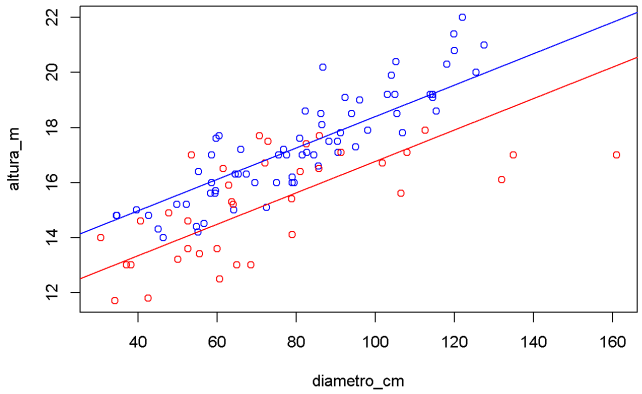


O segundo modelo segue o mesmo princípio, a Regressão Polinomial permite compreender correlações mais complexas (Imagem à direita), quando um segmento linear de reta não é suficiente para modelar bem o problema (Imagem à esquerda). Nesse modelo é importante definir o grau do polinômio, pois um polinômio de grau superior ao necessário, pode fazer com que o modelo perca a capacidade de fazer previsões.



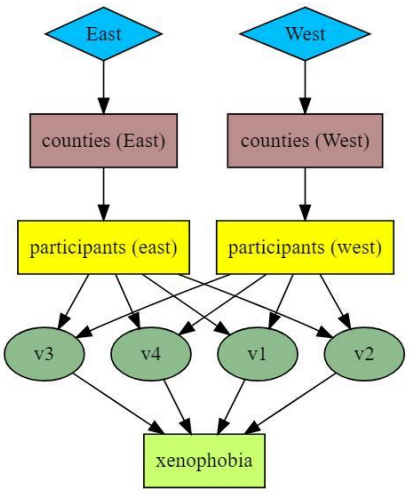
O terceiro modelo, chamado de Regressão Múltipla, é utilizado quando existe mais de uma variável que influencia determinada previsão, como o exemplo utilizado em aula, que o preço de um carro é influenciado por mais de um valor, como quantidade de cilindros, hodômetro e portas.

Na utilização desse tipo de regressão, o próprio algoritmo criará um sistema de pesos para determinado valor (quantidade de cilindros, hodômetro, portas), onde o primeiro possui o maior peso, sendo o determinante para aumentar o valor do carro, enquanto os outros viraram desconto no valor do carro (por algum motivo quanto mais portas, o valor diminui).



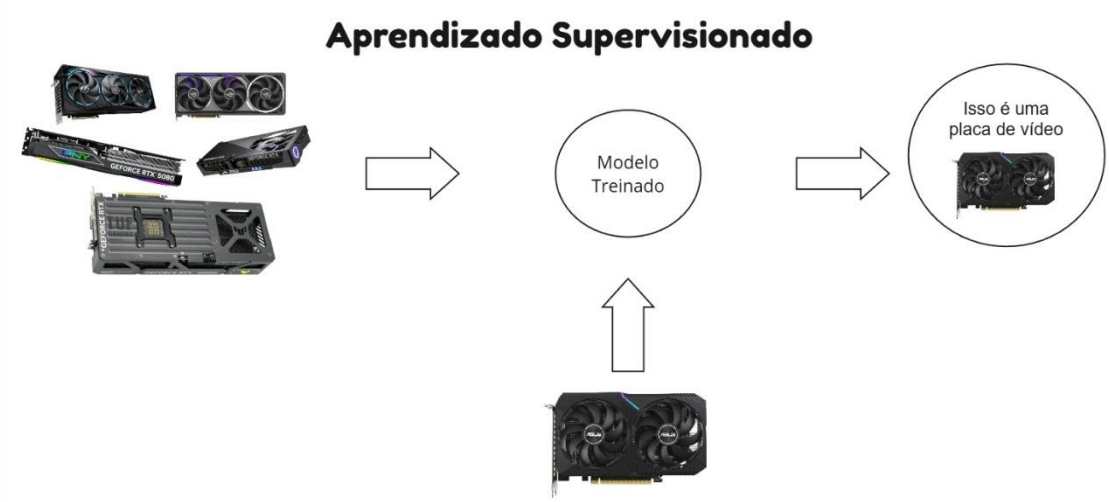
Por fim, temos o Modelo Multi-Nível, que é o mais complexo deles, onde parte-se do princípio que algo é influenciado em muitos níveis, resumidamente, é como se o bom desempenho de um carro dependesse de vários fatores, como o material empregado na sua construção, a sua altura do solo, que tipo de suspensão é utilizada, aerodinâmica do carro, dentre muitos outros fatores.

