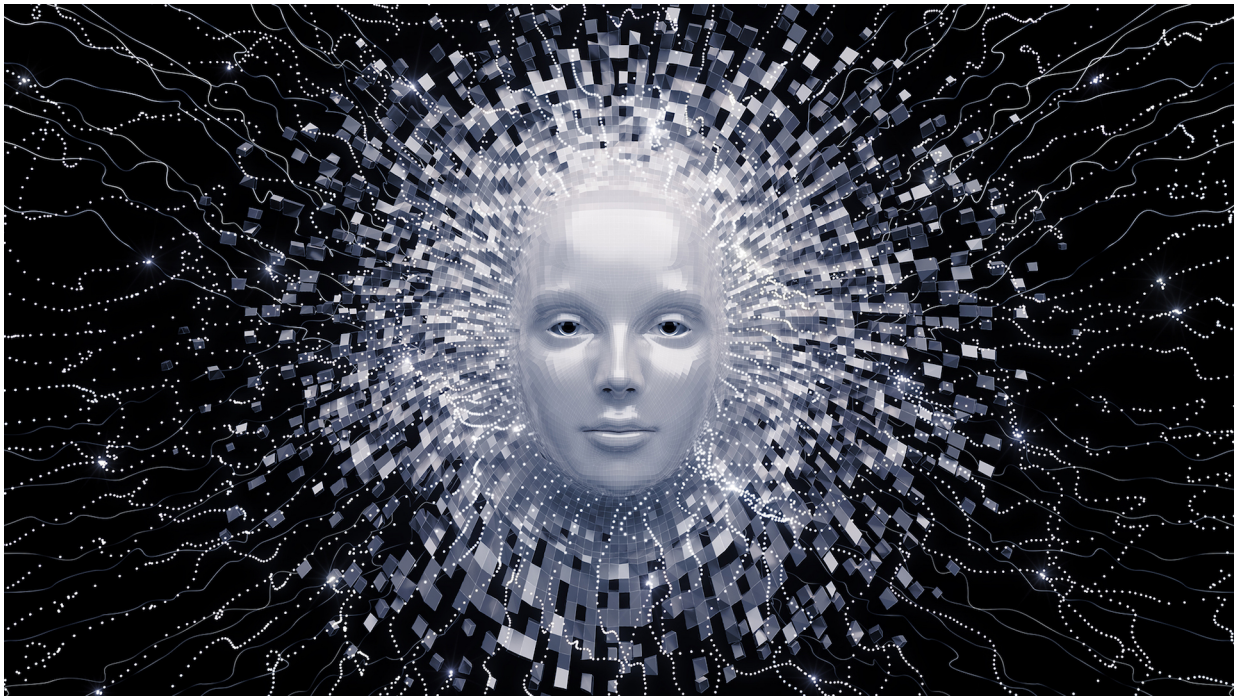
Máquina AZ Aprendizagem

Q & A

Todas as respostas para suas perguntas, seção por seção



Índice

1 Parte 1 - Pré-processamento de dados

1.1 A importação do conjunto de dados

Não consigo importar o conjunto de dados. Ele diz que o fi le não foi encontrado. O que devo fazer?
Python: Certifique-se que no Gerenciador de arquivos, você está na pasta que contém o fi le 'data.csv' que você deseja importar. Esta pasta é chamada de "pasta diretório de trabalho".
R: Certifique-se de que você defina o direito pasta diretório de trabalho como fazemos na palestra e que esta pasta diretório de trabalho contém o fi le 'data.csv'.

Qual é o ff rência di entre as variáveis independentes ea variável dependente?
As variáveis independentes são os dados de entrada que você tem, com cada você quer prever alguma coisa. Esse algo é a variável dependente.

Em Python, por que nós criamos X e Y separadamente?
Porque nós queremos trabalhar com arrays numpy, em vez de dataframes pandas. matrizes Numpy são o formato mais conveniente para trabalhar com ao fazer o pré-processamento de dados e modelos de aprendizagem de máquina de construção. Então nós criamos duas matrizes distintas, uma que contém nossas variáveis independentes (também chamadas as características de entrada), e outra que contém a nossa variável dependente (o que queremos prever).

Em Python, o que faz 'iLOC' exatamente?
Ele localiza a coluna pelo seu índice. Em outras palavras, usando 'iLOC' nos permite tirar colunas apenas tomando seu índice.

Em Python, o que faz' .values' exatamente fazer?
Ele retorna os valores das colunas que está a tomar (pelo seu índice) dentro de uma matriz Numpy. Isso é basicamente como X e Y se tornam matrizes Numpy.

Em R, por que não temos que criar matrizes?
Porque R funciona muito di ff erently do que Python. R contém ótimas ferramentas que permitem trabalhar diretamente e facilmente com dataframes.

1.2 Os dados em falta

Em Python, o que é o ff rência di entre fi t e transformar?
A parte fi t é usado para extrair algumas informações dos dados em que é aplicado o objeto (aqui, Imputer vai manchar os valores em falta e obter a média da coluna). Então, a parte transformar é usado para aplicar alguma transformação (aqui, Imputer irá substituir o valor em falta pela média).

Em R, por que usamos a função av () quando substituir os valores ausentes por uma média da coluna quando podemos fazê-lo muito mais simples desta forma:

```
dataset $ Age [is.na (dataset $ Age)] = média (dataset $ Age, na.rm = T)
```

Nós usamos a função ave () com o parâmetro FUN porque nos permite adicionar algumas opções no cálculo da média, como por exemplo, calculando a média das observações agrupadas por uma outra característica, ou calcular a média de subconjuntos, etc. Esta função pode ser muito útil se você quiser ser mais preciso ao substituir os valores ausentes pela média.

Está substituindo pela significar a melhor estratégia para lidar com valores em falta?

É uma boa estratégia, mas não o melhor sempre. Depende do seu problema de negócios, na forma como os dados são distribuídos e sobre o número de valores em falta. Se por exemplo você tem um monte de valores em falta, então significa a substituição não é a melhor coisa. Outras estratégias incluem imputação "mediano", "mais frequente" imputação ou imputação previsão. Previsão imputação é realmente outra grande estratégia que é recomendada, mas que eu não cobrir na parte 1 porque era muito avançada para fazê-lo na Parte 1. Esta estratégia fato requer a entender Parte 3 - Classi fi cação. Então, se você completou a Parte 3, aqui é a estratégia que é ainda melhor do que a imputação dizer: você levar o seu coluna recurso que contém os valores ausentes e você definir esta coluna recurso como a variável dependente, ao definir as outras colunas como variáveis independentes. Então você dividir o seu conjunto de dados em um conjunto de treinamento e um conjunto de teste onde o conjunto de treinamento contém todas as observações (as linhas), onde sua coluna recurso que você acabou de definir como a variável dependente não tem qualquer valor em falta eo conjunto de teste contém todas as observações onde sua coluna variável dependente contém os valores em falta. Em seguida, você executar um modelo fi cação classi (uma boa para esta situação é k-NN) para prever os valores ausentes no conjunto de teste. E eventualmente você substituir seus valores perdidos pelas previsões. Uma grande estratégia! Então você dividir o seu conjunto de dados em um conjunto de treinamento e um conjunto de teste onde o conjunto de treinamento contém todas as observações (as linhas), onde sua coluna recurso que você acabou de definir como a variável dependente não tem qualquer valor em falta eo conjunto de teste contém todas as observações onde sua coluna variável dependente contém os valores em falta. Em seguida, você executar um modelo fi cação classi (uma boa para esta situação é k-NN) para prever os valores ausentes no conjunto de teste. E eventualmente você substituir seus valores perdidos pelas previsões. Uma grande estratégia! Então você dividir o seu conjunto de dados em um conjunto de treinamento e um conjunto de teste onde o conjunto de treinamento contém todas as observações (as linhas), onde sua coluna recurso que você acabou de definir como a variável dependente não tem qualquer valor em falta eo conjunto de teste contém todas as observações or

1.3 Categórica de Dados

Em Python, o que os dois métodos 'fi t_transform' fazer?

Quando o método de 'fi t_transform ()' é chamada a partir da classe LabelEncoder (), ele transforma as cordas Categorias em inteiros. Por exemplo, ele transforma França, Espanha e Alemanha em 0, 1 e 2. Em seguida, quando o método 'fi t_transform ()' é chamada a partir da classe OneHotEncoder (), cria colunas separadas para cada di ff etiquetas erent com valores binários 0 e 1 . Essas colunas separadas são as variáveis dummy.

Em R, por que não criar manualmente as variáveis binárias como nós fazemos em Python

Porque eles são criados automaticamente quando usando o 'factor ()' função como fazemos no Palestra. Vamos ver visualmente que quando se inicia a regressão e classi fi cação.

1.4 Dividindo o conjunto de dados para o conjunto de treinamento conjunto e Teste

Qual é o ff rência di entre o conjunto de treinamento e conjunto de teste?

O conjunto de treinamento é um subconjunto de seus dados em que o modelo vai aprender a prever a variável dependente com as variáveis independentes. O conjunto de teste é o subconjunto cortesia do conjunto de treinamento, no qual você irá avaliar o seu modelo para ver se ele consegue prever corretamente a variável dependente com as variáveis independentes.

Por que dividir sobre a variável dependente?

Porque nós queremos ter bem distribuído valores da variável dependente no conjunto de treinamento e teste. Por exemplo, se nós só tinha o mesmo valor da variável dependente no conjunto de treinamento, o nosso modelo não seria capaz de aprender qualquer correlação entre as variáveis independentes e dependentes.

Scaling 1,5 Característica

Será que realmente tem que aplicar Scaling Característica nas variáveis binárias?

Sim, se você quer otimizar a precisão das suas previsões do modelo. Não, se você quiser manter o mais interpretação possível do seu modelo.

Quando devemos usar Normalização e Normalização?

Geralmente, você deve normalizar (normalização), quando os dados são normalmente distribuídos e escala (padronização) quando os dados não são normalmente distribuídos. Em caso de dúvida, você deve ir para a normalização. No entanto o que é comumente feito é que os dois métodos de escala são testados.

2 Part 2 - Regressão

2.1 Regressão Linear Simples

2.1.1 Regressão Linear Simples Intuição O que são exatamente os coeficientes coe FFI b_0 e b_1 na equação de regressão linear simples: 'Salário = $b_0 + b_1 \times$ Experiência'

b_0 é o salário que você começa com nenhuma experiência e b_1 é o aumento de salário por ano.

Por que tomar as rências di ff quadrados e simplesmente não os absolutos rências di ff?
Porque o quadrado di ff rências torna mais fácil para obter uma linha de regressão. Na verdade, de encontrar essa linha, precisamos calcular a derivada primeira da função de erro perda, e é muito mais difícil de calcular a derivada de valores absolutos do que os valores quadrados.

2.1.2 Regressão Linear Simples em Python

Por que não aplicar Scaling recurso no nosso simples modelo de regressão linear?
É simplesmente porque desde y é uma combinação linear das variáveis independentes, os coeficientes coe FFI pode adaptar sua escala de **colocar tudo na mesma escala. Por exemplo, se você tem duas variáveis independentes X_1 e X_2 e se y toma valores entre 0 e 1, X_1 toma valores entre 1 e 10 e X_2 toma valores entre 10 e 100, em seguida b_1 pode ser multiplicado por 0,1 e b_2 pode ser multiplicado por 0,01 para que y, $b_1 X_1$ e $b_2 X_2$ estão todos na mesma escala.**

O que significa 'regressor. Fi t (X_train, y_train)' faz exatamente?
O método fi t levará os valores de X_train e y_train e, em seguida, irá calcular os coeficientes coe ffi b_0 e b_1 da equação de regressão linear simples ($y = b_0 + b_1 x$) como pode ser visto na Palestra intuição. Esse é o objetivo desse método fi t aqui.

2.1.3 Regressão Linear Simples em R Qual é o valor-p?

Para entender o P-valor, precisamos começar por compreender a hipótese nula: a hipótese nula é a suposição de que os parâmetros associados às suas variáveis independentes são iguais a zero. Portanto, sob essa hipótese, suas observações são totalmente aleatórios, e não seguem um certo padrão. O valor P é a probabilidade de que os parâmetros associados às suas variáveis independentes têm certos valores diferentes de zero, dado que a hipótese nula é verdadeira. A coisa mais importante a ter em mente sobre a P-Value é que é uma medida estatística: quanto menor é o P-valor, mais estatisticamente significante é uma variável independente, que é a melhor preditor será.

Na última palestra R vemos uma área cinzenta em torno da linha azul. O que significa e como podemos obtê-lo?

A área cinzenta em torno da linha fi t é o intervalo de con fi ança. Você pode adicionar '+ geom_smooth (method = 'lm')' logo após 'geom_line (...)' para obter esses confiança limites.

Como obter os valores de p em Python como nós fazemos em R?
Você vai aprender como fazer isso na próxima seção sobre Regressão Linear Múltipla.

2.2 Regressão Linear Múltipla

2.2.1 Linear Múltipla Regressão Intuition

Quais são as várias hipóteses de regressão linear em mais detalhes?
Linearidade: Deve existir uma relação linear entre a variável dependente e as variáveis independentes. Scatterplots pode mostrar se há uma relação linear ou curvilínea.
Homoscedasticidade: Esta suposição afirma que a variância dos termos de erro é semelhante em todos os valores das variáveis independentes. Um lote de resíduos padronizados versus valores previstos podem mostrar-se os pontos são distribuídos igualmente em todos os valores das variáveis independentes.
Multivariada Normalidade: Regressão Linear Múltipla assume que os resíduos (o di ff rências entre o valor observado da variável dependente y e o valor previsto $\hat{}$ y são normalmente distribuídos.
Independência de erros: Regressão Linear Múltipla assume que os resíduos (o di ff rências entre o valor observado da variável dependente y e o valor previsto $\hat{}$ y são independentes.
Falta de multicolinearidade: Regressão Linear Múltipla assume que as variáveis independentes não são altamente correlacionadas entre si. Esta suposição é testado usando variância em valores inflação Factor (VIF).

Como é o cient coe ffi b_0 relacionada com a armadilha variável fictícia?
Desde a $D_2 = 1 - D_1$ em seguida, se você incluir tanto D_1 e D_2 você obtém:

$$\begin{aligned} y &= b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 D_1 + b_5 D_2 \\ &= b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_4 D_1 + b_5 (1 - D_1) \\ &= b_0 + b_5 + b_1 X_1 + b_2 X_2 + b_3 X_3 + (b_4 - b_5) D_1 \\ &= b^*_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b^*_4 D_1 \end{aligned}$$

com $b^*_0 = b_0 + b_5$ e $b^*_4 = b_4 - b_5$
Portanto, a informação da variável dummy redundante D_2 vai para a constante b_0.

2.2.2 Regressão Linear Múltipla em Python

Previmos os resultados de um conjunto de observações (o conjunto de teste). Como é que vamos fazer o mesmo para uma única observação?

Vamos dizer que esta observação tem as seguintes características: Valor Feature 1, Valor Recurso 2, ... e Valor função M (m variáveis independentes). Em seguida, o código para obter o resultado previsto seria o seguinte:

```
y_pred = regressor.predict (np.array ([[Valor Feature 1, ..., Característica Valor m]]))
```

Como implementar a eliminação retrógrada automática em Python?
Você pode implementá-lo desta maneira:

```
importar statsmodels.formula.api como backwardElimination sm def
(x, sl):
        numVars = len (X [0])
        para i na gama (0, numVars):
                regressor_OLS = sm.OLS ( y , x) .fit ()
                maxVar = max (regressor_OLS.pvalues) .astype (flutuador)
                E se maxVar> sl:
                        para j na gama (0, numVars - i):
                                E se ( regressor_OLS.pvalues [j] .astype (flutuador) == maxVar):
                                        x = np. excluir ( x, j, 1)
        regressor_OLS.summary ()
        Retorna X
```

```
SL = 0,05
X_opt = X [:, [0, 1, 2, 3, 4, 5]]
X_Modeled = backwardElimination (X_opt, SL)
```

2.2.3 Regressão Linear Múltipla em R

Previmos os resultados de um conjunto de observações (o conjunto de teste). Como é que vamos fazer o mesmo para uma única observação?

Você primeiro precisa criar uma nova variável 'single_observation' com os valores de entrada e definir esta variável como uma trama de dados:

```
single_observation = data.frame (RDSpend = 10,000,
                                 Administração = 20000, Marketing.Spend =
                                 30000, Estado = 1)
```

Então você pode simplesmente prever o resultado desta única observação exatamente como fizemos com o conjunto de teste, por apenas substituindo 'test_set' por 'single_observation':

```
y_pred = prever (regressor, newdata = single_observation)
```

Como implementar a eliminação retrógrada automática em R?

Você pode implementá-lo desta maneira:

```
backwardElimination <- função (x, sl) {
        numVars = comprimento ( x)
        para ( i no c (1: numVars)) {
            regressor = lm (fórmula = lucro ~., dados = x)
            maxVar = max (coef (resumo (regressoras)) [C (2: numVars), "Pr (> | t |)" ])
            E se ( maxVar> sl) {
                j = que (coef (resumo (regressoras)) [C (2: numVars), "Pr (> | t |)" ] == maxVar) x = x [, - j]}


            numVars = numVars - 1}

        Retorna ( Resumo (regressor))}


    SL = 0,05
    conjunto de dados do conjunto de dados = [, c (1,2,3,4,5)]
    backwardElimination (conjunto de dados, SL)
```

2,3 regressão polinomial

2.3.1 Regressão Polinomial Intuition

Regressão Polinomial é um modelo não linear linear ou?
Isso depende do que você está se referindo. Polinômio de regressão é linear nos coeficientes coe FFI uma vez que não tem qualquer poder dos coeficientes coe FFI (todos os coeficientes coe FFI são elevado à potência de 1: b_0, b_1, ..., b_n). No entanto, polinomial é uma função não linear da entrada *x,* uma vez que temos os dados levantados a vários poderes: x (alimentação 1), X_2 (poder 2), ..., X_n (poder n). É assim que podemos ver também o polinômio de regressão como um modelo não linear. Além de fato, Regressão Polinomial é apropriado quando os dados são não distribuídos de forma linear (ou seja, você não pode fi t uma linha reta entre y e x).