O grande problema destacado na aula, é conseguir fazer essa modelagem dos fatores influenciadores, por conta de que alguns deles podem afetar outros fatores, não é muito trivial entender se a posição dos espelhos retrovisores realmente impacta no desempenho do carro.

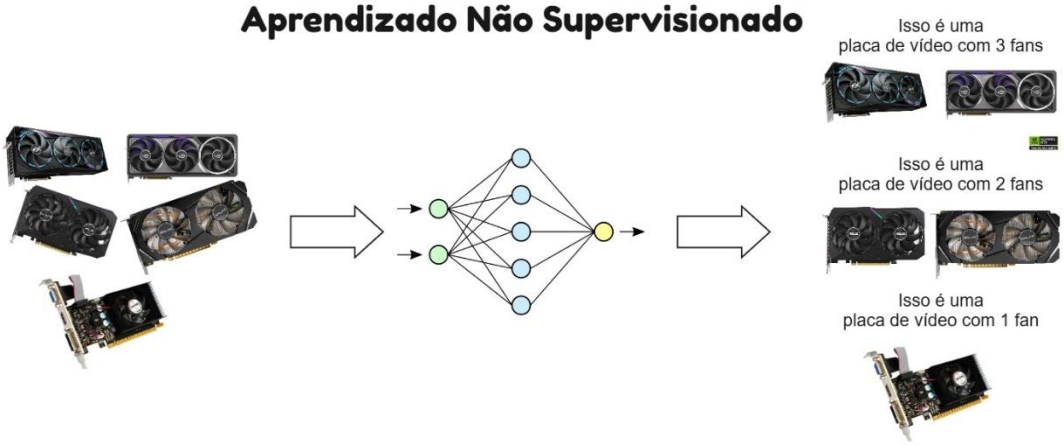


Machine Learning with Python

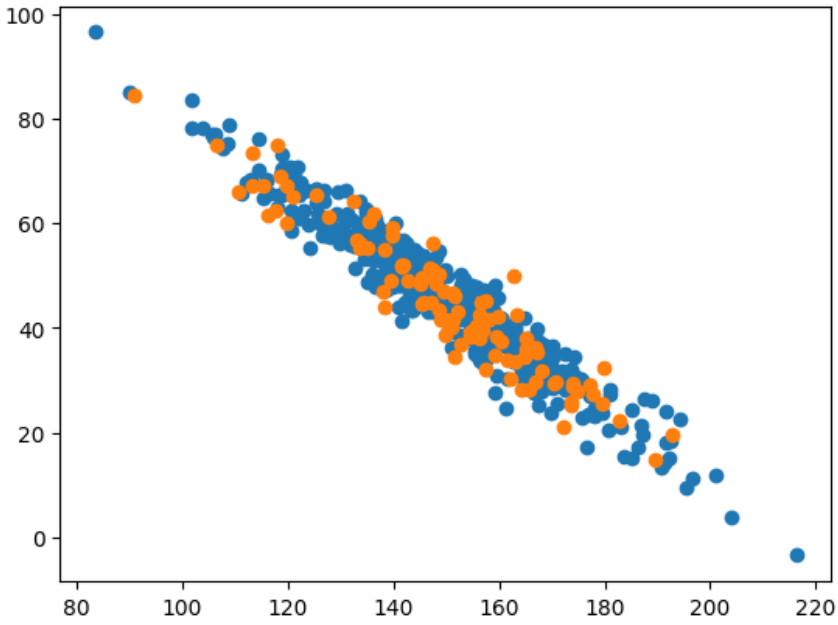
Aprendizado Supervisionado – Consiste em passar para um modelo, dados já classificados como corretos e incorretos, fazendo assim com que ao final, o modelo consiga fazer previsões em um tempo curto de treinamento.



Aprendizado Não Supervisionado – Esse tipo de aprendizado permite que o próprio modelo encontre padrões para aqueles dados, padrões esses que não são óbvios para humanos, ele faz isso por conta de que os dados são basicamente “jogados” para o modelo, e parte dele compreender o que essas informações significam. Geralmente esse modelo tem um sistema de pesos, que possui peso negativo quando não é para que o modelo faça aquilo, e positivo quando ele conclui uma resposta correta.



TrainTest – É utilizado para avaliar os modelos de predição, onde eles são treinados com 80% dos dados disponíveis, e os outros 20% são utilizados para verificar a capacidade do modelo prever dados, ajudando a evitar um caso de overfitting. É importante destacar que o conjunto de dados deve ser escolhido aleatoriamente para evitar viés. Na imagem abaixo, em azul, temos os dados utilizados para treinamento (80%), e em amarelo, os dados utilizados para teste (20%).



Método Bayesiano (Naive Bayes)

Esse método de Machine Learning é utilizado para quando se deseja analisar a probabilidade de algo acontecer, como por exemplo, a probabilidade de um email ser spam se existir a palavra “free”. É importante salientar que esse método assume que as palavras não possuem correlação entre as outras.

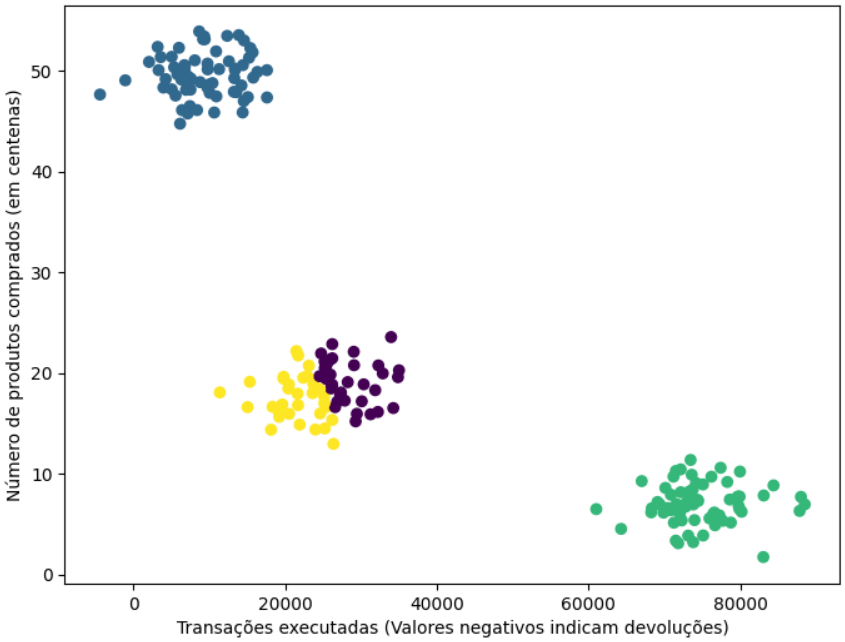
Na implementação desse método, utilizado para identificar emails sendo spam com a palavra “free”, as palavras contidas em um email foram transformadas em números e em seguida em uma matriz de contagem de palavras.

```
vectorizer = CountVectorizer()

# Essa função transforma o texto em uma matriz de contagem de palavras
# (as palavras são transformadas em números)
counts = vectorizer.fit_transform(data['message'].values)

classifier = MultinomialNB() # Cria um classificador Naive Bayes
targets = data['class'].values # Pega as classificações das mensagens
classifier.fit(counts, targets) # Treina o classificador
```

K-means Clustering

O K-means divide os dados em K grupos e em seguida analisa e marca os que estão mais próximos de uma determinada centróide. Deve-se destacar também que o valor de K não é algo trivial, mas a recomendação é que o valor dele seja aumentado até não haver mais grandes reduções no erro.

Na imagem abaixo, definiu-se 4 grupos com o número de transações executadas definidas aleatoriamente entre um valor de -100 mil e 100 mil, e o número de produtos comprados definida aleatoriamente entre 0 e 70.