2.3.2 Regressão Polinomial em Python

Por que não aplicar Scaling recurso no nosso modelo de regressão polinomial?
É simplesmente porque, desde que y é uma combinação linear de X e X_2, os coeficientes coe FFI pode adaptar sua escala de colocar tudo na mesma escala. Por exemplo, se y toma valores entre 0 e 1, X toma valores entre 1 e 10 e X_2 toma valores entre 1 e 100, em seguida b_1 pode ser multiplicado por 0,1 e b_2 pode ser multiplicado por 0,01 para que y, $b_1 X_1$ e $b_2 X_2$ estão todos na mesma escala.

Como é que vamos achar o melhor grau?
A principal forma de encontrando um bom fi t é traçar o modelo e ver o que ele se parece visualmente. Você simplesmente testar vários graus e você ver qual deles oferece o melhor fi t. A outra opção é encontrar o menor erro root mean square (RMSE) para o seu modelo, mas, nesse caso, ter cuidado para não sobre fi t os dados.

Por que nós temos que criar uma segunda regressor linear 'lin_reg_2'? Não poderíamos ter usado 'lin_reg' diretamente?

Não, porque 'lin_reg' já está fi enquadrados para X e y e agora queremos encaixa um novo modelo linear para X_p oly e y. Então nós temos que criar um novo objeto regressor. Um must importante que o método fi t aqui fi nds o cient ffi coe entre as variáveis independentes e a variável dependente. Por isso, desde 'lin_reg' já tem os coeficientes coe FFI de correlação entre X e y, 'Lin_reg_2' tem de ser criado para obter alguns novos coeficientes coe FFI de correlações entre X_p oly e y.

2.3.3 regressão polinomial em R

Em R, eu tenho que criar manualmente as colunas di ff erent para cada função polinomial? E se há muitas características polinomiais?

Você muito raramente terá que criar mais de 4 características polinomiais, caso contrário você iria superar fi tting, que deve ser absolutamente evitado em Machine Learning. Assim, mesmo se você tem que criá-los manualmente, ele nunca vai tomar muito do seu tempo. Além disso, você pode usar o modelo.

Como é que vamos achar o melhor grau? (Fazer esta pergunta novamente no caso alguns alunos só fazem R
A principal forma de encontrando um bom fi t é traçar o modelo e ver o que ele se parece visualmente. Você simplesmente testar vários graus e você ver qual deles oferece o melhor fi t. A outra opção é encontrar o menor erro root mean square (RMSE) para o seu modelo, mas, nesse caso, ter cuidado para não sobre fi t os dados.

2.4 SVR

2.4.1 SVR Intuição Quando devo usar

SVR?
Você deve usar SVR se um modelo linear como regressão linear não encaixa muito bem os seus dados. Isso significa que você está lidando com um problema não linear, onde os dados não está distribuído de forma linear. Portanto, nesse caso SVR poderia ser uma solução muito melhor.

Eu não entendia o Palestra intuição. Estou em sarilhos?
De modo nenhum. SVR é um modelo abstrato bonito e além disso não é que comumente usados. O que você deve sim entender é o modelo SVM, que você vai ver na Parte 3 - Classi fi cação. Em seguida, uma vez que você entender o modelo SVM, você vai obter uma melhor compreensão do modelo SVR, uma vez que o SVR é simplesmente o SVM para a regressão. No entanto, queria incluir SVR neste curso para dar-lhe uma opção extra no seu kit de ferramentas de aprendizagem automática.

2.4.2 SVR em Python

Por que precisamos 'sc_Y.inverse_transform'?
Precisamos do método inverse_transform para voltar à escala original. Na verdade nós aplicamos função Dimensionamento então temos esta escala em torno de 0 e se fazer uma previsão sem inversing a escala que vai ter o escalado previu salário. E, claro, queremos que o salário real, não aquela em escala, por isso temos de usar 'sc_Y.inverse_transform'. Também o que é importante entender é que 'transformar' e 'inverse_transform' são métodos emparelhados.

2.4.3 RVS em R

Por que não aplicar recurso escala como fizemos explicitamente em Python
Isso porque, em função SVM () de R, os valores são automaticamente escalado.

podemos selecionar os mais signi fi cativos variáveis graças ao valor-p, como fizemos em R antes?
Você não poderia usar p-valor, porque SVR não é um modelo linear, e p-valores aplicam-se apenas aos modelos lineares. Portanto apresentam selecção está fora de questão. Mas você poderia fazer extração de características, que você vai ver na Parte 9 - dimensionalidade Redução. Que você pode aplicar para árvores de decisão, e que irá reduzir o número de suas características.

2.5 Árvore de Decisão Regressão

2.5.1 Árvore de Decisão Regressão Intuição Como o algoritmo de dividir

os pontos de dados?
Ele usa diminuição do desvio padrão das previsões. Em outras palavras, o desvio padrão é reduzido direita depois de uma separação. Assim, a construção de uma árvore de decisão é toda sobre encontrando o atributo que retorna a maior redução desvio padrão (ou seja, os ramos mais homogéneos).

Qual é o ganho de informação e como ele funciona em árvores de decisão?
O ganho de informação na árvore de decisão de regressão é exatamente a Redução Desvio Padrão estamos à procura de alcançar. Calculamos por quanto o desvio padrão diminui após cada divisão. Porque quanto mais o desvio padrão é diminuído depois de uma separação, as mais homogêneas os nós filho será.

Qual é a Entropia e como ele funciona em árvores de decisão?
A entropia mede a desordem num conjunto, aqui em parte resultante a partir de uma divisão. Assim, quanto mais homogênea é seus dados em uma parte, menor será a entropia. Quanto mais você tem splits, quanto mais você tem chance de fi partes nd em que os seus dados são homogêneos, e, portanto, menor será a entropia (próximo a 0) por estas bandas. No entanto, você ainda pode achar alguns nós onde os dados não é homogênea, e, portanto, a entropia não seria tão pequena.

2.5.2 Árvore de Decisão Regressão em Python Será que uma árvore de

decisão faz muito sentido em 1D?
Nem por isso, como vimos na parte prática desta seção. Em 1D (ou seja, uma variável independente), a árvore de decisão tende claramente a mais fi t os dados. A árvore de decisão seria muito mais relevante na dimensão mais elevada, mas tenha em mente que a implementação que fizemos aqui em 1D seria exatamente o mesmo em dimensão superior. Portanto, você pode querer manter esse modelo em sua caixa de ferramentas, no caso você está lidando com um espaço dimensional mais elevado. Isso vai realmente ser o caso na Parte 3 - Classi fi cação, onde usaremos Árvore de Decisão para Classi fi cação em 2D, o que você vai ver acaba por ser mais relevante.

2.5.3 Árvore de Decisão Regressão em R

Por que temos di resultados ff erent entre Python e R?
O erência di ff é provavelmente devido à divisão aleatória de dados. Se nós fizemos uma validação cruzada (ver Parte 10) em todos os modelos em ambas as línguas, então você provavelmente obter uma precisão média similar. Dito isto, nós recomendamos mais usando Python para árvores de decisão, uma vez que o modelo é um pouco melhor implementado em Python.

É a Árvore de Decisão apropriado aqui?
Aqui neste exemplo, podemos ver claramente que a curva fi tting é uma escada com grandes lacunas nas descontinuidades. Esse modelo de regressão árvore de decisão não é, portanto, o mais adequado, e que é porque temos apenas um variáveis independentes que tomam valores discretos. Então, o que aconteceu é que a previsão foi feita na parte inferior do gap em Python, e feita na parte superior da lacuna na R. E uma vez que a diferença é grande, que faz uma grande rência di ff. Se tivéssemos muito mais observações, tendo os valores com mais continuidade (como com um
0,1 passo), as lacunas seria menor e, portanto, as previsões em Python e R longe um do outro.

podemos selecionar os mais signi fi cativos variáveis graças ao valor-p, como fizemos em R antes?
Você não poderia usar p-valor, porque Árvore de Decisão não é um modelo linear, e p-valores aplicam-se apenas aos modelos lineares. Portanto apresentam selecção está fora de questão. Mas você poderia fazer extração de características, que você vai ver na Parte 9 - dimensionalidade Redução. Que você pode aplicar para árvores de decisão, e que irá reduzir o número de suas características.

2.6 Aleatório Floresta Regressão

2.6.1 aleatória Floresta Regressão Intuition

Qual é a vantagem e desvantagem de aleatórios Florestas comparação com árvores de decisão? Vantagem: Random florestas podem dar-lhe um melhor poder preditivo de Árvores de Decisão.
Drawback: Árvore de Decisão vai lhe dar mais do que interpretability aleatórios Florestas, porque você pode traçar o gráfico de uma árvore de decisão para ver as separações di ff erent que levam à previsão, como visto no Palestra intuição. Isso é algo que você não pode fazer com aleatórios Florestas.

Quando usar aleatória Floresta e quando usar os outros modelos?
A melhor resposta para essa pergunta é: julgá-los todos!
Na verdade, graças aos modelos que só levará 10 minutos para tentar todos os modelos, o que é muito pouco em comparação com o tempo dedicado às outras partes de um projeto de ciência de dados (como dados de pré-processamento, por exemplo). Então, basta não ter medo de experimentar todos os modelos de regressão e comparar os resultados (por meio de validação cruzada que veremos na Parte 10). Isso é que lhe deu os modelos máximas neste curso para você ter no seu kit de ferramentas e aumentar a sua chance de conseguir melhores resultados.

No entanto, em seguida, se você quiser alguns atalhos, aqui estão algumas regras de polegares para ajudá-lo a decidir qual modelo usar:

Primeiro, você precisa figura se o seu problema é linear linear ou não. Você vai aprender como fazer isso na Parte 10 - Selecção do modelo. Então: Se o seu problema é linear, você deve ir para Regressão Linear Simples, se você só tem uma característica, e Regressão Linear Múltipla, se você tem várias características. Se o seu problema é não linear, você deve ir para o polinômio de regressão, SVR, Árvore de Decisão ou aleatória Forest. Então qual delas você deve escolher entre esses quatro? Que você vai aprender na Parte 10 - Selecção do modelo. O método consiste na utilização de uma técnica muito relevante que avalia o seu desempenho modelos, chamado k-Fold Cruz Validação e, em seguida, escolher o modelo que apresenta os melhores resultados. Sinta-se livre para pular diretamente para a Parte 10 se você já quer aprender como fazer isso.

2.6.2 aleatória Floresta Regressão em Python Como posso saber

quantas árvores devo usar?
Em primeiro lugar, eu recomendaria para escolher o número de árvores através da experimentação. Geralmente, leva menos tempo do que nós pensamos que a fi gura um melhor valor por ajustes e ajustar seu modelo manualmente. Isso é realmente o que fazemos em geral, quando nós construímos um modelo de Aprendizado de Máquina: nós fazê-lo em vários tiros, experimentando vários valores de hiperparâmetros como o número de árvores. No entanto, também sabemos que na Parte 10 vamos cobrir k-Fold Cruz Validação e grade Search, que são técnicas poderosas que você pode usar para fi nd o valor ideal de um hiperparâmetro, como aqui o número de árvores.

2.6.3 aleatória Floresta de regressão em R

Como é que vamos decidir quantas árvores seria suficiente para obter um resultado relativamente precisas?
Mesmo que em Python, você pode encontrar um número relevante de árvore através da experimentação e ajustes manual. Você deve usar árvores suficientes para obter uma boa precisão, mas você não deve usar muitas árvores porque isso poderia causar mais fi tting. Você vai aprender a achar o número ideal de árvores na seção de primeiros da Parte 10 - Selecção do modelo. É feito com uma técnica chamada de ajuste dos parâmetros (Grid Pesquisa com k-Fold Validação Cruzada).

Por que temos di resultados ff erent entre Python e R?
O erência di ff é provavelmente devido à divisão aleatória de dados. Se nós fizemos uma validação cruzada (ver Parte 10) em todos os modelos em ambas as línguas, então você provavelmente obter uma precisão média similar.

podemos selecionar os mais signi fi cativos variáveis graças ao valor-p, como fizemos em R antes?
Você não poderia usar p-valor, porque aleatórios Florestas não são modelos lineares, e p-valores aplicam-se apenas aos modelos lineares. Portanto apresentam selecção está fora de questão. Mas você poderia fazer extração de características, que você vai ver na Parte 9 - dimensionalidade Redução. Que você pode aplicar a aleatórios Florestas, e vai reduzir o número de suas características.

2.7 Avaliação de Modelos de regressão Desempenho

Eu entendo que, a fim de avaliar vários modelos lineares eu posso usar o Rsquared ajustado ou a matriz de Pearson. Mas como posso avaliar modelos de regressão polinomial ou modelos aleatória Floresta de regressão que não são lineares?

Você pode avaliar modelos de regressão polinomial e modelos de regressão florestais aleatório, calculando a "média de Squared Resíduos" (a média dos erros ao quadrado). Você pode calcular isso facilmente com uma função de soma ou usando um loop for, pelo cálculo da soma do quadrado di ff rências entre os resultados previstos e os resultados reais, sobre todas as observações do conjunto de teste.

Você realmente não poderia aplicar Eliminação Backward para modelos regressão polinomial e aleatória Floresta regressão porque esses modelos não têm coeficientes coe FFI combinadas em uma equação de regressão linear e, portanto, não têm valores de p.

No entanto, na Parte 9 - dimensionalidade Redução, vamos ver algumas técnicas como a Eliminação para trás, para reduzir o número de recursos, com base na função Seleção & Feature Extraction. O curso ainda está em desenvolvimento e agora estamos trabalhando na Parte 8 - aprendizagem profunda, mas Parte 9 - dimensionalidade Redução sairá rapidamente depois.

Quais são Baixo / Viés / Desvio alta em Machine Learning?
Low Viés é quando suas previsões do modelo são muito próximos dos valores reais. Alta Viés é quando
suas previsões do modelo estão longe dos valores reais.
Baixa Variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação

não vai variar muito.

Alta variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação irá variar muito.

O que você deseja obter quando você constrói um modelo é: Low preconceito e baixa variância.

Em Python, como implementar Eliminação Backward automatizado com Ajustado R-Squared?

Vamos dar novamente o problema da regressão linear múltipla, com 5 variáveis independentes. Eliminação Backward Automated incluindo Ajustado R Squared pode ser implementado desta maneira:

```
importar statsmodels.formula.api como backwardElimination sm def
(x, SL):
        numVars = len (X [0])
        temp = np.zeros ((50,6)). (astype int )
5       para i na gama (0, numVars):
                regressor_OLS = sm.OLS ( y , x) .fit ()
                maxVar = max (regressor_OLS.pvalues) .astype (flutuador) adjR_before =
                regressor_OLS.rsquared_adj.astype (flutuador)
                E se maxVar> SL:
10                      para j na gama (0, numVars - i):
                                E se ( regressor_OLS.pvalues [j] .astype (flutuador) == maxVar):
                                        temp [:, j] = x [:, j] X = np. excluir ( X, J, 1) = tmp_regressor
                                        sm.OLS ( y , x) .fit ()

15                                      adjR_after = tmp_regressor.rsquared_adj.astype (flutuador)
                                        E se ( adjR_before> = adjR_after):
                                                x_rollback = np.hstack ((x, temp [:, [0, j]])) x_rollback = np. excluir ( x_rollback, j, 1)

                                                impressão ( regressor_OLS.summary ())
20                                              Retorna x_rollback
                                        outro :
                                                continuar
        regressor_OLS.summary ()
        Retorna X
25
SL = 0,05
X_opt = X [:, [0, 1, 2, 3, 4, 5]] X_Modeled = backwardElimination (X_opt, SL)
```

Como obter o Ajustado R-Squared em R?

Vamos denotar a variável dependente por DV, e as variáveis independentes por IV1, IV2, etc. Em seguida, o código R para obter o Ajustado R-Squared é o seguinte:

```
resumo (lm (DV ~ IV1 + IV2 + ..., conjunto de dados)) $ adj.r.squared
```

3 Parte 3 - Classi fi cação

3.1 Regressão Logística

3.1.1 Regressão Logística Intuition

Regressão Logística é um modelo não linear linear ou?
É um modelo linear. Você irá visualizar isso no final da seção ao ver que separador de classi fi de er é uma linha reta.

Quais são as hipóteses de regressão logística?
Primeiro, regressão logística binária requer a variável dependente para ser binários e ordinal de regressão logística requer a variável dependente para ser ordinal.
Em segundo lugar, a regressão logística exige as observações a serem independentes um do outro. Em outras palavras, as observações não deve vir a partir de medições repetidas ou dados combinados.
regressão terceiro lugar, logística exige que haja pouca ou nenhuma multicolinearidade entre as variáveis independentes. Isto significa que as variáveis independentes não devem ser demasiado altamente correlacionadas entre si. Em quarto lugar, a regressão logística assume linearidade de variáveis independentes e as probabilidades de log. Embora essa análise não requer as variáveis dependentes e independentes para ser relacionado linearmente, é necessário que as variáveis independentes são linearmente relacionada com as chances de log.

Regressão Logística pode ser usado para muitas variáveis independentes, bem como?
Sim, regressão logística pode ser usado para tantas variáveis independentes como você quer. No entanto estar ciente de que você não será capaz de visualizar os resultados em mais de 3 dimensões.

3.1.2 Regressão Logística em Python que faz o método fi t

fazer aqui?
O método fi t, basicamente treinar o modelo de regressão logística sobre os dados de treinamento. Por isso, irá calcular e obter os pesos (coeficientes coe FFI) do modelo de regressão logística (ver o Palestra intuição) para que determinado conjunto de dados de treinamento composto por X_train e y_train. Em seguida, logo depois que recolhe os pesos / coe coeficientes FFI, você tem um modelo de regressão logística totalmente treinados em seus dados de treinamento, e pronto para prever novos resultados graças ao método de prever.

Previmos os resultados de um conjunto de observações (o conjunto de teste). Como é que vamos fazer o mesmo para uma única observação, para prever um único resultado?
Vamos dizer que esta observação tem as seguintes características: Idade = 30, o salário estimado = 50000. Em seguida, o código para obter o resultado previsto seria o seguinte (note como não devemos esquecer a escala que única observação primeiro):

```
y_pred = classifier.predict (sc_X.transform (np.array ([[20, 50000]])))
```

Confusão Matrix é a melhor maneira de avaliar o desempenho do modelo?
Não, ele só lhe dá uma idéia de quão bem o seu modelo pode executar. Se você receber uma boa matriz de confusão com alguns erros de previsão sobre o conjunto de teste, então há uma chance de que seu modelo tem um bom poder preditivo. No entanto, a forma mais relevante para avaliar o seu modelo é através de K-Fold validação cruzada, que você vai ver na Parte
10. Trata-se de avaliar o seu modelo em vários conjuntos de testes (chamados os conjuntos de validação), para que possamos ter certeza de que não ter sorte em um conjunto único teste. Hoje a maioria dos dados Cientistas ou AI Developers avaliar o seu modelo através de K-Fold Validação Cruzada. No entanto, a técnica é uma di ff erent assunto, então eu preferia deixá-lo para a Parte 10 depois de cobrir todos os modelos di ff erent.

Como re-transformar idade e salário de volta à sua escala original?

Você pode re-transformar idade e salário usando o método inverse_transform (). Você levá-lo scaler objeto "sc" e você aplicar o método inverse_transform () sobre ele desta maneira:

```
X_train = sc.inverse_transform (X_train) X_test = sc.inverse_transform
(X_test)
```

Como a fazer o mesmo classi fi cação quando as variáveis dependentes tem mais de duas classes?

Vamos explicar como fazê-lo com três classes.

Primeiro, ele ainda é o mesmo código para fazer as previsões. Além embora seja baseado no Um contra todos método, você não precisa mesmo de criar variáveis dummy de sua variável dependente. Quanto às variáveis independentes, você só precisa aplicar LabelEncoder seguido por OneHotEncoder. Ele vai criar suas variáveis dummy, qualquer que seja o número de categorias suas variáveis categóricas ter. E, eventualmente, você simplesmente precisa adicionar uma cor da seguinte maneira:

```
# Visualizar os resultados conjunto de treinamento
de matplotlib.colors importar ListedColormap X_set, y_set = X_train, y_train

X1, X2 = np.meshgrid (np.arange (start = X_set [:, 0] .min () - 1,
                                 parar = X_set [:, 0] .max () + 1, passo = 0,01),

                      np.arange (start = X_set [:, 1] .min () - 1,
                                 parar = X_set [:, 1] .max () + 1, passo = 0,01))

plt.contourf (X1,
              X2,
              classifier.predict (np.array ([X1.ravel (), X2.ravel ()]). t) .reshape (X1.shape), alfa = 0,75,

              cmap = ListedColormap (( 'vermelho' , 'verde' , 'azul' )))
plt.xlim (X1.min (), X1.max ())
plt.ylim (X2.min (), X2.max ())
para i, j em enumerar (np.unique (y_set)): plt.scatter (X_set [y_set == j, 0],

                                                        X_set [y_set == j, 1],
                                                        c = ListedColormap (( 'vermelho' , 'verde' , 'azul' )) (I), label = j)
plt.title ( 'Regressão Logística (conjunto de treinamento)' ) Plt.xlabel ( 'Era' ) Plt.ylabel ( 'Salário
estimado' ) Plt.legend ()


plt.show ()
```

Como você pode ver neste código, eu só adicionou a cor 'azul', que eu tinha que fazer porque desde agora temos três classes, então np.unique (y_set) torna-se 3.

Você pode explicar o método meshgrid () em mais detalhes e alguns de seus usos?

No método meshgrid (), você digitar dois argumentos: primeiro argumento é o valores do intervalo de coordenadas x em sua grade. Em segundo lugar está os valores do intervalo das coordenadas y em sua grade.

Então, vamos dizer que estes argumentos 1º e 2º são, respectivamente, [-1, + 1] e [0,10], então você vai ter uma grade onde os valores vão de [-1, + 1] no eixo-x e [0,10] no eixo dos y.

Você pode explicar o método contourf () em mais detalhes e alguns de seus usos?
Antes de utilizar o método contourf, você precisa para construir uma grade. Isso é o que fazemos na linha logo acima na construção X1 e X2.

Em seguida, o método contourf () leva vários argumentos:
1. Os valores do intervalo de coordenadas x de sua grade,
2. Os valores do intervalo das coordenadas y de sua grade,
3. Uma linha fi tting (ou curva) que serão plotados neste grade (marcamos esta linha fi tting usando a função de prever porque esta linha são as previsões contínuas de nosso modelo),
4. Em seguida, o resto são argumentos opcionais como as cores para traçar regiões de cores erent di ff. As regiões serão separadas por esta linha tting fi, que é, de facto, a linha de contorno.

3.1.3 Regressão Logística em R

Previmos os resultados de um conjunto de observações (o conjunto de teste). Como é que vamos fazer o mesmo para uma única observação?

Você primeiro precisa criar uma nova variável 'single_observation' com os valores de entrada (por exemplo Idade = 30 e estimado Salário = 50000), e definir esta variável como uma trama de dados:

```
single_observation = data.frame (Idade = 30, EstimatedSalary = 50000)
```

Então você pode simplesmente prever o resultado desta única observação exatamente como fizemos com o conjunto de teste, por apenas substituindo 'test_set' por 'single_observation':

```
prob_pred = prever (classificador, type = 'resposta' , Newdata = single_observation) y_pred = ifelse (prob_pred> 0,5, 1, 0)
```

Como inverter a escala em R?
Você pode fazê-lo com o seguinte código:

```
training_set [-3] * attr (training_set [-3], 'Escalado: Escala' )
+    attr (training_set [-3], 'Escalado: Centro' )
test_set [-3] * attr (test_set [-3], 'Escalado: Escala' )
+    attr (training_set [-3], 'Escalado: Centro' )
```

Como visualizar a mesma classi fi cação quando as variáveis dependentes tem mais de duas classes?

Vamos supor que temos por exemplo três categorias (ou classes). Em seguida, o código para a visualização seria o seguinte:

```
#   Visualizar os resultados conjunto de treinamento
biblioteca (ElemStatLearn) definir =
training_set
X1 = seq (min (conjunto [, 1]) - 1, max (set [, 1]) + 1, por = 0,01)
X2 = seq (min (conjunto [, 2]) - 1, max (set [, 2]) + 1, por = 0,01)
grid_set = expand.grid (x1, x2) COLNAMES (grid_set) = C ( 'PC1' , 'PC2' ) Y_grid = prever
(classificador, newdata = grid_set) trama (set [, -3],


     principais = 'SVM (conjunto de treinamento)' , XLAB = 'PC1' , Ylab = 'PC2' , Xlim = intervalo (X1), ylim = intervalo (X2)) de contorno (X1, X2,
   matriz (as.numeric (y_grid), comprimento ( X1), comprimento ( X2)), adicione = TRUE)
```

```
pontos (grid_set,
        pch = '' ,
        col = ifelse (== y_grid 2, 'Deepskyblue' , Ifelse (y_grid == 1, 'Springgreen3' , 'tomate' )))
pontos (set,
        PCH = 21,
        BG = ifelse (conjunto [, 3] == 2,
        'Blue3' , Ifelse (set [, 3] == 1,
        'Green4' ,
        'red3' )))
```

3.2 K vizinhos mais próximos (K-NN)

3.2.1 K-NN intuição é K-NN um modelo

linear?
Não, K-NN é um modelo não linear, como você verá nas seções práticas deste curso.

E se encontramos igual número de vizinhos? Por exemplo, se escolhermos K = 6 e se achar 3 pontos de dados em cada categoria, onde devemos encaixa o novo ponto de dados?
O ponto é então atribuído aleatoriamente a um dos dois.

Que tipo de problemas de negócios requer K-NN?
K-NN pode ser uma boa solução para processamento de linguagem natural. Você vai ver que na Parte 7. Além disso, K-NN funciona bem para problemas não lineares (quando os dados são não linearmente separáveis), como você verá nesta seção (Python ou R).

3.2.2 K-NN em Python

Qual é o significado de n = 5, quando a construção do modelo K-NN?
n = 5 significa que você vai treinar o modelo K-NN com 5 vizinhos. Quando você assistir o tutorial intuição para K-NN, n = 5 é o número de vizinhos Kirill menciona nos vídeos.

O número de vizinhos devemos escolher?
Quanto mais você tem vizinhos, mais essa equipe de vizinhos tem chance de fi previsões nd corretos, e, portanto, quanto mais a sua precisão modelo tem chance de aumentar. No entanto tenha cuidado, se você tem muitos vizinhos, que fará com que mais de fi tting no conjunto de treinamento e as previsões será pobre em novas observações no conjunto de teste.

Como posso inverter a escala para obter a escala original dos recursos?
Você pode re-transformar idade e salário usando o método inverse_transform (). Você levá-lo scaler objeto "sc" e você aplicar o método inverse_transform () sobre ele desta maneira:

```
X_train = sc.inverse_transform (X_train) X_test = sc.inverse_transform
(X_test)
```

3.2.3 K-NN em R

Como visualizar a saída na forma de gráfico, se o número de preditores é mais do que 2?
Você não podia porque um preditor corresponde a uma dimensão. Então, se temos mais de 2 preditores, seria difícil para representar a saída visual. Se temos três preditores, temos ainda pode fazê-lo usando algum

pacotes avançados e, nesse caso, as regiões de previsão se tornaria alguns volumes de previsão (você sabe espaços região em 3 dimensões), eo limite previsão não seria mais uma curva fi tting, mas se tornaria algum plano curvo separa os volumes de previsão. Seria um pouco mais difícil de ver que visualmente, mas ainda é possível. Mas, em seguida, em 4+ dimensões, que seria muito difícil.

O número de vizinhos devemos escolher?
Quanto mais você tem vizinhos, mais essa equipe de vizinhos tem chance de fi previsões nd corretos, e, portanto, quanto mais a sua precisão modelo tem chance de aumentar. No entanto tenha cuidado, se você tem muitos vizinhos, que fará com que mais de fi tting no conjunto de treinamento e as previsões será pobre em novas observações no conjunto de teste.

3.3 Support Vector Machine (SVM)

3.3.1 SVM Intuição SVM é um

modelo linear?
Sim, SVM é um modelo linear. Você verá que facilmente nas seções práticas deste curso, ao visualizar os resultados no gráfico (você vai notar que o limite previsão é uma linha reta). No entanto, podemos fazer o SVM um modelo não linear, através da adição de um kernel, o que você verá na próxima seção.

Por que vemos os vetores de suporte como vetores não como pontos?
Os vetores são pontos no espaço 2-D (como neste exemplo), mas em problemas do mundo real, temos conjuntos de dados de dimensões superiores. Em um espaço n-dimensional, vetores fazer mais sentido e é mais fácil de fazer aritmética vetorial e manipulações de matriz, em vez de considerá-los como pontos. É por isso que generalizar os pontos de dados para vetores. Isso também nos permite pensar deles em um espaço dimensional-N.

3.3.2 SVM em Python

O que o método fi t fazer aqui?
Ele simplesmente irá treinar o modelo SVM em X_train e y_train. Mais precisamente, o método fi t irá recolher os dados em X_train e y_train, e desde que ele irá calcular os vetores de suporte. Uma vez que os vetores de suporte são computados, o seu modelo de fi classificador está totalmente pronto para fazer novas previsões com o método de prever (porque ele requer apenas os vetores de suporte para classificar novos dados).

3.3.3 SVM em R

Se o meu conjunto de dados tem mais de 2 variáveis independentes, como faço para traçar a visualização?
Nesse caso, você não poderia traçar a visualização, porque uma variável independente corresponde a uma dimensão. Então se você tivesse, por exemplo, 8 variáveis independentes, você precisa criar uma trama em 8 dimensões. Isso seria impossível. No entanto, você pode usar técnicas de redução de dimensionalidade para reduzir o número de variáveis independentes, extraindo os cativas mais estatisticamente signi fi. Então, se você conseguir reduzir seus 8 variáveis originais para baixo para dois, então você pode traçar a visualização em 2D.

Por que recebo di ff erent resultado em Python e R?
O erência di ff podem vir de uma variedade de factores:
- os fatores aleatórios no modelo e o fato de que usamos di sementes ff erent,
- os valores erent padrão di ff dos hiperparâmetros (os parâmetros que não são aprendidas), mas você dizer que eles são os mesmos,

- ligeira di ff rências nos algoritmos através Python e R.

3,4 Kernel SVM

3.4.1 Kernel SVM Intuition

Por que exatamente estamos convertendo o espaço dimensional elevada em 3D de volta para 2D?
Isso é porque nós precisamos voltar ao nosso espaço original que contém nossas variáveis independentes. Se você ficou no espaço 3D, você perderia a informação de suas variáveis independentes porque neste espaço 3D não são suas variáveis independentes originais, mas as projeções deles. Então você quer voltar para o espaço original 2D, projetando para trás os pontos.

Quando aplicamos a transformação $f = x - 5$ os pontos apenas mover para a esquerda no mesmo eixo 1D. Mas quando aplicamos a transformação $f = (x - 5)^2$, por que os pontos de mover-se para uma curva U em 2D? Se os pontos não ficar no mesmo eixo em 1D?

Dois tipos di ff erent de transformação estão envolvidos aqui, o primeiro deles é uma transformação em uma dimensão para a coordenada x, e deve ser visto desta maneira:

$$X \leftarrow X - 5$$

Com esta transformação (chamado de tradução), os pontos são movidos 5 unidades para a esquerda. Em seguida, a transformação ocorre de mapeamento, que é a transformação de mapeamento envolvido em Kernel SVM. Mas este deve ser visto desta maneira:

$$Y = (x' - 5)^2$$

onde y é a nova coordenada que você criar com esse mapeamento em dimensão superior.

Que Kernel para escolher?
Uma boa maneira de decidir qual kernel é o mais adequado é fazer vários modelos com kernels erent di ff, em seguida, avaliar cada um de seu desempenho, e finalmente comparar os resultados. Então você escolhe o kernel com os melhores resultados. Tenha o cuidado de avaliar o desempenho do modelo em novas observações (de preferência com K-Fold Cruz Validação que veremos na Parte 10) e considerar métricas di ff erent (precisão, a pontuação F1, etc.).

3.4.2 Kernel SVM em Python que faz o método fi t fazer

aqui?
Ele simplesmente irá treinar o modelo SVM em X_train e y_train com um não linear Kernel. Mais precisamente, o método fi t irá recolher os dados em X_train e y_train, e desde que ele vai fazer todas as operações sucessivas que você viu no Palestra Intuição: primeiro, um mapeamento em um espaço dimensional mais elevada, onde os dados são linearmente separáveis, então computar os vectores de apoio neste espaço dimensional mais elevada, e, eventualmente, uma projecção para trás em 2D.

Como prever o resultado de uma única nova observação?
Você precisa inserir os dados da sua única observação em uma matriz, escalá-lo com o nosso objeto scaler sc (precisamos dimensionar aqui porque o nosso modelo foi treinado nos dados em escala), e use o método de prever como esta:

```
single_prediction = classifier.predict (sc.transform (np.array ([[30, 80000]])))
```

3.4.3 Kernel SVM em R

Como saber facilmente com R se o conjunto de dados não é linearmente separáveis?
Se é um problema de regressão, fazer um modelo de regressão linear múltipla, com todas as suas variáveis independentes incluídos, e se é um problema fi cação classificadas, fazer um modelo de regressão logística com todo o teu independente

variáveis incluídas. Em seguida, use o resumo estatísticas () função em R para ver as estatísticas níveis de significância de suas variáveis independentes. Se alguns deles têm estatística signi fi cado (p-valores pequenos), em seguida, o conjunto de dados é provável que seja linearmente separáveis. Se não, então o seu conjunto de dados pode não ser linearmente separáveis.

Previmos os resultados de um conjunto de observações (o conjunto de teste). Como é que vamos fazer o mesmo para uma única observação?

Você primeiro precisa criar uma nova variável 'single_observation' com os valores de entrada (por exemplo Idade = 30 e estimado Salário = 50000), e definir esta variável como uma trama de dados:

```
single_observation = data.frame (Idade = 30, EstimatedSalary = 50000)
```

Então você pode simplesmente prever o resultado desta única observação exatamente como fizemos com o conjunto de teste, por apenas substituindo 'test_set' por 'single_observation':

```
y_pred = prever (classificador, newdata = single_observation)
```

Como inverter a escala em R?
Você pode fazê-lo com o seguinte código:

```
training_set [-3] * attr (training_set [-3], 'Escalado: Escala' )
+    attr (training_set [-3], 'Escalado: Centro' )
test_set [-3] * attr (test_set [-3], 'Escalado: Escala' )
+    attr (training_set [-3], 'Escalado: Centro' )
```

3,5 Naive Bayes

3.5.1 Naive Bayes Intuition

Naive Bayes é um modelo linear ou um modelo não linear?
Naive Bayes é um modelo não linear. Irá ver muito claramente que em Python ou R quando traçando o limite de previsão que será uma curva muito agradável bem separando as observações não linearmente distribuídos.

lata *P (x)* ser zero?
sim *P (X)* pode ser zero. Isso pode acontecer quando o novo ponto de dados é um outlier, e, portanto, não há outros pontos de dados dentro do raio do círculo. No entanto, a fórmula do teorema de Bayes Naive só é verdade quando *P (X)* é erent di ff do que zero. Quando *P (X)* é igual a zero, em seguida, *P (anda | X)* é apenas definido para zero.

Como o algoritmo de decidir o círculo?
Na palestra intuição, vemos que um círculo é desenhado para criar uma coleção de pontos de dados semelhantes para o novo datapoint. O novo datapoint era mais ou menos no centro do círculo e, portanto, nós vimos esse número de pontos verdes foram menor do que o número de pontos vermelhos e, portanto, o novo ponto foi para a categoria vermelha. Mas se tinha desenhado o círculo um pouco di ff erently em torno do novo datapoint, em seguida, o número de pontos verdes poderia ter sido mais do que vermelho. Então, como é que o círculo escolhido? Existe um parâmetro no modelo que decide o raio do círculo, assim como não há um parâmetro que escolhe o número de vizinhos em K-NN.