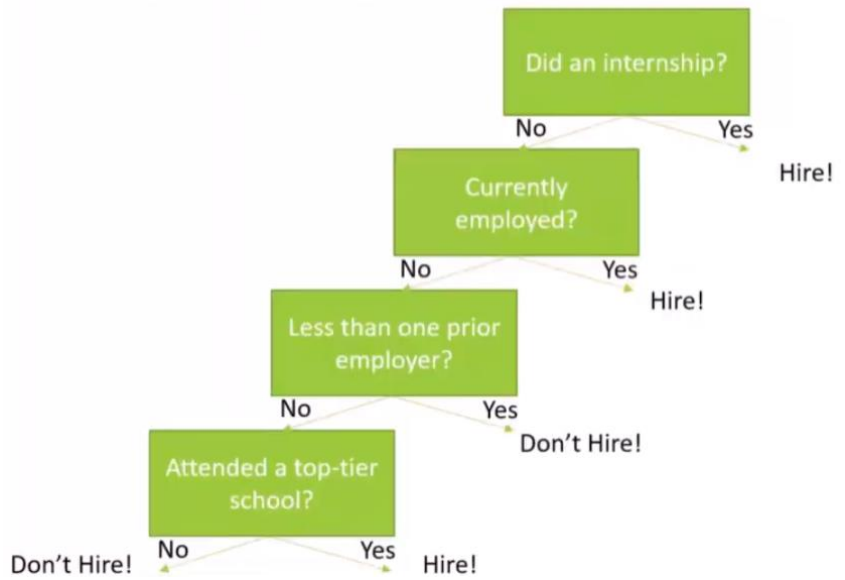
É um método de aprendizado não supervisionado que é usado em vários campos, como segmentação de imagem, segmentação de clientes e detecção de anomalias.

Entropia (Entropy)

A entropia é uma forma de quantificar a aleatoriedade de um conjunto de dados, como por exemplo, um estacionamento com 10 carros de 10 cores diferentes, o seu valor de entropia seria 1, que indica que os dados são totalmente diferentes entre si.

Árvores de Decisão (Decision Trees)

É um diagrama que representa as consequências de uma série de decisões. Elas são geralmente utilizadas para minimizar a entropia de um conjunto de dados, criando segmentações a partir da árvore de decisão.

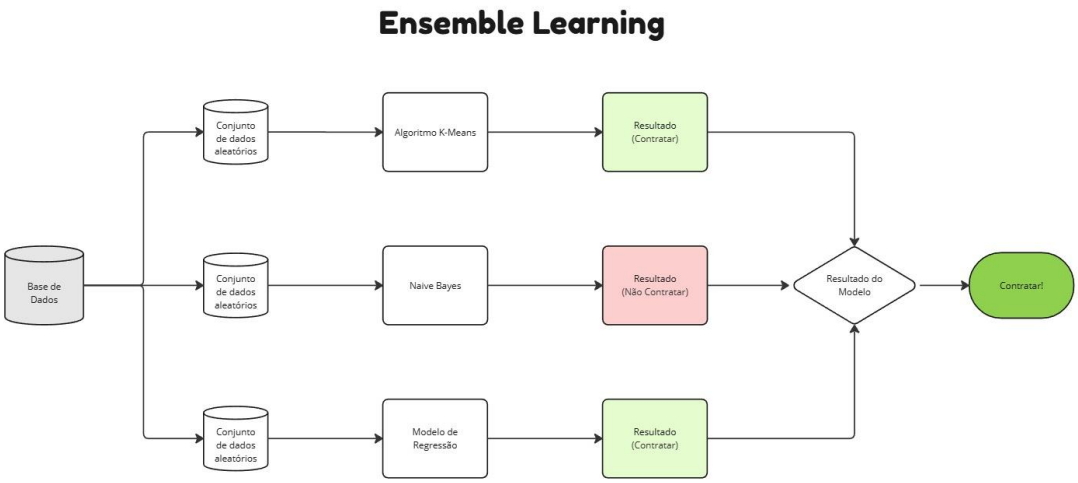


O exemplo de aula consiste na árvore de decisão acima voltada para contratar funcionários, onde cada aspecto levará a uma decisão. É importante destacar também

que dados categóricos devem ser convertidos em numéricos para poderem ser utilizados no algoritmo ID3 (algoritmo que constrói árvores de decisão a partir de conjuntos de dados).

Ensemble Learning

Esse tipo de treinamento consiste em utilizar modelos diferentes de treinamento para resolver o mesmo problema, ou seja, utilizar o K-means, Naive Bayes, Random Forests, dentre outros, em conjunto, e permitir que os algoritmos possam votar na melhor escolha.