Naive Bayes vs K-NN
A melhor maneira de responder a essa pergunta é tentar ambos. Graças aos modelos levará apenas alguns minutos para comparar os seus resultados. Basta compreender que o principal rência di ff entre os dois é que Naive Bayes é baseado em uma abordagem probabilística, ao contrário do clássico K-NN.

3.5.2 Naive Bayes em Python que faz o método fi t

fazer aqui?
Ele simplesmente irá treinar o modelo Naive Bayes na X_train e y_train. Mais precisamente, o método fi t irá recolher os dados em X_train e y_train, e desde que ele vai fazer todas as operações sucessivas que você viu no Palestra Intuição: primeiro ele vai ficar a probabilidade prévia, a probabilidade marginal, a probabilidade ea probabilidade posterior e, em seguida, usando os círculos ao redor das observações ele vai aprender a classificar essas observações e prever resultados futuros.

Como obter as probabilidades como saída?
Aqui é a maneira de fazer isso:

```
y_proba = classifier.predict_proba (X_test)
```

Isto lhe dará uma matriz com 2 colunas, uma para 0 e um para um.

3.5.3 Naive Bayes em R

Por que precisamos para converter a variável dependente como um fator?
Fazemos isso para especificar que a variável dependente é uma variável categórica e que 0 e 1 devem ser tratados como categorias (fatores).

Nos naiveBayes () função, está a utilizar a fórmula argumento'= Comprada ~. 'Semelhante ao especificar x e y como fizemos no vídeo?

Sim, isso seria o mesmo, mas você precisa especificar o argumento de dados. Basicamente, você deve especificar o que é a variável dependente, a variável independente e o conjunto de treinamento. Isso é su informações ffi ciente para a função de saber como treinar o modelo.

3.6 Árvore de Decisão Classi fi cação

3.6.1 Árvore de Decisão Classi fi cação Intuição Como o algoritmo de

dividir os pontos de dados?
Ele usa diminuição do desvio padrão das previsões. Em outras palavras, o desvio padrão é reduzido direita depois de uma separação. Assim, a construção de uma árvore de decisão é toda sobre encontrando o atributo que retorna a maior redução desvio padrão (ou seja, os ramos mais homogéneos).

Qual é o ganho de informação e como ele funciona em árvores de decisão?
O ganho de informação na árvore de decisão de regressão é exatamente a Redução Desvio Padrão estamos à procura de alcançar. Calculamos por quanto o desvio padrão diminui após cada divisão. Porque quanto mais o desvio padrão é diminuído depois de uma separação, as mais homogêneas os nós filho será.

Qual é a Entropia e como ele funciona em árvores de decisão?
A entropia mede a desordem num conjunto, aqui em parte resultante a partir de uma divisão. Assim, quanto mais homogênea é seus dados em uma parte, menor será a entropia. Quanto mais você tem splits, quanto mais você tem chance de fi partes nd em que os seus dados são homogêneos, e, portanto, menor será a entropia (próximo a 0) por estas bandas. No entanto, você ainda pode achar alguns nós onde os dados não é homogênea, e, portanto, a entropia não seria tão pequena.

3.6.2 Árvore de Decisão Classi fi cação em Python que faz o método fi

t fazer aqui?

Ele simplesmente irá treinar o modelo de árvore de decisão relativa X_train e y_train. Mais precisamente, o método fi t irá recolher os dados em X_train e y_train, e desde que ele vai fazer todas as operações sucessivas que você viu no Palestra Intuição: primeiro ele irá selecionar o critério (que nós escolhemos para ser a entropia em nosso código) , e, em seguida, ele vai dividir os dados em di ff erent nódulos sucessivos na direcção de redução de entropia, no sentido de nodos mais homogéneos. Eventualmente, quando todas as divisões são feitas, será totalmente treinado e pronto para prever os resultados de novas observações graças ao método de prever ().

Como traçar o gráfico de uma árvore em Python?

Aqui está como você pode traçar uma árvore de decisão em Python (basta adicionar o seguinte código no final do seu código):

```
# Conspirando a árvore
# No terminal digite: pip instalar pydot2
de sklearn importar árvore de sklearn.externals.six importar StringIO

de IPython.display importar Imagem
importar pydot
dot_data = StringIO ()
tree.export_graphviz (classificador,
                                    out_file = dot_data,
                                    feature_names = [ 'Era' , 'Salário estimado' ], class_names = [ 'Sim' , 'Não' ], cheios =
                                    TRUE, arredondado = TRUE, special_characters = TRUE)



gráfico = pydot.graph_from_dot_data (dot_data.getvalue ())
Imagem (graph.create_png ())
```

3.6.3 Árvore de Decisão Classi fi cação em R Como foram as

divisões feitas no R

Ele fez essas divisões com salários agrupados em três níveis, com base na entropia e ganho de informação. As separações foram efectuadas de modo a que as partes resultantes das separações são tão homogéneo quanto possível.

Por que é y_pred uma matriz neste modelo, enquanto em todo o outro modelo é um vetor?

É uma cidade fi específica do modelo de árvore de decisão: as previsões são retornados em uma matriz de duas colunas com as probabilidades prov (y = 0) e prov (y = 1).

3,7 aleatória Floresta Classi fi cação

3.7.1 aleatória Floresta Classi fi cação Intuition

Qual é a vantagem e desvantagem de aleatórios Florestas comparação com árvores de decisão? Vantagem: Random florestas podem dar-lhe um melhor poder preditivo de Árvores de Decisão.

Drawback: Árvore de Decisão vai lhe dar mais do que interpretability aleatórios Florestas, porque você pode traçar o gráfico de uma árvore de decisão para ver as separações di ff erent que levam à previsão, como visto no Palestra intuição. Isso é algo que você não pode fazer com aleatórios Florestas.

Quando usar aleatória Floresta e quando usar os outros modelos?
A melhor resposta para essa pergunta é: julgá-los todos!
Na verdade, graças aos modelos que só levará 10 minutos para tentar todos os modelos, o que é muito pouco em comparação com o tempo dedicado às outras partes de um projeto de ciência de dados (como dados de pré-processamento, por exemplo). Então, basta não ter medo de experimentar todos os modelos fi cação classi e comparar os resultados (por meio de validação cruzada que veremos na Parte 10). Isso é que lhe deu os modelos máximas neste curso para você ter no seu kit de ferramentas e aumentar a sua chance de conseguir melhores resultados.

No entanto, em seguida, se você quiser alguns atalhos, aqui estão algumas regras de polegares para ajudá-lo a decidir qual modelo usar:

Primeiro, você precisa figura se o seu problema é linear linear ou não. Você vai aprender como fazer isso na Parte 10 - Selecção do modelo.
Então: Se o seu problema é linear, você deve ir para um modelo de regressão logística ou um modelo SVM, porque estes são os dois modelos lineares.
Se o seu problema é não linear, você deve ir para qualquer K-NN, Kernel SVM, Naive Bayes, árvore de decisão ou aleatória Forest. Então qual delas você deve escolher entre estes cinco? Que você vai aprender na Parte 10 - Selecção do modelo. O método consiste na utilização de uma técnica muito relevante que avalia o seu desempenho modelos, chamado k-Fold Cruz Validação e, em seguida, escolher o modelo que apresenta os melhores resultados. Sinta-se livre para pular diretamente para a Parte 10 se você já quer aprender como fazer isso.

3.7.2 aleatória Floresta Regressão em Python Como posso saber

quantas árvores devo usar?
Em primeiro lugar, eu recomendaria para escolher o número de árvores através da experimentação. Geralmente, leva menos tempo do que nós pensamos que a fi gura um melhor valor por ajustes e ajustar seu modelo manualmente. Isso é realmente o que fazemos em geral, quando nós construímos um modelo de Aprendizado de Máquina: nós fazê-lo em vários tiros, experimentando vários valores de hiperparâmetros como o número de árvores. No entanto, também sabemos que na Parte 10 vamos cobrir k-Fold Cruz Validação e grade Search, que são técnicas poderosas que você pode usar para fi nd o valor ideal de um hiperparâmetro, como aqui o número de árvores.

3.7.3 aleatória Floresta de regressão em R

Como é que vamos decidir quantas árvores seria suficiente para obter um resultado relativamente precisas?
Mesmo que em Python, você pode encontrar um número relevante de árvore através da experimentação e ajustes manual. Você deve usar árvores suficientes para obter uma boa precisão, mas você não deve usar muitas árvores porque isso poderia causar mais fi tting. Você vai aprender a achar o número ideal de árvores na seção de primeiros da Parte 10 - Selecção do modelo. É feito com uma técnica chamada de ajuste dos parâmetros (Grid Pesquisa com k-Fold Validação Cruzada).

podemos selecionar os mais signi fi cativos variáveis graças ao valor-p, como fizemos em R antes?
Você não pode usar o valor-p devido aleatórios Florestas não são modelos lineares, e p-valores aplicam-se apenas aos modelos lineares. Portanto apresentam selecção está fora de questão. Mas você poderia fazer extração de características, que você vai ver na Parte 9 - dimensionalidade Redução. Que você pode aplicar a aleatórios Florestas, e vai reduzir o número de suas características.

Como pode ser reduzido ao longo fi tting da Random Florestas?
Você pode fazer isso por jogar com os parâmetros de penalização e de regularização, se houver algum na classe utilizada para construir o modelo. No entanto, o melhor e mais maneira e ffi ciente para reduzir o excesso fi tting, especialmente para Random Forests, é a aplicação de validação cruzada k vezes e optimizar um parâmetro de ajuste de dados bootstrap. Não se preocupe, vamos ver tudo isso em profundidade na Parte 10 - Selecção do modelo & Ensemble Learning.

3.8 Avaliando Classi fi cação modelos de desempenho

Quanto devemos contar com a matriz de confusão?
Podemos usar a matriz de confusão para ter uma noção de como potencialmente bem o nosso modelo pode executar. No entanto, não devemos limitar a nossa avaliação do desempenho para a precisão na divisão teste de um trem, como é o caso da matriz de confusão. Em primeiro lugar, há o problema variância para que você deve sim aplicar k-Fold Cruz Validação para obter uma medida mais relevante da precisão. Você vai aprender como fazer isso na primeira seção da Parte 10 - Selecção do modelo. Então, não há apenas a precisão que você deve considerar. Você também deve considerar outras métricas, como precisão, Recall, ea pontuação F1. Você vai ver que na Parte 7.

É curva de ganho acumulado a mesma coisa que curva CAP?
Sim, isso é o mesmo, Gain Curve é o nome para CAP Curve em um contexto de marketing.

Se você segmentar 20000 clientes e se 4000 respondem, então por que não ter uma multiplicação de um ff ect? Então, se eu direcionar 100000 clientes, então eu deveria obter uma resposta de 50000?
A idéia aqui é que, podemos maximizar os retornos, escolhendo os clientes que podem comprar mais, de tal forma que, se p (customer_will_purchase) é maior do que um limite, então devemos enviar uma mensagem de e-mail.

A idéia é que você nunca vai obter uma resposta de 50.000 uma vez que, aconteça o que acontecer, apenas 10.000 clientes vão comprar em cada er o ff.

Então, o que nós gostaríamos de fazer é que a partir de nossa base de clientes de 1,00000, escolher esses 10.000 clientes exatamente o que vai comprar para a corrente er o ff. Se temos um modelo tão bom, que pode com precisão pin-point todos que vai comprar, então podemos apenas enviar um e-mail que muitas pessoas.
Não há e multiplicativo ff ect aqui. Deixe-me explicar. Mesmo que enviar e-mails para todos os clientes (100k), então só 10k vai comprar. (90% dos clientes não é normalmente interessados em cada ff er o). Nosso objetivo aqui é minimizar os e-mails que enviamos através do envio de e-mails para a 10k que vai comprar nosso er o ff. Assim, podemos usar modelos probabilísticos para prever tempo ou não um cliente vai comprar. Se a probabilidade de que o cliente vai comprar é maior do que um limite, podemos enviar-lhes um e-mail. Isto irá aumentar a chance para cada cliente a comprar o ff er o, mantendo o número de e-mails a um mínimo.

Devemos traçar a curva CAP em Python ou R?
O modelo do Excel fornecido no curso é mais recomendado. Em primeiro lugar, exportar os seus valores em Python:

```
classificador = SVC (kernel = 'RBF' , Probabilidade = 1, random_state = 0) Y_pred_proba = classifier.predict_proba (X_test)
Y_cap = np.c_ [Y_test, Y_pred_proba [:, 1:]] trama de dados = pd.DataFrame (dados = Y_cap)


dataframe.to_csv ( 'CAP.csv' , Setembro = ';' )
```

Em seguida, usar este modelo Curve PAC que lhe dará diretamente a curva CAP uma vez que você colar seus valores exportados dentro deste modelo.

Quanto a R, há um par de R pacotes você pode usar para calcular uma curva CAP: "ROCR" e "em massa".

Quais são Baixo / Viés / Desvio alta em Machine Learning?
Low Viés é quando suas previsões do modelo são muito próximos dos valores reais. Alta Viés é quando
suas previsões do modelo estão longe dos valores reais.
Baixa Variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação não irá variar muito.

Alta variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação irá variar muito.

O que você deseja obter quando você constrói um modelo é: Low preconceito e baixa variância.

4 Parte 4 - Clustering

4.1 K-Means Clustering

4.1.1 K-means intuição

Onde podemos aplicar algoritmo de agrupamento na vida real?
Você pode aplicá-los para fins erent di ff:

- Segmentação de mercado,
- Medicamento com para a detecção do tumor exemplo,
- Detecção de fraude,
- para identificar simplesmente alguns clusters de seus clientes em sua empresa ou negócio.

Como é que a linha perpendicular truque trabalho quando $k \geq 3$?
Este truque é usado principalmente apenas em espaços 2D ou 3D. Geralmente, a distância Euclidiana é usado em todos os pontos altos para dimensões para realizar a aglomeração. Em dados dimensionais muito elevadas, que poderia fazer um truque, que é uma "esfera ng engul fi", ou seja a partir de cada centróide (esferas k no total), começar a crescer de um espaço esférico (círculo em 2D, esfera em 3D, etc.), para fora radialmente . Até as esferas se cruzam, tudo o que a esfera engolfa pertence ao mesmo cluster. Todos os outros pontos podem ser atribuídos a grupos calculando a distância perpendicular entre os centróides.

Como o método de cotovelo trabalho quando mais de 2 características estão envolvidos?
É o mesmo, mas que utilize a distância Euclidiana em n dimensões para calcular o WCSS (p e q são dois pontos de observação, e as suas **coordenadas $p_1, ..., p_N, q_1, ..., q_n$ são as características desses pontos de observação):**

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

Existe uma maneira alternativa ao método Elbow de encontrar o número máximo de clusters?
Claro, uma forma alternativa seria através de ajuste dos parâmetros com Grid Search, que você vai ver na Parte 10 - Selecção do modelo.

4.1.2 K-means em Python

O que é exatamente o parâmetro 'max_iter' na classe kmeans ()?
Quando aplicamos k-Means, que saltar de volta para a etapa anterior muitas vezes (ver Lectures Intuição). Cada vez que fazemos isso, que é uma iteração. E max_iter é o número máximo dessas iterações.

Por que não considerar a idade como um parâmetro?
Nós não consideramos a idade por duas razões:
1. Eu tinha pré-marcada não tivesse impacto sobre a variável dependente.
2. queria ter duas variáveis independentes na extremidade, de modo que podemos visualizar os resultados em duas dimensões (desde uma variável independente corresponde a uma dimensão).

Como obter estatísticas para cada um dos grupos?
A maneira mais simples de obter alguma informação estatística dos clusters é usar o atributo cluster_centers_ da classe kmeans. Ele irá dar uma matriz com as coordenadas de cada aglomerado, que é a média de cada coluna característica de cada aglomerado. Em seguida, você também pode obter algumas informações sobre os pontos de observação usando o atributo labels_. Você pode ligar para esses atributos desta forma:

```
kmeans.cluster_centers_ kmeans.labels_
```

Então, você poderia obter alguns valores úteis como o número de elementos em cada cluster, a média dos salários ou pontuações de gastos em cada cluster, etc. A melhor maneira de fazer isso é colocar os cinco conjuntos em 5 variáveis da seguinte maneira:

```
Cluster_0 = X [y_kmeans == 0] Cluster_1 = X
[y_kmeans == 1] Cluster_2 = X [y_kmeans == 2]
Cluster_3 = X [y_kmeans == 3]

Cluster_4 = X [y_kmeans == 4]
```

Então você começa o que quiser com estas variáveis que são exatamente os clusters. Por exemplo, se você estiver interessado em ter a contagem de observações para cada cluster, bem que agora você pode vê-lo diretamente no Explorer variável, olhando para o número de linhas para cada cluster. Ou você pode fazer isso:

```
len (Cluster_0) len
(Cluster_1) len (Cluster_2)
len (Cluster_3)

len (Cluster_4)
```

Então você pode começar a média eo desvio padrão de cada recurso do cluster desta maneira:

```
Cluster_0 [0] .mean () Cluster_1 [0]
.mean () Cluster_2 [0] .mean ()
Cluster_3 [0] .mean ()

Cluster_4 [0] .mean ()
Cluster_0 [0] .std () Cluster_1 [0]
.std () Cluster_2 [0] .std ()
Cluster_3 [0] .std ()

Cluster_4 [0] .std ()
```

Você também pode obter a percentagem de observações pelos seguintes cálculos simples:

```
len (Cluster_0) / 200 len (Cluster_1) /
200 len (Cluster_2) / 200 len
(Cluster_3) / 200

len (Cluster_4) / 200
```

Por favor, explique a seguinte linha de código em mais detalhes:

```
plt.scatter (kmeans.cluster_centers _ [:, 0],
                        kmeans.cluster_centers_ [:, 1],
                        s = 300, c = 'amarelo' ,

                        label = 'Centróides' )
```

Na função de dispersão () do módulo de PLT:

- o argumento primeira é a coordenada x dos centros dos grupos (o centróide),
- o segundo argumento é a coordenada y dos centros dos grupos (o centróide),
- o terceiro argumento s é o tamanho dos pontos centróide na trama,
- o quarto argumento é a cor dos pontos centróide na trama,
- o argumento quinto é o rótulo que aparece na legenda correspondente aos centróides.

4.1.3 K-means em R

Como não cair na armadilha de inicialização aleatória na R?
R cuida de não cair na armadilha de inicialização aleatória nos bastidores. No entanto você tem algumas outras grandes pacotes em R onde você pode usar explicitamente ++-means k, é o pacote KMeans_rcpp.

Por que não usou todas as características como a idade?
Eu não usá-los porque eu pré-marcada que não teve impacto sobre a variável dependente. Além disso, eu queria ter duas variáveis independentes, para que pudéssemos visualizar os resultados em duas dimensões (porque uma variável independente corresponde a uma dimensão).

Qual é o papel de niniciar aqui?
Porque K-means começa com k centroides escolhidos aleatoriamente, uma solução erent di ff pode ser obtido sempre que a função é invocada. Usando a função set.seed () garante que os resultados são reprodutíveis. Além disso, esta abordagem de agrupamento pode ser sensível à selecção inicial do centróide. A função kmeans () tem uma opção niniciar que tenta vários gurações e relatórios sobre a melhor iniciais con fi. Por exemplo, a adição de niniciar = 25 gera 25 iniciais gurações con fi. Esta abordagem é frequentemente recomendada.

Como posso traçar os resultados sem a escala e com os nomes dos clusters?
Aqui é uma implementação de um plano tal:

```
trama (x = conjunto de dados [, 1],
        y = conjunto de dados [, 2], col
        = y_kmeans, PCH = 19,

5       xlim = C (a partir de = min (conjunto de dados [, 1]), a = max (conjunto de dados [, 1] 30)), XLAB = "Rendimento
        anual" , Ylab = "Passar Score" )
= c (aglomerados "Descuidado" , "Padrão" , "Sensible" , "Alvo" , "Cuidado" ) lenda( 'canto inferior direito' , Legenda = aglomerados, col =
1: 5, PCH = 19, horiz = F)
```

4,2 agrupamento hierárquico

4.2.1 hierárquica Clustering Intuition

Qual é o ponto de Hierarchical Clustering se ele sempre leva a um cluster por ponto de observação?

O principal ponto de Hierarchical Clustering é fazer o dendrograma, porque você precisa para começar com um único cluster, em seguida, trabalhar o seu caminho para baixo para ver as combinações di ff erent de aglomerados até ter um número de clusters igual ao número de observações. E é o dendrograma própria que permite achar o melhor agrupamento con fi guração.

Quando você está comparando a distância entre dois clusters ou um cluster e um ponto, exatamente como é medido? Você está tomando o centróide do cluster e medir a distância?
Precisamente, a métrica é a distância euclidiana entre o centróide do agrupamento primeiro e o ponto, (ou o centróide do outro conjunto para a distância entre dois agregados).

Você também precisa executar recurso de escala para Hierarchical Clustering?
Sim, porque as equações dos problemas de clustering envolvem a Distância Euclidiana. A qualquer momento as equações do modelo envolve a Distância Euclidiana, você deve aplicar função Dimensionamento.

4.2.2 agrupamento hierárquico em Python

Devemos usar o dendrograma ou o método cotovelo para descobrir que número ideal de clusters?
Você deve usar ambos (é mais rápido para tentar os dois do que você pensa, graças aos modelos), apenas para confirmar que o número ideal. No entanto, se você realmente só tem tempo para um, eu recomendaria o método cotovelo. O dendrograma nem sempre é a maneira mais fácil de encontrar o número ideal de clusters. Mas com o método de cotovelo é muito fácil, uma vez que o cotovelo é na maioria das vezes muito óbvia de detectar.

você poderia expandir sobre como os clusters são formados através do método ala da classe AgglomerativeClustering () Python?

agrupamento hierárquico é uma família geral de algoritmos de agrupamento que construir conjuntos aninhados por fusão ou dividindo-os sucessivamente. Esta hierarquia de clusters é representado como uma árvore (ou dendrograma). A raiz da árvore é o aglomerado único que reúne todas as amostras, as folhas sendo os clusters com apenas uma amostra. A classe AgglomerativeClustering () executa um agrupamento hierárquico utilizando uma abordagem de baixo para cima: cada observação começa no seu próprio conjunto, e os agrupamentos são sucessivamente fundidos juntos. O método de Ward minimiza a soma de rências di ff quadrado dentro de todos os grupos. É uma abordagem de minimização de variância e, nesse sentido, é semelhante à função objetiva-k significa, mas combatida com uma abordagem hierárquico aglomerativo.

Pensei que dendrogramas eram como a memória do algoritmo HC. Se assim for, por que primeiro fazer o dendrograma e depois aplicar o agrupamento aglomerativo? Isso significa que nós executamos o dobro do algoritmo de agrupamento?

A única finalidade de fazer primeiro o dendrograma é ter uma idéia do número ideal de clusters. É como a memória ou a história do algoritmo HC. Mas através dessa história / memória podemos ver o número ideal de clusters. E isso é o que fazemos nesta etapa. Então, quando nós aplicamos agrupamento aglomerativo, podemos introduzir o número ideal de agrupamentos que encontramos graças ao dendrograma.

O que é 'uma nidade ffi' na classe AgglomerativeClustering ()?
'Nidade uma ffi' refere-se à distância usada no algoritmo K-Meios, que é a distância euclidiana (o mais clássico, a uma geométrica). Refere-se à forma como os HC algoritmo de fi ne (fi NDS) as centroids mais próximos de cada ponto. E então o que vai junto com ele é o método ala usada para construir os clusters, o que significa que nós usamos o Dentro Cluster Soma dos Quadrados como um critério para se reunir e formar nossos clusters.

Por que não vamos simplesmente implementar um código que fi automaticamente NDS o número ideal de agrupamentos, em vez de selecionar manualmente ou visualmente?
Porque a escolha do número ótimo de clusters também é influenciada pelo problema de negócio (metas e restrições). Por exemplo, alguns problemas de negócios tem um número mínimo ou máximo de aglomerados, ou um número min / max de elementos por aglomerado. Tais restrições nos obrigam a ter várias opções, e a melhor maneira de figura as melhores opções estão olhando os gráficos (dendrograma ou cotovelo).

4.2.3 agrupamento hierárquico em R Por que não aplicar recurso a

escala em R?

Porque a função que usamos em R automaticamente se encarrega da função Dimensionamento em si.

Como posso visualizar os clusters com a escala original?
Se você quiser olhar para os clusters com escalas originais que você pode usar os seguintes comandos:

```
aglomerados = 1: 5 trama (x = conjunto de dados [, 1], y = conjunto de
dados [, 2],
        col = y_hc, PCH = 19, xlim = C (a partir de = min (conjunto de dados [, 1]), a = max (conjunto de dados [, 1] 20))) legenda ( 'canto inferior
direito' , Inserir = 0,01, legenda = aglomerados, col = 1: 5, PCH = 19, horiz = F)
```

5 Parte 5 - Associação regra de aprendizagem

5.1 Apriori

5.1.1 Apriori Intuition

Quais são as três relações essenciais entre o apoio, confiança e levantar?

Dado dois filmes M_1 e M_2, Aqui estão as três relações essenciais para lembrar: Relação entre o apoio eo ança

con fi:

$$\text{con fi ança}\ (M_1 \rightarrow M_2) = \frac{\text{Apoio, suporte}(M_1, M_2)}{\text{Apoio, suporte}(M_1)}$$

Relação entre o elevador eo apoio:

$$\text{lift}(M_1 \rightarrow M_2) = \frac{\text{Apoio, suporte}(M_1, M_2)}{\text{Apoio, suporte}(M_1) \times \text{Apoio, suporte}(M_2)}$$

Relação entre o elevador e o dência con fi (consequência das duas equações anteriores):

$$\text{lift}(M_1 \rightarrow M_2) = \frac{\text{con fi ança}\ (M_1 \rightarrow M_2)}{\text{Apoio, suporte}(M_2)}$$

São a con fi ança e levante funções simétricas?

Dadas as três equações da questão anterior, podemos ver facilmente que: Con fi ança não é simétrica:

$$\text{con fi ança}\ (M_1 \rightarrow M_2)\ 6 = \text{con fi ança}\ (M_2 \rightarrow M_1)$$

Elevador é simétrico:

$$\text{lift}(M_1 \rightarrow M_2) = \text{lift}(M_2 \rightarrow M_1)$$

5.1.2 Apriori em Python

questão importante: é de R melhor do que Python para a Associação regra de aprendizagem?

Sim absolutamente. O pacote R é muito mais otimizado e fácil de usar. Mas para aborrecedores R fizemos questão de fornecer uma solução Python. No entanto, se você não se importa, é altamente recomendável para ir com R.

Tenho dificuldade encontrando as regras em Python. Como posso vê-los de forma mais explícita?

Desculpe por isso, a versão do pacote mudou depois que eu gravei a Palestra. Mas você pode vê-los apenas adicionando o seguinte ao final do seu código Python (em uma nova seção logo após o "Visualizando os resultados" seção):

```
the_rules = []
para resultar em resultados:
        the_rules.append ({ 'regra' : '' . Junte-se ( result.items),
                                    'Apoio, suporte' : Result.support,
                                    'confiança' : Result.ordered_statistics [0] .confidence,
                                    'lift' : Result.ordered_statistics [0] .lift})
df = pd.DataFrame (the_rules, colunas = [ 'regra' , 'Apoio, suporte' , 'confiança' , 'lift' ])
```

Após a execução desse código, você deve ver uma trama de dados chamado 'df' que aparece no 'Explorer Variável' com todas as informações necessárias.

5.1.3 Apriori em R

Onde é que o 3 vem no cálculo de apoio?
Peguei 3 porque eu a considerar os produtos que são comprados pelo menos 3 vezes por dia. Porque menos de 3 vezes por dia parecia não ser relevante o suficiente para incluí-los na análise.

Por que o apoio e con fi ança ser mínimo?
É porque queremos que as nossas regras para ser signi fi cativa e re fl ect tendência real, não algo realmente raro que acontece uma vez em alguns anos, mas por acidente tem em nossos registros.

Por que não é uma regra como água mineral (LHS) e macarrão de trigo integral (RHS) não aparecendo no topo da lista se as regras?

É porque a água mineral pode ir com qualquer coisa, de modo que a regra de associação entre a água mineral e macarrão de trigo integral não é o mais relevante. Pense nisso desta maneira: se você comprar água mineral, você quer comprar macarrão de trigo integral como primeira escolha?

Como você medir o quão bem as regras aqui são? Você mencionou como Amazon e Net fl ix utilizar algoritmos mais avançados, mas como eles têm chegado à conclusão de que os seus algoritmos são superiores?

Na Associação regra de aprendizagem, você simplesmente medir isso com o elevador. Quanto maior são os elevadores, o melhor suas regras. Amazon e Net fl ix trabalhar com lotes de algoritmos, incluindo regras de associação, Sistemas de Recomendação e outros modelos de aprendizagem de máquina avançadas. Eles avaliam seus modelos através de métricas de desempenho e experimentar.

No cenário em tempo real o que é o período de tempo ideal que devemos considerar para fazer um bom modelo de Análise de Market Basket? E isso deve ser feito em cada loja separadamente ou região sábio?
Um mês é um período de tempo bom. No entanto, você também pode considerar 3-6 meses para normalizar a sazonalidade Effect, ou você pode executar o mesmo modelo a cada mês, o que eu preferiria fazer para pegar as cidades fi cações de cada mês (temporada, taxa de turismo, etc.).

Então Análise Market Basket deve ser feito em cada loja separadamente, uma vez que depende dos comportamentos de clientes dentro de um bairro fi c específica. Basicamente os clientes podem se comportar di ff erently entre os bairros erent di ff.

5.2 Eclat

5.2.1 Eclat Intuition

Quando devemos usar Eclat em vez de Apriori?
A única vantagem de Eclat comparação com Apriori é que é mais simples e rápido de usar. No entanto, se você precisa executar uma análise profunda da sua cesta de mercado, então você deve de fi nitivamente ir para Apriori.

5.2.2 Eclat em Python

você poderia fornecer uma implementação Eclat em Python?
Crédito para Marcello Morchio, um aluno deste curso que gentilmente partilhou a sua implementação:

```
#  Eclat, por Marcello Morchio, 04 de dezembro de 2017

#  Importando as bibliotecas
importar numpy como np
importar matplotlib.pyplot como plt
importar pandas como Pd
```

```
# A importação do conjunto de dados
conjunto de dados = pd.read_csv ( 'Market_Basket_Optimisation.csv' , Cabeçalho = Nenhum)
transações = [[str (conjunto de dados. valores [ eu j]) para j na gama (0, dataset.shape [1])
                                    E se str (conjunto de dados. valores [ i, j])! = 'Nan' ] para i na gama (0, dataset.shape [0])]

# Gerar uma lista de itens exclusivos nas transações
items = list ()
para t em transações:
        para x em t:
                E se ( não x em itens):
                        items.append (x)

# Gerar uma lista de pares de itens com valor relevante apoio
# [[(Item_a, item_b), support_value]]
# support_value é inicializado para 0 para todos os pares
brilho = lista ()
para i no intervalo (0, len (itens)):
        para j na faixa de (i + 1, itens len ()):
                eclat.append ([(itens [i], itens [j]), 0])

# Calcular valor de suporte para cada par de olhar para as operações com os dois itens
para p em brilho:
        para t em transações:
                E se ( p [0] [0] no t) e (p [0] [1] em t):
                        P [1] + = 1
        P [1] = P [1] / len (operações)

# Converte eclat na trama de dados ordenada para ser visualizado no Explorer variável
eclatDataFrame = pd.DataFrame (eclat, colunas = [ 'regra' , 'Apoio, suporte' ]). Sort_values (por = 'Apoio, suporte' , Ascendente = falso)
```

5.2.3 Eclat em R

Parece que Água Mineral está distorcendo os resultados das associações, porque é a compra mais frequente, de modo que seria razoável simplesmente excluí-lo a partir dos dados e re-construir o modelo sem ele?

Sim, isso poderia ser relevante. No entanto seria preciso verificar a parcela da receita proveniente de água mineral. Se a proporção é alta, talvez não seja uma má idéia para colocá-lo ao lado de seu melhor produto de associação depois de tudo.

Encontrámos 5 duplicatas. não devemos removê-los?

Absolutamente não. Isso poderia significar que 5 pessoas por acidente comprou o mesmo, assim que esta registros representam di clientes ff erent e, em seguida, eles devem ser salvos.

Qual é a densidade explicou em 3:23 na palestra Eclat R?

densidade é um termo que pode ter significados di ff erent em contextos erent di ff. Neste caso particular, a densidade é a proporção de células com os itens (valores não nulos na matriz) sobre o número total de células na matriz.

6 Parte 6 - Reforço de Aprendizagem

6.1 Alta Con fi ança Bound (UCB)

6.1.1 UCB Intuition

Você poderia explicar em maiores detalhes o que uma distribuição é? Quais são no eixo x e eixo y?

Vamos supor que temos uma experiência com anúncios clicando 100 vezes, e calcular qual é a frequência de cliques do anúncio. Em seguida, repeti-lo novamente por mais de 100 vezes. E novamente por mais de 100 vezes. Daí obtemos muitas frequências. Se repetirmos esses cálculos frequência muitas vezes, por exemplo 500 vezes, podemos traçar um histograma destas frequências. Pelo Teorema do Limite Central será em forma de sino, e sua média será a média de todas as frequências que obtivemos ao longo do experimento. Portanto, em conclusão, sobre o eixo x que tem os valores possíveis di ff erent destas frequências, e no eixo dos Y que tem o número de vezes que cada frequência obtida através da experiência.

Na Intuição Palestra primeiro, você poderia explicar por que D5 (em laranja) é a melhor distribuição? Por que não é D3 (em rosa)?

Nesta situação, 0 é a perda, e 1 é o ganho, ou vitória. O D5 é o melhor porque é distorcida por isso vamos ter resultados médios próximos de 1, o que significa que não temos mais vitórias, ou ganhos. E, na verdade, todas as máquinas de casino hoje em dia são cuidadosamente programado para ter distribuição como D1 ou D3. Mas é um bom exemplo concreto.

6.1.2 UCB em Python

Por que uma única rodada pode ter vários 1s para anúncios di ff erent?
Cada rodada corresponde a um usuário se conectar a uma página da web, eo usuário só pode ver um anúncio quando ele / ela se conecta à página da web, o anúncio que está sendo mostrado para ele / ela. Se ele / ela clica no anúncio, a recompensa é
1, caso contrário é 0. E não pode haver múltiplas queridos, porque existem vários anúncios que o usuário gostariam de clicar. Mas só uma bola de cristal que nos dizem quais os anúncios que o usuário deverá clicar em. E o conjunto de dados é exatamente essa bola de cristal (estamos fazendo uma simulação aqui).

Você poderia explicar em mais detalhes o que fazemos com as recompensas nas linhas 33 e 34?
Vamos explicar claramente:
- Na linha 33 nós estamos apenas começando a recompensa no especi fi c rodada n. Então temos um 1 se o usuário clicou no anúncio nesta rodada n, e 0 se o usuário não fez.
- Na linha 34 nós estamos começando a soma de todas as recompensas até rodada n. Portanto, esta soma é incrementado por 1 se o usuário clicou no anúncio nesta rodada n, e permanece o mesmo se o usuário não fez.

A estratégia UCB realmente entrar em Effect durante as rodadas primeiros?
Não no início, primeiro temos que ter alguns insights primeiros dos usuários resposta aos anúncios. Isso é apenas o começo da estratégia. No mundo real para a primeira dez usuários se conectar à página da Web, você iria mostrar ad 1 para o usuário em primeiro lugar, ad 2 para o segundo usuário, ad 3 para a terceira usuário, ..., ad 10 a 10 de utilizador. Em seguida, o algoritmo começa.