Técnicas utilizadas nesse modelo:

a) Bagging - Treina múltiplos modelos em subconjuntos diferentes dos dados, com isso reduz o overfitting e melhora também a estabilidade do modelo.
b) Boosting - Treina modelos sequencialmente, onde cada novo modelo foca nos erros do anterior. Essa técnica reduz o overfitting e funciona para dados complexos.
c) Stacking - Combina diferentes tipos de modelos (como regressão, árvores de decisão e redes neurais). Pode capturar padrões mais complexos do que bagging e boosting, no entanto, por utilizar vários modelos pode ser difícil de ajustar.

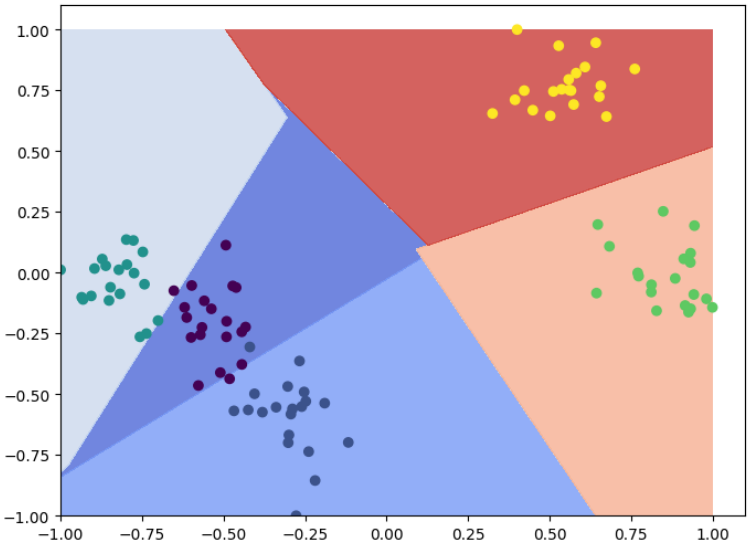
Support Vector Machines (SVM)

As SVMs encontram um hiperplano que separa as classes de dados, maximizando a distância entre elas, sendo muito utilizada quando existe muitas features a serem analisadas (planos além do 3D).

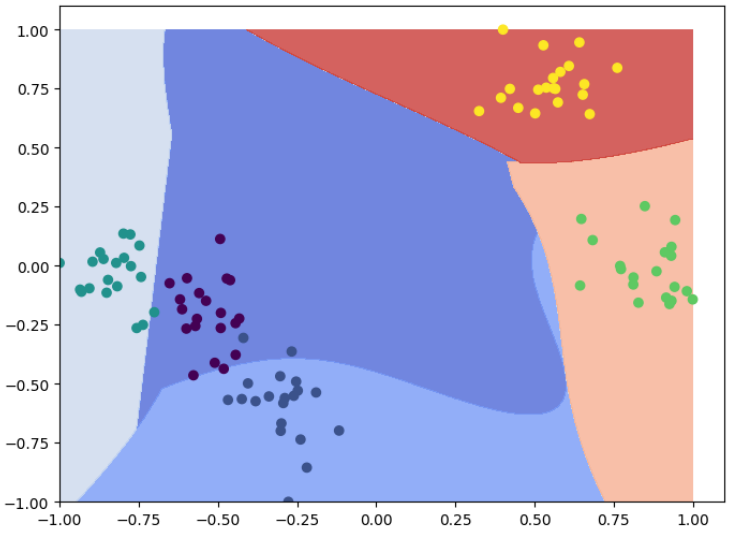
Além disso, utiliza uma técnica chamada “Kernel Trick”, que é usado quando os dados não são linearmente separáveis, transformando os dados para um espaço dimensional maior.

Existem vários tipos de kernels utilizados no SVM, como:

a) Linear - dados são linearmente separáveis;
b) Polinomial – captura interações mais complexas;
c) RBF – para dados complexos e não se sabe a relação entre eles;
d) Sigmoide – comportamento semelhante a uma rede neural.



Na imagem acima, utiliza-se o kernel do tipo linear, onde os dados foram separados por segmentos de reta, nesse caso, é possível perceber pontos “invadindo” áreas vizinhas, o que indica que esse pode não ser a melhor escolha de kernel. Já na imagem abaixo, utiliza-se o kernel polinomial, onde os planos são melhor representados.



Referencias

Machine Learning, Data Science and Deep Learning with Python (Seção 3-4)