Qual foi o objetivo deste algoritmo? Por que eu não podia simplesmente contar que tipo de anúncio convertido a mais e, em seguida, usado isso?

Porque em cada rodada há um custo (o custo de colocar um anúncio na página). Então, basicamente o propósito de UCB não é apenas para maximizar a receita, mas também para minimizar o custo. E se você apenas contar que tipo de anúncio convertido a mais, que exigiria que você experimentar um monte de rodadas e, portanto, o custo seria alto e de fi nitivamente não minimizado.

6.1.3 UCB em R

Como obter este tipo de conjunto de dados?

Na vida real, você poderia fazer isso por dados de streaming, usando faísca ou Hive, ou seja, você iria receber dados em tempo real. Caso contrário, se você quer avaliar os seus modelos de reforço de aprendizagem, você pode simular um conjunto de dados como a que usamos nos tutoriais.

O que é exatamente o "número de seleções?

'Número de seleções' é o número de vezes que o anúncio foi exibido para um usuário.

Eu não entendo por que não há seleção de anúncios pelo algoritmo na primeira 10 rodadas. O instrutor disse que não há estratégia no início, para a primeira rodada vai exibir anúncios 1, 2a rodada anúncio 2, 3 dC round 3, e assim por diante .. Então, por que não há seleção por UCB?

Porque neste momento não sabemos nada sobre os anúncios e não podemos aplicar nossa fórmula, pois envolve divisão por quantas vezes um anúncio foi mostrado. Nós não podemos dividir por 0.

Em Python o anúncio vencedor foi anúncio ref. 4 e R era anúncio ref. foi utilizado 5. No entanto, o mesmo fi Ads_CTR_Optimisation.csv le. Por que estamos recebendo resultados de dois di ff erent usando a mesma lógica?

Como os índices em Python começar a partir de 0 e índices em R começar a partir de 1. Assim Python e R tanto obter o mesmo anúncio.

Poderia, por favor fornecer uma implementação R da curva pesar UCB?

```
# A importação do conjunto de dados
= conjunto de dados ler . CSV ( 'Ads_CTR_Optimisation.csv' )

# Implementando UCB
N = 10.000
d = 10
ads_selected = número inteiro (0) numbers_of_selections =
número inteiro (d) sums_of_rewards = número inteiro (d)

total_reward = 0
rewards_at_each_step = número inteiro (0) best_selection = (rowSums (conjunto de dados) ==
0) # minha disso
best_rewards_at_each_step = número inteiro (0) # minha disso
para ( n em 1: N) {
    ad = 0
    max_upper_bound = 0
    para ( i no 1: d) {
        E se ( numbers_of_selections [i]> 0) {
            average_reward = sums_of_rewards [i] / numbers_of_selections [i]
            delta_i = sqrt ( 3/2 * registro ( n) / numbers_of_selections [i]) = upper_bound average_reward +} delta_i outro
        {

            upper_bound = 1e400}

        E se ( upper_bound> max_upper_bound) {
            max_upper_bound = ad upper_bound = i}}
```

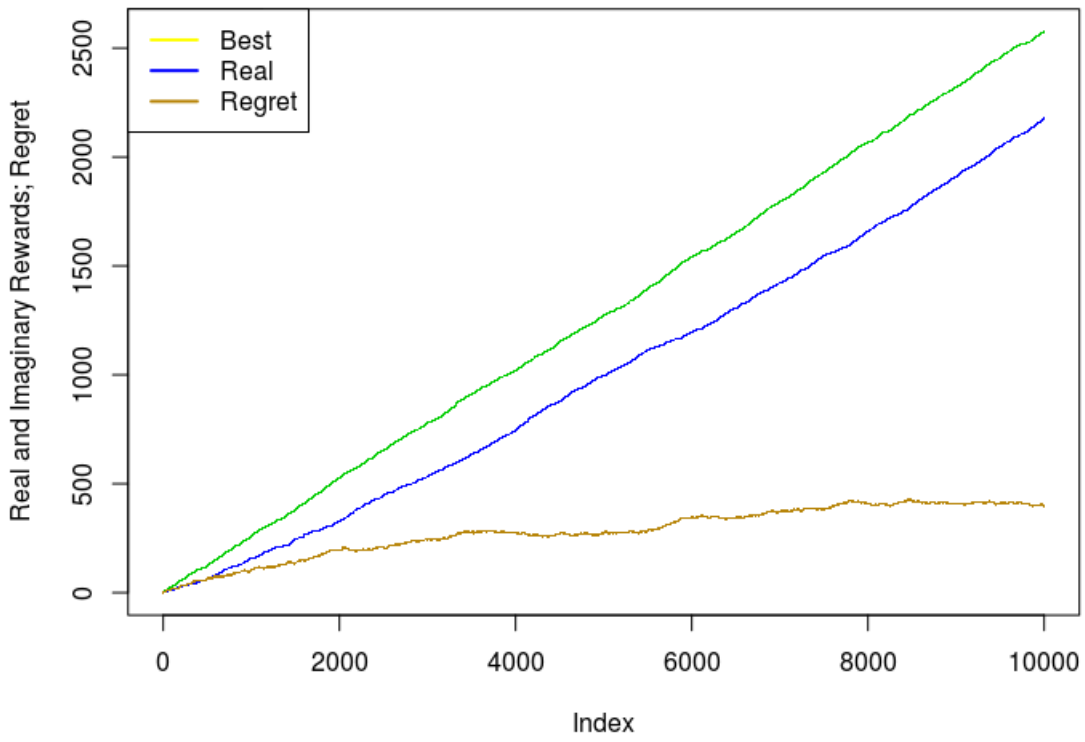
```
    ads_selected = append (ads_selected, ad)
    numbers_of_selections [ad] = numbers_of_selections [AD] + 1 = recompensa conjunto de dados [n, anúncio]

    sums_of_rewards [ad] = sums_of_rewards [ad] + recompensa total_reward = total_reward +
    recompensa
    rewards_at_each_step = C (rewards_at_each_step, total_reward) # minha disso
    best_rewards_at_each_step [n] = soma (best_selection [1: N]) # minha disso
}

#  curva de pesar
trama (best_rewards_at_each_step, PCH = " , Col = 3, principal = "Real e Imaginário
        Recompensas; Regret" , Ylab = "Números de recompensa" )

Os pontos (rewards_at_each_step, PCH = "" , Col = 4)
Os pontos ((best_rewards_at_each_step-rewards_at_each_step), PCH = " , Col = "Darkgoldenrod" )
lenda( 'TopLeft' , Legenda = C ( "Melhor" , "Real" , "Arrepender" ), col = c (7,4, "Darkgoldenrod" ), Horiz = F, LTY =
        1, LWD = 2)
```

E obtemos a seguinte trama:



6,2 Thompson Amostragem

6.2.1 Thompson Amostragem Intuition

Por que o amarelo marcar a melhor escolha, e não a marca verde?
A marca amarela é a melhor escolha porque é a mais afastada da origem no eixo x, o que significa, portanto, que ele tem o retorno mais alto estimado.

Como é Thompson Amostragem melhor do que UCB?
Thompson amostragem é melhor do que a UCB em termos de convergência do arrependimento. O arrependimento é o erência di ff entre a recompensa ideal e a recompensa que você acumular com o seu algoritmo. Thompson Amostragem mostra uma curva de pesar melhor do que UCB na minha experiência. Caso contrário, eu concordo com o ponto que você faz sobre UCB sendo determinista vs Thompson Amostragem sendo estocástica, tornando Thompson Amostragem superar UCB. Além disso você vai ver nas seções práticas que Thompson amostragem fi nds o melhor anúncio mais rápido e com mais certeza do que UCB.

Eu não entendo como Thompson Sampling pode aceitar o feedback atrasado. Por favor explique.
Ao fazer Thompson amostragem, ainda podemos realizar atualizações em nosso algoritmo (como fazer novos palpites para as distribuições com os dados existentes, a amostragem da distribuição adivinhado, etc), enquanto estamos aguardando os resultados de uma experiência no mundo real. Isso não iria dificultar nosso algoritmo de trabalhar. É por isso que ele pode aceitar o feedback atrasado.

Quais são outros exemplos de aplicações Thompson amostragem?
Outra aplicação potencial de bandidos Multi-armados (MAB) pode ser o teste on-line de algoritmos. Por exemplo, vamos supor que você estiver executando um site de comércio eletrônico e você tem à sua disposição vários algoritmos de Aprendizado de Máquina para fornecer recomendações para os utilizadores (de qualquer que seja o site está vendendo), mas você não sabe qual algoritmo leva a melhores recomendações .

Você poderia considerar o problema como um problema MAB e de ne fi cada algoritmo Aprendizado de Máquina como um "braço": a cada rodada, quando um usuário solicita uma recomendação, um braço (ou seja, um dos algoritmos) serão selecionados para fazer as recomendações, e você receberá uma recompensa. Neste caso, você pode de fi nir a sua recompensa de várias maneiras, um exemplo simples é "1" se o usuário clicar / compra um item e "0" caso contrário. Eventualmente seu algoritmo bandido irá convergir e acabam sempre escolhendo o algoritmo que é a mais e ffi ciente fornecer recomendações. Esta é uma boa maneira de encontrar o modelo mais adequado em um problema online. Outro exemplo que vem à minha mente está encontrando o melhor tratamento clínico para os pacientes: cada tratamento possível poderia ser considerado como um "braço", e uma forma simples de definir a recompensa seria um número entre 0 (o tratamento não tem Effect de todo) e um (o paciente é curado perfeitamente). Neste caso, o objetivo é encontrar o mais rápido possível o melhor tratamento, minimizando o pesar cumulativo (o que equivale a dizer que você quer evitar, tanto quanto possível selecionando "maus" ou mesmo sub-óptimos tratamentos durante o processo).

6.2.2 Thompson Amostragem em Python

Onde posso achar algum grande recurso sobre a distribuição Beta?
O melhor é o seguinte: Distribuição Beta

O histograma mostra 4 como o melhor anúncio para selecionar. A barra mais alta é a 4 no eixo x. Por que é dito como 5?

É só porque índices em Python começar a partir de 0. Assim, o anúncio do índice de 4-lo ad número 5.

Estou curioso para saber como seria de aplicar Thompson Amostragem de forma proativa ao executar esta campanha publicitária teórica. Você iteração o programa sobre cada rodada (ou seja, cada vez que tudo isso acrescenta foram apresentados para o usuário)?

Primeiro, um engenheiro de dados torna um gasoduto para ler os dados a partir do site e reagir a ela em tempo real. Em seguida, uma visita web desencadeia uma resposta para recalcular os nossos parâmetros e escolher um anúncio para a próxima vez.

6.2.3 Thompson Amostragem em R

Você poderia explicar em suas próprias palavras o ff rência di entre o Bernoulli e distribuição beta?

A distribuição de Bernoulli é aplicada às probabilidades discretas porque dá a probabilidade de sucesso de um resultado, enquanto que a distribuição Beta é aplicada a denso probabilidades (contínua), utilizando uma função de probabilidade densa. Essa é a principal rência di ff. Aqui estão alguns excelentes recursos: distribuição Beta distribuição de Bernoulli

Poderia fornecer uma grande fonte de Inferência Bayesiana?
Aqui é um grande: Bayesien Inference

Como é Thomson Amostragem heurística rapidamente capaz de descobrir que 5 anúncio é o melhor em comparação com o Alto Con fi ança heurística Bound?
É difícil explicar o motivo, teoricamente, que seria necessário para fazer a pesquisa e escrever uma prova matemática longa. Mas intuitivamente, pode ser porque UCB se baseia em pressupostos optimistas enquanto Thompson A amostragem é baseada em probabilidades relevantes através da abordagem bayesiana.

Como construir a curva Regret Thompson Amostragem em R?
Confira a última pergunta na UCB na seção R. Basicamente, você só precisa adicionar a seguinte no final do seu código:

```
rewards_at_each_step = C (rewards_at_each_step, total_reward)
    best_rewards_at_each_step [n] = soma (best_selection [1: N])}

trama (best_rewards_at_each_step, PCH = '' , Col = 3,
5           principais = "Real e Imaginário Recompensas; Regret" , Ylab = "Números de
            recompensa" )
Os pontos (rewards_at_each_step, PCH = "" , Col = 4)
Os pontos ((best_rewards_at_each_step-rewards_at_each_step), PCH = '' , Col = "Darkgoldenrod" ) lenda( 'TopLeft' , Legenda = C ( "Melhor" , "Real" , "Arrepender" ),

10              col = c (7,4, "Darkgoldenrod" ), Horiz = F, LTY = 1, LWD = 2)
```

7 Parte 7 - Processamento de Língua Natural

7.1 Natural Language Processing Intuition

Por que o saco do modelo Palavras substituir todas as letras maiúsculas pelos casos mais baixos?
Muitas linguagens de computador, em particular Python e R, tratar capitais e letras minúsculas como completamente di símbolos ff erent. Então, se nós não irá converter tudo para casos menores, em seguida, devemos levar em conta que algumas palavras podem conter letras maiúsculas e incluir código adicional para eles. Muito mais simples para substituir todas as capitais por casos menores.

O que é sparsity?
sparsity ocorre quando você tem lotes e lotes de zeros em sua matriz (portanto, chamado de matriz esparsa). Então, quando nós reduzimos sparsity, isso significa que reduzir a proporção de zeros na matriz.

7.2 Processamento de Língua Natural em Python

Poderia explicar o que 'citar = 3' significa?
citando = 3 irá ignorar as aspas duplas nos textos. O número de observações no conjunto de dados não corresponde ao número de textos que era devido a uma anomalia dividir com as aspas duplas. O problema potencial é que se você tem aspas duplas em uma revisão, esta revisão pode acidentalmente ser separados e considerada como outro comentário sem resultado. Então, para ter certeza de que evitar esse tipo de situação que ignorar as aspas duplas usando citando = 3.

Como podemos baixar todo o material NLTK em um tiro?
Para baixar todo o material em NLTK em um tiro que você precisa para abrir um terminal e execute:

```
pip instalar -U nltk
```

Poderia explicar o que é PorterStemmer ()?
PorterStemmer () aplica decorrentes para as palavras seus textos. Assim, por exemplo, "amado" "amor" vai se tornar após o desengace.

Como fazer PNL em francês ou outros?
Verifique se existem algumas classes de PNL específico para o seu próprio idioma, como é o caso em francês, por exemplo:

```
de nltk.stem.snowball importar FrenchStemmer Stemmer = FrenchStemmer ()
```

e para os stopwords, incluir isto em seu código:

```
Conjunto (stopwords.words ( 'francês' ))
```

Por que nós mudamos os comentários a partir de listas de palavras de volta para cordas?
Ele é feito porque o método fi t_transform () da classe CountVectorizer () requer cordas para trabalhar.

Qual é o ff rência di entre "TF-IDF" e "CountVectorizer"?
O erência di ff é que TF-IDF aplica normalização, ao contrário CountVectorizer. Caso contrário, é o mesmo, se você inserir o parâmetro 'normalizar = None', então é muito similar.
Para usar TF-IDF, é o mesmo que para as outras técnicas de PNL, você acabou de importar a classe que você quer (dizem T fi dfVectorizer de 'sklearn.feature_extraction.text'), então você cria um objeto desta classe, enquanto entrando com os parâmetros desejados para pré-processar seus textos.

Por que excluir 'NÃO', dado que "Crust é bom" não é o mesmo que "Crosta não é bom"?
Ele também funciona ao contrário: "Crust é ruim" não é o mesmo que "Crosta não é ruim". Assim, este geralmente não é levado em conta. Por isso acreditamos / espero que, em média, o número de equívocos causados por esta seria menos no total.

Depois que é feito com a limpeza e obter o nosso saco de modelo de palavras, por que estamos fazendo classi fi cação?

Estamos fazendo classi fi cação nos comentários para prever o resultado de novas, exatamente como análise de sentimentos. Só para fazer essas previsões nas melhores condições, precisamos aplicar o saco de palavras modelo de primeira. Que vai preparar os dados em um formato correto para que possa ser enquadrados com a classi fi cador.

O modelo parece ser a melhor solução para o Desafio dos trabalhos de casa?
De fi nitivamente Naive Bayes, que por sinal é geralmente recomendado primeiro ao fazer Processamento de Língua Natural.

7.3 Processamento de Língua Natural em R

Por que remover a pontuação do texto? No primeiro parece que eles não contêm informações. Mas por que não vamos deixar algoritmo fi cação classi de decidir?
Geralmente é um trade-o ff entre dimensionalidade e informações. Nós remover pontuação para tornar o conjunto de dados mais gerenciável com menos dimensões.
Além disso, a pontuação não é realmente levar o máximo de informação discriminatória como palavras processos usuais fi cação classi.

Através da experiência, a validação cruzada tem apontado para a remoção de pontuação melhor desempenho. No entanto sim, podemos deixar o algoritmo cação classi fi decidir. Talvez alguns conjuntos de dados têm melhor desempenho com pontuação, mas como regra geral, a pontuação é melhor ser evitado.

"Crosta não é bom" está sendo transformado em "crosta bom" depois de remover palavras irrelevantes. A revisão está mudando para positiva de negativa. pode esclarecer como isso vai impactar aprendizagem e previsão?

A palavra "não" não é tão útil quanto parece. Considere as seguintes 2 comentários: "Eu não pedi sua
pasta O seu pão era bom.". "Eu pedi a sua massa. O seu pão não era bom."

Do "saco de palavras" abordagem eles são os mesmos, embora o seu significado é muito erent di ff. Normalmente quando as pessoas querem tomar "não" na conta que iria substituir a palavra seguinte com antônimo e remover a negação.

Existe alguma maneira ou de recursos através do qual eu posso dar uma olhada na lista das palavras não relevantes?

Você pode vê-los entrando em R o seguinte comando:

```
biblioteca (tm) stopwords ( "Inglês" )
```

O que seria uma outra técnica de matriz de confusão e curva CAP para testar o modelo PNL?

Outro e melhor técnica é k-Fold validação cruzada, que você vai ver na Parte 10 - Selecção do modelo.

8 Parte 8 - Aprendizagem Profunda

8.1 Arti fi cial Redes Neurais

8.1.1 Arti fi cial Neural Networks Intuition

Em que categoria não ANN cair? Supervisionada, não supervisionada, reforçado de aprendizagem?
RNAs pode ser tudo 3. Depende de como você implementar o modelo. Exemplos: Aprendizagem
Supervisionada: CNNs classificar imagens em IMAGEnet.
Aprendizagem não supervisionada: Boltzmann Machines, AutoEncoders, GAN, DC-Gans, VAE, SOMS, etc. Reforço: Deep Convolucional Q-Learning que joga videogames desde a entrada pixel, AlphaGO, etc. Este ramo é chamado "Deep Reinforcement Learning" e pertence inteligência artificial".

Como é que a Rede Neural figura para fora o que os pesos ótimos são. Quando isso vai parar a sua aprendizagem?

É figuras fora os pesos ideais com base num algoritmo de optimização empregues para minimizar a perda de função
wrt todos os pontos de dados de entrada. Ele pára quando a perda de formação vai muito próximo de 0. Além nós geralmente escolher um determinado número de épocas que é o número de vezes que a rede neural é treinada. Depois da última época, o treinamento é longo. Há também o que chamamos de "interrupção precoce", que é uma técnica que impede a formação uma vez que temos uma muito pequena redução de perdas em uma validação de conjunto sobre as épocas. Nesse caso, o treinamento pára antes da última época.

Por que a função de custo pode ser reduzido?
A função de custo é uma estimativa numérica de quão errado as nossas previsões de rede neural são. Se reduzirmos a função de custo, isso significa que estamos a reduzir os erros cometidos pela rede neural, em vez tornando-se prever com maior precisão.

Como é o peso calculado para cada neurônio?
No início do treinamento, os pesos são inicializados perto de zero. Em seguida, sobre cada uma das várias iterações (ou épocas), os pesos são actualizados por meio de gradiente descendente, em direcção que diminui a mais do (ou função de custo) erro perda entre as previsões e os alvos.

Poderia, por favor recapitular o processo de cada iteração de treinamento?
Após os pesos são inicializados aleatoriamente com valores próximos a zero, o treinamento vai ao longo de muitas iterações (o número de iterações, também chamado o número de épocas, normalmente é decidida na implementação). Aqui está o processo de cada iteração:

1. Adiante Propagação: a entrada é propagada para a frente no interior da rede neural que, no final retorna a predição.

2. Comparamos que a previsão para o alvo (o valor real que temos porque estamos lidando com o conjunto de treinamento) e calculamos o erro perda entre a previsão eo alvo.

3. Que erro de perda é back-propagado dentro da rede neural.

4. Através Gradient Descent ou Stochastic Gradient Descent, os pesos são atualizados nas direções que reduzem ao máximo o erro perda (uma vez que, de facto, o erro perda é uma função de custo dos pesos).

5. Nós, assim, obter novos pesos, pronto para fazer uma nova previsão para a próxima iteração. Em seguida, o mesmo processo todo vai outra vez, até que o erro perda pára reduzindo (início de parar) ou até que o treinamento atinge a última época.

8.1.2 Arti fi cial Redes Neurais em Python

Por que um ANN ser um modelo / solução melhor do que apenas usando um dos modelos di ff erent que aprendemos na parte 3?

Não é necessariamente a melhor solução para todos os problemas. Por exemplo, se o seu problema é linear, então este é de fi nitivamente não a melhor solução desde que você seria muito mais e ffi ciente com um modelo de regressão linear. No entanto, se o seu problema é não linear e complexo, então ANN pode ser um modelo / solução melhor do que os modelos na Parte 3. Na Parte 10 - Selecção do modelo você vai aprender como avaliar isso.

Por que estamos usando Sequential e denso?

Basicamente, é muito simples:

- Sequencial é usado para inicializar o modelo de aprendizagem profunda como uma sequência de camadas (em oposição a um gráfico computacional).

- Denso é usado para adicionar uma camada de neurónios na rede neural.

Recebo alguns avisos quando eu executar o código da seção onde eu adicionar as camadas com a função Densa (). Como posso fi x isso?

Você simplesmente precisa substituir:

```
classifier.add (Densa (output_dim = 6,
                       o init = 'uniforme' , Activação = 'Relu' ,
                       Input_dim = 11))
```

de:

```
classifier.add (densas (unidades = 6,
                        kernel_initializer = "uniforme" , Activação = "Relu" ,
                        Input_dim = 11))
```

Basicamente alguns nomes de parâmetros alterados na API: 'output_dim' se tornou 'unidades', 'init' se tornou 'kernel_initializer' e 'nb_epoch' se tornou 'épocas'. Isso acontece frequentemente em bibliotecas.

Como podemos decidir o número de camadas ocultas?

Não há nenhuma regra de ouro e geralmente camadas mais ocultos pode dar-lhe maior precisão, mas também pode dar-lhe mais de fi tting então você precisa ser cauteloso. A chave é a intuição e experiência. Você também pode experimentar com ajuste dos parâmetros de que você vai ver na Parte 10 - Selecção do modelo.

O que faz a função de ativação fi er retos fazer?

Desde o aprendizado profundo é usado para resolver problemas não lineares, os modelos precisam ser não linear. E o uso da função fi er retos é realmente torná-lo não linear. Aplicando-o você está quebrando a linearidade entre os neurônios de saída e os neurônios de entrada.

O que eu preciso para mudar se eu tiver uma saída não-binário, por exemplo, ganhar, perder e desenhar?

Nesse caso, você precisaria criar três variáveis dummy para sua variável dependente: (1,0,0) → win (0,1,0) → perder (0,0,1) → desenhar

e, portanto, você precisa fazer a seguinte alteração:

```
output_dim = 3 # agora com a nova API é: unidades = 3
activação = 'Softmax'
perda = 'Categorical_crossentropy'
```

nb_epoch foi atualizado?

Sim, em seu código que você precisa para substituir 'nb_epoch' por 'épocas'.

Como construir a mesma rede neural para a regressão?

Primeiro você precisa importar:

```
KerasRegressor
```

Então é quase o mesmo que uma RNA para Classi fi cação. A principal rência di ff é a função de compilação, onde é claro que precisamos mudar a função de perda e dar uma função de perda de regressão como o erro médio quadrático, e remover a métrica precisão:

```
# Inicialização do ANN
regressor = sequencial ()

# Adicionando a camada de entrada e a primeira camada oculta
regressor.add (densas (unidades = 6,
                       kernel_initializer = 'uniforme' , Activação = 'Relu' , Input_dim
                       = 11))


# Adicionando a segunda camada oculta
regressor.add (densas (unidades = 6,
                       kernel_initializer = 'uniforme' , Activação = 'Relu' ))


# Adicionando a camada de saída
regressor.add (densas (unidades = 1,
                       kernel_initializer = 'uniforme' ))

# Compilando o ANN
regressor.compile (optimizador = 'Adão' , Perda = 'Mean_squared_error' )

# Montagem da ANN para o conjunto de treinamento
regressor.fit (X_train,
               y_train,
               batch_size = 10, épocas =
               100)
```

Onde todos os hiperparâmetros vem? Como é que nós escolher o número de camadas, o número de neurônios em cada camada, e todos os outros parâmetros?

Através de muita pesquisa e experimentação. Não há nenhuma regra de ouro como escolher esses números. Normalmente você pode achar algum pronto para usar neural redes arquiteturas on-line que provar para obter bons resultados. Mas você pode experimentar por meio do ajuste seu modelo manualmente com outros valores de parâmetros. E, por último, você pode usar algumas técnicas de ajuste dos parâmetros, como Grade Pesquisa com k-Fold Cruz Validação Para encontrar os valores ideais. Você vai aprender como fazer isso na Parte 10 - Selecção do modelo.

Como posso melhorar a precisão da ANN?

Esta é a pergunta do milhão de dólares. cientista de dados têm vindo a tentar figura fora esta resposta por um tempo. Normalmente modelos ANN tomar uma boa quantidade de tuning, este é geralmente sob a forma de mudar número de camadas ocultas, alterando fórmulas e dados de normalização. Normalmente estas coisas, poderá ir mais longe.

8.1.3 Arti fi ciais Redes Neurais em R Por que usamos

'as.numeric (factor ())'?
Uma vez que para construir a rede neural que, então, utilizar o pacote de H2O que espera que as entradas como factores numéricos (ou numéricos variáveis categóricas).

Se eu optar escondida = *c (uma*$_1$, *uma*$_2$, *uma*$_3$ .. *uma*$_n$), isso significa existem *n* camadas ocultas no modelo, e *uma*$_1$, *uma*$_2$, ..., *uma*$_n$ são os números de nós em cada camada escondida?
Sim esta correto!

É o número de nós em cada camada oculta sempre o mesmo? Ou podemos especificar di nós ff erent em cada camada escondida? Como é que vamos escolher esses números?
Não, você pode brincar com ele e ver se você pode obter melhores resultados. Aconselhamos a ver o seu gerenciador de tarefas, porém, para ver se a memória do computador é su ffi ciente para lidar com tantas camadas e seus nós. Você pode escolhê-los por qualquer experimentando manualmente (tentativa e erro) ou através de ajuste dos parâmetros com Grid Search (ver Parte 10 - Selecção do modelo).

Em Python foi aplicado o Recti fi er Função para as camadas ocultas, mas a função sigmóide para a camada Resultado. Como escolher as mesmas opções com R?
Utilizando H2O em R, a Função Recti fi cador foi aplicado a cada camada. Aqui não há opção de escolher funções de ativação erent di ff para as camadas erent di ff.

Python ou R para Deep Aprendizagem?
Python. Sem dúvida. Python é usado por todos os melhores cientistas profunda Aprendizagem hoje e tem bibliotecas surpreendentes (Tensor fluxo, Keras, PyTorch) desenvolvido para aplicações poderosas (Imagem Classi fi cação, Computer Vision, Arti fi cial inteligência, chatbots, máquina de tradução, etc.).

8.2 Convolucionais Redes Neurais

8.2.1 Convolucional Neural Networks Intuition

Quais são os rências di ff entre a CNN ea ANN? É a CNN aplicada a imagem classi fi cação única?

ANN significa redes neurais em geral. Assim, um CNN é uma ANN. CNNs não são utilizados apenas para processamento de imagens, eles também podem ser aplicados ao texto como Compreensão de Texto.
Para baixo a quanto o tamanho da imagem será reduzida para o detector recurso?
Na secção prático que irá ser reduzida até 64 por 64 dimensões.

Qual é o objetivo dos mapas de características?
Estamos construindo mapas de características com cada fi ltro convolucional, o que significa que estamos a fazer características que nos ajudam a classificar o objeto. O exemplo mostrado na Palestra intuição é como uma borda unidimensional detecção fi ltro. Esse é um mapa recurso. Então, queremos que o nosso modelo para "ativar" em um mapa recurso só onde há uma borda. Teremos vários mapas de características como este que, quando todos juntos vai nos ajudar a identificar o objeto. Isto é ajudado pela remoção do preto ou os valores negativos.

Qual é o propósito do Relu?
A maior razão pela qual usamos Relu é porque queremos aumentar a não-linearidade à nossa imagem. E Relu atua como uma função que quebra a linearidade. E a razão pela qual queremos quebrar a linearidade em nossa rede é porque próprias imagens são altamente não-linear. Na verdade, eles têm um monte de elementos não lineares, como as transições entre pixels.

Por que Max-Pooling considerar apenas 4 valores para tomar um máximo de e não 2 ou 8 valores? Também em convolução como é o detector recurso formado?
Porque um máximo é tomada de um ponto na imagem que é representado por um quadrado de 2x2 sobre a nossa imagem. Por isso, abrange 4 pixels.

Depois attening fl, vamos obter um longo vector para todas as camadas agrupadas ou um vetor para cada camada agrupada?

Será um longo vector recolhendo todas as camadas sejam combinadas.

8.2.2 Convolucionais Redes Neurais em Python

Quanto imagens são obrigados a treinar um bom modelo? Existe alguma regra de ouro para esta decisão?

10.000 é um bom número. Nenhuma regra de polegares, apenas quanto mais você tem, mais chances você terá de obter uma boa precisão.

Poderia explicar os números 0, 1, 2, 3 e 4 nos mapas de recursos?
Quando o detector recurso 3x3 está cobrindo uma parte da imagem de entrada, torna-se:
0 se não houver um em comum entre a subtabela 3x3 da imagem de entrada e o detector de recurso 3x3, 1 se houver um 1 em comum entre a subtabela 3x3 da imagem de entrada e o detector de recurso 3x3, 2, se houver dois 1 em comum entre a subtabela 3x3 da imagem de entrada e o detector de recurso 3x3, 3 se houver três 1 em comum entre a subtabela 3x3 da imagem de entrada e o detector de recurso 3x3, 4, se houver quatro 1 em comum entre a subtabela 3x3 da imagem de entrada eo detector recurso 3x3.

Poderia, por favor recapitular a propagação frente das imagens de entrada que acontecem dentro da CNN por descrever em poucas palavras as classes que usamos em Python?
Claro, aqui é o processo com as descrições essenciais das classes usadas:
Sequencial é primeiro utilizado para especificar que introduzir uma sequência de camadas, em oposição a um gráfico computacional. Convolution2D é então usado para adicionar a camada convolucional. MaxPooling2D é então usado para aplicar Max Pooling para as imagens de entrada. Achatar é então utilizado para fl atten as imagens reunidas. Densa é usada para adicionar a camada de saída com softmax.

Como fazer uma única previsão se uma imagem especi fi c contém um gato ou um cão?
Dentro da pasta conjunto de dados que você precisa para criar uma pasta adicional separada (vamos chamá-lo "single_prediction") que contém a imagem (vamos chamá-lo "cat_or_dog.jpg") que pretende prever e execute o seguinte código:

```
importar numpy como NP de keras.preprocessing importar imagem como image_utils test_image = image_utils.load_img ( 'Dataset /
single_prediction / cat_or_dog.jpg' ,

                                                    target_size = (64, 64))
test_image = image_utils.img_to_array (test_image)
test_image = np.expand_dims (test_image, eixo = 0) = resultado (classifier.predict_on_batch
test_image) training_set.class_indices

 E se resultar [0] [0] == 1:
        predição = 'cachorro'
outro :
        predição = 'gato'
```

9 Parte 9 - dimensionalidade Redução

9.1 Análise de Componentes Principais (PCA)

9.1.1 PCA Intuition

Qual é o verdadeiro propósito da PCA?
O verdadeiro propósito é principalmente para diminuir a complexidade do modelo. É para simplificar o modelo, mantendo relevância e desempenho. Às vezes você pode ter conjuntos de dados com centenas de recursos para que, nesse caso, você só quer extrair muito menos variáveis independentes que explicam a maior variância.

Qual é o ff rência di entre PCA e Análise de Fator?
Análise de componentes principais envolve extração de compostos lineares de variáveis observadas. A análise fatorial é baseado em um modelo formal prever variáveis observadas de fatores latentes teóricas. PCA se destina a maximizar a variância total para procurar padrões distintos e análise fatorial olha para maximizar a variância compartilhada para construtos latentes ou variáveis.

Devo aplicar PCA se o meu conjunto de dados tem variáveis categóricas?
Você poderia tentar PCA, mas eu gostaria de ter muito cuidado, porque os valores categóricos podem ter altas variâncias por padrão e geralmente será instável à matriz inversão.
Aplicar PCA e fazer a validação cruzada para ver se ele pode generalizar melhor do que os dados reais. Se isso acontecer, então PCA é bom para o seu modelo. (Sua matriz de formação é numericamente estável). No entanto, estou certo de que na maioria dos casos, PCA não funciona bem em conjuntos de dados que contêm apenas os dados categóricos. Vanilla PCA é projetado com base em capturar a covariância de variáveis contínuas. Existem outros métodos de redução de dados que você pode tentar comprimir os dados, como análise de correspondência múltipla e categórica PCA etc.

Qual é o melhor recurso extra em PCA?
Confira este vídeo que tem uma explicação surpreendente de PCA e estuda-la com mais profundidade.

9.1.2 PCA em Python

O que a fi t_transform fazer aqui? Por que aplicar fi t_transform ao conjunto de treinamento, e só transformar o conjunto de teste?

No método t_transform fi existe fi t e transformar.
A parte fi t é usado para analisar os dados em que aplicamos o objeto (recebendo os valores eigen e os eigenvetores da matriz de covariância, etc.), a fim de obter as informações necessárias para aplicar a transformação PCA, ou seja, extrair alguns os principais recursos que explicam a maior variância. Então uma vez que o objeto recebe essas informações graças ao método fi t, a parte transformar é usado para aplicar a transformação PCA.

E uma vez que o conjunto de teste e o conjunto de treinamento têm estruturas muito semelhantes, não precisamos para criar um novo objeto que encaixa ao conjunto de teste e, em seguida, usar para transformar o conjunto de teste, podemos usar diretamente o objeto já criado e fi enquadrados para o conjunto de treinamento, para transformar o conjunto de teste.

Como você achar fora que das variáveis independentes são os 2 principais componentes?
PCA é uma técnica de extração de características de modo que os componentes não são um queridos das variáveis independentes originais. Estes são novos, como algum tipo de transformações dos originais. É somente com a seleção de recurso que você acabar com os das variáveis independentes originais, como com eliminação retrógrada.

É melhor usar Feature Extraction ou Seleção de recursos, ou ambos? Se ambos, em que ordem?
Extração de características e Seleção de Recursos são duas técnicas grande redução de dimensionalidade, e, portanto, você deve sempre considerar ambos.
O que eu recomendo é fazer Seleção de Recursos primeiro para só manter as características relevantes, e depois aplicar Feature Extraction sobre esses recursos relevantes selecionados para reduzir ainda mais a dimensionalidade do conjunto de dados, mantendo variância suficiente.

Quanto proporção da variância total que precisamos de usar? Existe algum limite para um bom rácio total de variância?

De um modo geral uma boa limiar é de 50%. Mas 60% é mais recomendado.

É mais comum usar exatamente 2 variáveis independentes para construir uma fi classificador, ou que as pessoas normalmente usam mais do que isso?
Em geral as pessoas apenas extrair um número de variáveis independentes que explicam a su ffi ciente proporção da variância (normalmente 60%). Por isso, nem sempre é dois. Pode ser mais. E se é dois que é ótimo, porque então você pode visualizar melhor.

Existe um método Python ou atributo que pode ajudar a fornecer os componentes subjacentes e sinais da PC1 dois componentes principais e PC2?
Claro que você pode acessá-los com os atributos components_:

```
componentes = pca.components_
```

9.1.3 PCA em R

Se eu queria ver os valores reais para todas as colunas, como eu iria unscale os dados que foi característica escalado?

Dizer que o seu vector escalado foi y_train, em seguida, para unscale seu resultado y_pred faça o seguinte:

```
y_pred_unscaled = y_pred * attr (y_train, 'Escalado: Escala' ) + Atr (y_train, 'Escalado: Centro' )
```

Como posso ver os principais componentes em R?
Você pode executar o seguinte código:

```
pca $ rotação
```

Temos uma precisão de 100%, não deveríamos estar preocupados de mais de fi tting?
Se você olhar para os dados, você vai ver que é quase perfeitamente separáveis pelas 3 linhas traçadas, o que significa que a separabilidade é uma característica dos dados em vez de um problema tting sobre fi.
Se você tinha algo como 50 linhas e 100% de precisão (ou seja, cada seção seria capturar precisamente uma pequena quantidade de pontos, garantindo que eles são justamente classi fi cado), que provavelmente seria um excesso questão tting fi, mas aqui isso não é claramente o caso .

Além obtivemos 100% de precisão apenas no conjunto de teste. Havia alguns pontos fi cados mis-classificadas no conjunto de treinamento. E mais fi tting é justamente o contrário: uma precisão quase perfeita no conjunto de treinamento e um pobre no conjunto de teste.

Como podemos saber quais são as duas variáveis tomadas para traçar?
As duas variáveis não estão entre suas variáveis independentes originais, uma vez que eles foram extraídos (Extracção de funcionalidades), em vez de ser selecionada (Seleção de Recursos). As duas novas variáveis são as direções em que há mais variância, ou seja as direções onde os dados são mais espalhadas.

9.2 análise discriminante linear (LDA)

9.2.1 LDA Intuition

Você poderia explicar de forma mais simples o ff rência di entre PCA e LDA?
Uma maneira simples de visualizar o erência di ff entre APC e LDA é que APC trata os dados de todo o conjunto como um todo, enquanto LDA tentativas para modelar o di ff rências entre classes, dentro dos dados. Além disso, PCA extrai alguns componentes que explicam a maior variância, enquanto LDA extrai alguns componentes que maximizam classe separability.

Seleção de recursos ou Extração recurso?
Você prefere escolher a seleção recurso se você quer manter toda a interpretação do seu problema, seu conjunto de dados e os resultados do modelo. Mas se você não se preocupam com a interpretação e único carro sobre a obtenção de previsões precisas, então você pode tentar ambos, separadamente ou em conjunto, e comparar os resultados de desempenho. Então, sim seleção de recursos e extração de características podem ser aplicadas simultaneamente em um determinado problema.

Podemos usar LDA para a regressão?
LDA é Linear Análise Discriminante. É uma generalização do discriminante linear de Fisher, um método utilizado em estatísticas, reconhecimento de padrões e aprendizagem de máquina para encontrar uma combinação linear de recursos que caracteriza ou separa duas ou mais classes de objetos ou eventos. A combinação resultante pode ser utilizado como uma classi fi cador linear, ou, mais comumente, para redução dimensionalidade antes depois classi fi cação. No entanto, para a regressão, temos que usar ANOVA, uma variação de LDA. LDA também está intimamente relacionada com a análise de componentes principais (PCA) e análise fatorial em que eles tanto olhar para combinações lineares de variáveis que melhor explicam os dados. LDA tenta explicitamente para modelar a erência di ff entre as classes de dados. O PCA por outro lado, não leva em conta qualquer ff rência di em sala de aula, e análise factorial constrói as combinações de características com base na rências di ff em vez de semelhanças. A análise discriminante é também di ff erent de análise factorial na medida em que não é uma técnica interdependência: uma distinção entre as variáveis independentes e variáveis dependentes (também chamado variáveis de critério) deve ser feita. LDA funciona quando as medições feitas em variáveis independentes para cada observação são quantidades contínuas. Ao lidar com variáveis independentes categóricas, a técnica equivalente é análise de correspondência discriminante. LDA funciona quando as medições feitas em variáveis independentes para cada observação são quantidades contínuas. Ao lidar com variáveis independentes categóricas, a técnica equivalente é análise de correspondência discriminante. LDA funciona quando as medições feitas em variáveis independentes para cada observação são quantidades contínuas. Ao lidar com variáveis independentes categóricas, a técnica equivalente é análise de correspondência discriminante.

9.2.2 LDA em Python

Quais as variáveis independentes são encontrados após a aplicação LDA?
As duas variáveis independentes que você vê, indexados por 0 e 1, são novas variáveis independentes que não estão entre as 12 variáveis independentes originais. Estes são totalmente novas variáveis independentes que foram extraídos através de LDA, e é por isso que chamamos LDA Feature Extraction, em oposição ao recurso de seleção onde você manter algumas de suas variáveis independentes originais.

Como decidir o parâmetro n_component LDA, a fim de encontrar o resultado mais preciso?
Você pode correr:

```
LDA (n \ _components = Nenhum)
```

e deve dar-lhe automaticamente os n_components ideais.

Como posso obter os dois Linear Discriminantes LD1 e LD2 em Python?
Você pode obtê-los, executando a seguinte linha de código:

```
lda.scalings_
```

9.2.3 LDA em R

Como posso obter os dois Linear Discriminantes LD1 e LD2 em R?
Você pode obtê-los, executando a seguinte linha de código:

```
lda $ escalonamento
```

Eu não começ completamente por que a versão R da LDA pegou 2 variáveis independentes automaticamente?
Porque em R, quando você tem aulas de k, k-1 é igual ao número de discriminantes lineares que você recebe.

Por que estamos recebendo matriz no LDA, mas com uma trama de dados no PCA, mesmo que o procedimento é semelhante?

Esta é apenas devido a um erência di ff no formato que a função fornece como saída. Estes procedimentos são semelhantes independentemente do formato que está de saída.

9,3 Kernel APC

9.3.1 Kernel PCA Intuition

Deve Kernel PCA ser usado para converter dados não-linearmente separáveis em dados linearmente separáveis?

É isso mesmo, mas você não precisa usar Kernel PCA com uma classi fi er não linear desde que os dados serão linearmente separáveis após a aplicação Kernel PCA, e, portanto, uma classi fi er linear será su ffi ciente.

Quando devemos usar PCA vs Kernel PCA?
Você deve começar com PCA. Então, se você obter resultados pobres, tente Kernel PCA.

9.3.2 Kernel PCA em Python

Como posso saber se meus dados estão linearmente separáveis ou não?
Um bom truque é para treinar um modelo de regressão logística nele primeiro. Se você receber uma realmente boa precisão, deve ser (quase) linearmente separáveis.

Existe uma enorme rência di ff eo que é melhor usar entre Kernel PCA + SVM vs PCA + Kernel SVM?

Sim, há uma rência di ff.
Use Kernel PCA + SVM quando você pode transformar seus dados em uma variedade de baixa dimensionalidade não-linear, onde os pontos são separáveis.
Use PCA + Kernel SVM quando você precisa para transformar os seus dados através de uma transformação linear em um colector de baixa dimensionalidade, utilizando esses pontos para ser transformado em um espaço não-linear onde eles são separáveis.

Como é que vamos decidir qual kernel é melhor para Kernel PCA?
O RBF Kernel é um grande kernel, e é a melhor opção em geral. Mas a melhor maneira de figura para fora o que kernel que você precisa aplicar é fazer algum ajuste dos parâmetros com Grid Pesquisa e k-Fold Validação Cruzada. Veremos que na Parte 10 - Selecção do modelo.

9.3.3 Kernel PCA em R

Como posso obter os principais componentes do Kernel PCA em R?
Você só precisa executar a seguinte linha de código:

```
attr (kpca, "PCV" )
```

10 Parte 10 - Selecção do modelo e impulsionar

Validação cruzada 10,1 k-dobra

10.1.1 k-Fold Cruz Validação Intuição O que é baixa / alta viés /

variância?
Estes conceitos são importantes para entender K-Fold Cruz Validação: Low Viés é quando suas previsões do modelo são muito próximos dos valores reais. Alta Viés é quando suas previsões do modelo estão longe dos valores reais.
Baixa Variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação não irá variar muito.

Alta variação: quando você executar seu modelo várias vezes, as previsões erent di ff de seus pontos de observação irá variar muito.

O k-Fold Cruz Validação melhorar o modelo ou é apenas um método de validação?
k-Fold Cruz Validação é usado para avaliar o seu modelo. Isso não significa necessariamente melhorar o seu modelo, mas melhora a sua compreensão do modelo. No entanto, você pode usá-lo para melhorar o seu modelo, combinando-a com algumas técnicas de ajuste dos parâmetros, como Grade Search (próxima seção).

Qual é o ff rência di entre um parâmetro e uma hiperparâmetro?
parâmetros hiper e parâmetros são muito semelhantes, mas não exatamente a mesma coisa. Um parâmetro é uma variável gurable con fi que é interno a um modelo cujo valor pode ser calculado a partir dos dados. A hiperparâmetro é uma con fi valor gurable externo a um modelo cujo valor não pode ser determinado pelos dados, e que estamos tentando otimizar (fi nd o valor ideal) através de técnicas de ajuste dos parâmetros, como Aleatório Pesquisa ou Grade Search.

O que é um bom / melhor valor de k para escolher quando realizar k-Fold Cruz Validação?
Recomendamos vivamente 10.

10.1.2 k-Fold validação cruzada em Python

Fiquei me perguntando se precisávamos dividir os dados em X_train, y_train e X_test, y_test aplicar o método de reamostragem (k-Fold) ou podemos aplicá-lo diretamente sobre X e Y?
Você pode fazer as duas coisas. O bom método é:

1. Dividir o seu conjunto de dados em um conjunto de treinamento e um conjunto de teste.
2. Efectuar a validação cruzada k vezes sobre o conjunto de treino.
3. Faça a avaliação fi nal do seu modelo selecionado no conjunto de teste.

Mas você também pode executar k vezes Cross-Validação em todo o conjunto de dados (x, y).

O que isso Desvio padrão nos dizer exatamente?
O desvio padrão de precisão do modelo simplesmente mostra a variação da precisão do modelo é de 6%. Isto significa que o modelo pode variar de cerca de 6%, o que significa que, se eu executar o meu modelo em novos dados e obter uma precisão de 86%, eu sei que isso é como uma precisão de 80-92%. Viés e precisão, por vezes, não têm uma relação óbvia, mas na maioria das vezes você pode manchar algum viés na validação ou testes de seu modelo quando não pré-forma corretamente em novos dados.

Como calcular a pontuação F1, Recall ou precisão de k-Fold Cruz Validação?

Você pode usar a biblioteca métricas de sklearn para isso. Aqui está um link.

Por que só Kernel SVM?

Você pode aplicar k-Fold Cruz de validação para qualquer modelo de aprendizagem de máquina, e seria muito inteligente para fazer isso para cada modelo que você criar. Kernel SVM foi apenas um exemplo aqui para explicar como aplicar k-Fold Validação Cruzada.

10.1.3 k-Fold validação cruzada em R

Utilizou-se a matriz de confusão para determinar a precisão para cada dobra. O que você usaria para modelos de regressão? R-quadrado?

Não preferimos usar o MSE (Quadrado Médio do Erro), que mede a soma do quadrado di ff rências entre os resultados reais e as previsões.

O exemplo validação cruzada k vezes em R 10 cria modelos di ff erent utilizando o conjunto de treino. Como pode a média dos modelos 10 di ff erent avaliar o modelo original? Será que precisa mesmo o modelo original?

Através k-Fold Validação Cruzada, precisamos estimar o "erro imparcial" sobre o conjunto de dados a partir deste modelo. A média é uma boa estimativa de como o modelo vai realizar no conjunto de dados. Então uma vez que estamos convencidos de que o erro médio é aceitável, nós treinamos o mesmo modelo em todo o conjunto de dados.

Em R criamos dobras do conjunto de treinamento. não devemos criá-los para todo o conjunto de dados? Por que apenas o conjunto de treinamento?

Nós só usamos o conjunto de treinamento para fazer o nosso modelo. O conjunto de teste é considerado como algo que não tem com a gente ainda. (Mesmo que nós fazemos). Então, nós usamos o conjunto de treinamento para fazer o melhor modelo e, em seguida, avaliar o seu desempenho no conjunto de teste. Se ajustar o desempenho do conjunto de teste, em seguida, temos um modelo tendencioso para um bom desempenho no teste-set, que é apenas um pequeno exemplo de toda a dados. Precisamos nosso modelo para um bom desempenho para os dados que nós não vimos ainda bem.

10,2 Grade Pesquisa

10.2.1 Grade Pesquisa em Python

Não consigo importar 'GridSearchCV' no meu computador. Quaisquer outras opções para este pacote?
Depende da versão do sistema e um pacote, mas tentar substituir:

```
de sklearn.model_selection importar GridSearchCV
```

de:

```
de sklearn.cross_validation importar GridSearchCV
```

O que é este parâmetro de penalidade C?

O parâmetro C pena é um parâmetro de regularização, que pode permitir que você faça duas coisas:

- reduzir o excesso fi tting,
- substituir valores discrepantes. É igual a $\frac{1}{\lambda}$ onde λ é o parâmetro clássico regularização usado em Ridge Regression, Lasso, Elastic Líquido.

Pode Grade Pesquisa ser estendido para ANN também?
Sim, pesquisa Grid é possível em ANN também. Nós cobri-lo em nosso curso "Deep Aprendizagem AZ".

Como sabemos que valoriza devemos testar na grelha de Pesquisa?
Um bom começo é tomar valores padrão e experimentar com valores em torno deles. Por exemplo, o valor padrão do parâmetro de penalidade C é de 10, por isso alguns valores relevantes para tentar seria 1, 10 e 100.

Nesta seção Python, estamos ajustando os parâmetros em grade Pesquisa para produzir a precisão ótima no conjunto de treinamento com k-Fold Cruz Validação? Como podemos assegurar que estes parâmetros será o mais ideal quando se trabalha com o conjunto de teste?

Estes parâmetros serão mais óptima quando se trabalha com o equipamento de teste, porque eles provou ser óptima em conjuntos de teste ff erent dez di (as dez pregas conjunto de teste no processo de validação cruzada).

10.2.2 Grade Pesquisa em R

Como usar C e sigma valor calculado na grade Busca?
Depois de ter encontrado o melhor C e valor sigma, você pode colocar esses valores em seus SVM argumentos modelo e obter uma precisão ideal para o seu modelo.

Quais são as vantagens e desvantagens do uso de acento circunflexo em R?
Vantagens de acento circunflexo:

1. Ao fazer Grade Pesquisa você não tem que escolher valores de hiperparâmetros tentar-se, ele vai achar alguns mais ideais para você.
2. Tem um monte de outras funções além de ser um método invólucro Aprendizado de Máquina conveniente.
3. Tem a sua própria divisão em treinamento e teste conjuntos e contém uma função que cria divisão com séries temporais.
4. Tem ferramentas para dados de limpeza, bem como, e todos eles têm aplicação muito e ffi ciente.
5. Se você usá-lo para fi t um modelo, pode inicializar seus dados para melhorar o seu modelo.

desvantagens:

1. É um pacote grande e leva um pedaço de sua memória.
2. O recurso de inicialização é muito computação intensiva.

Em conclusão, acento circunflexo é um pacote de R incrível para seleção do modelo.

O que é este valor Kappa que vemos em R?
Isso é kappa de Cohen. É uma métrica usada para avaliar o desempenho de um fi classificador e também ajuda a avaliar ers fi classi entre si também. Para mais informações, dê uma olhada neste boa referência aqui.

Ao fazer Grade Pesquisa em R não podemos saber se os dados são linearmente separáveis ou não, assim como em Python?

Claro, assim como em Python você pode ver quais Kernel SVM lhe dá a melhor precisão usando grade Pesquisa com acento circunflexo. Se o Kernel é linear, então seus dados é provavelmente linearmente separáveis. E se o Kernel não é linear, os dados provavelmente não é linearmente separáveis.

10,3 XGBoost

10.3.1 XGBoost Intuition

Poderia fornecer uma boa fonte que explica como XGBoost funciona?
Recomendamos a ser introduzido primeiro a Gradiente impulsionar algorigthms passando por este grande fonte. Em seguida, recomendamos ir aqui para entender XGBoost.

10.3.2 XGBoost em Python

Tenho problemas ao instalar XGBoost. Como posso instalá-lo facilmente?
Quando o Palestra foi feito, não havia nenhum comando de instalação Conda. Mas agora há e é muito mais simples de instalar o pacote XGBoost. Você só precisa digitar o seguinte comando dentro de um terminal (ou prompt de anaconda para usuários do Windows):

```
Conda instalar -c xgboost Conda-forja
```

Pode XGBoost ser aplicado tanto Classi fi cação e Regressão em Python?
Absolutamente. Para Classi fi cação, você usa a classe fi er XGBClassi. E para Regressão, você usa a classe XGBRegressor.

10.3.3 XGBoost em R

Na seção de código Cruz Validação k-Fold, linha 35, que não deveria ser 'training_fold' em vez de 'training_set'?

De fato! Obrigado por reparar isso. É claro que é a dobra de treinamento. As nossas desculpas pelo erro. Portanto, a seção de código correta é:

```
classificador = xgboost (dados = as.matrix (training_fold [-11]),
                                        label = training_fold $ Exited, nrounds = 10)
```

Pode XGBoost ser aplicado tanto Classi fi cação e Regressão em R?
Absolutamente. E você usa a função mesmo xgboost () tanto para Classi fi cação e Regressão.