Projeto IA

Faculdade de Ciências da Universidade do Porto

Pedro Santos, up201907254

Tiago Eusébio, up201904872

Introdução

Neste projeto abordamos diferentes métodos de *machine learning* dentro de um certo conjunto de dados que neste caso foram obtidos de um estudo realizado pela Columbia University sobre eventos experimentais de speed dating, e apartir do estudo dos mesmos incorporar um algoritmo que voltando a encarar dados da mesma natureza seja capaz de adivinhar/responder com a maior precisão possível.

Desenvolvimento

Com o objetivo de criar um algoritmo como o referido a cima utilizamos a linguagem de programação python por conter um maior conjunto de bibliotecas de funções como por exemplo, para fazer a leitura dos dados recorremos a uma biblioteca de funções chamada "pandas" que permitia carregar e tratar os dados iniciais e deste modo consigir analisar separadamente os diferentes atributos dados para este conjunto de dados obtivemos os seguintes atributos e sua respetiva classificação:

Atributos	Tipo	Atributos	Tipo
Id	Categórico	Partner	Categórico
Age	Numérico Discreto	Age_o	Numérico Discreto
Goal	Categórico Nominal	Date	Categórico Ordinal
Go_out	Categórico Ordinal	Int_corr	Numérico Continuo
Lenght	Categórico Ordinal	Met	Categórico Nominal
Like	Categórico Ordinal	Prob	Categórico Nominal

Id - Número de identificação do participante

Partner - Número de identificação do par

Age - Idade do participante

Age_o - idade do par

Goal - Define o objetivo principal do participante no encontro

Date - Define à frequência de encontros do participante

Go_out - Refere-se à frequência de saidas de casa do participante

Int_corr - Correlação entre os ratings de interesses comuns

Lenght - Opinião sobre a duração do encontro

Met - Conhecimento em antemão do par

Like - Classificação do encontro pelo participante

Prob - Classificação do encontro pelo par do participante

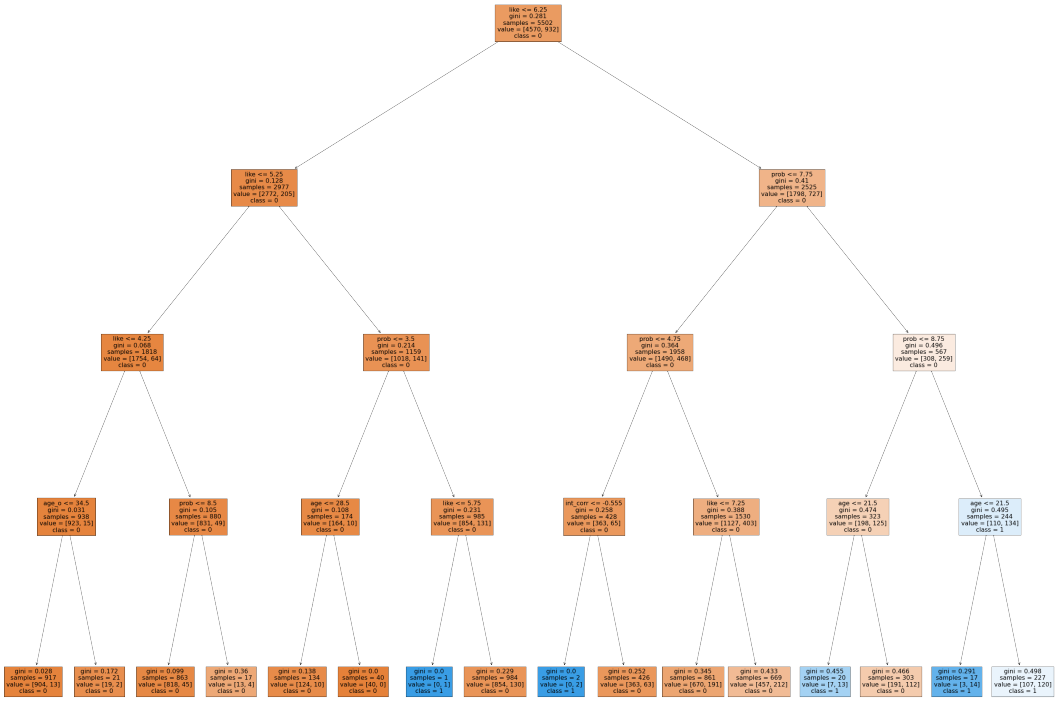
Carregados os dados com os respetivos atributos,fazemos uma filtragem dos mesmos removendo todas as linhas que contenham pelo menos um atributo não defenido com recurso a uma função "dropna", depois de eliminadas todas as linhas como mencionado acima passamos os restantes dados para a função "train_test_split" que permite fazer a divisão dos mesmos separando-os em dados de treino nos quais lhe serão dados o conjunto de parâmetros e a resposta que seja devolvida com esse conjunto para o algoritmo ser capaz perceber o quanto é que a variação de cada um dos parâmetros influencia na alteração da resposta final podendo em alguns casos ser inferido um grau de importância diferente para cada um desses parâmetros, e em dados de teste nos quais lhe serão fornecidos os parâmetros sem a própria resposta para de seguida testarmos a

capacidade de acerto do algoritmo comparando as suas respostas para esses dados com as respostas dadas pelos participantes.

Realizada a analise e tratamento de dados usamos métodos como o ID3 (Decision Tree) e o Naive Bayes que de seguida apresentaremos com maior detalhe de que forma é que variam a sua actuação

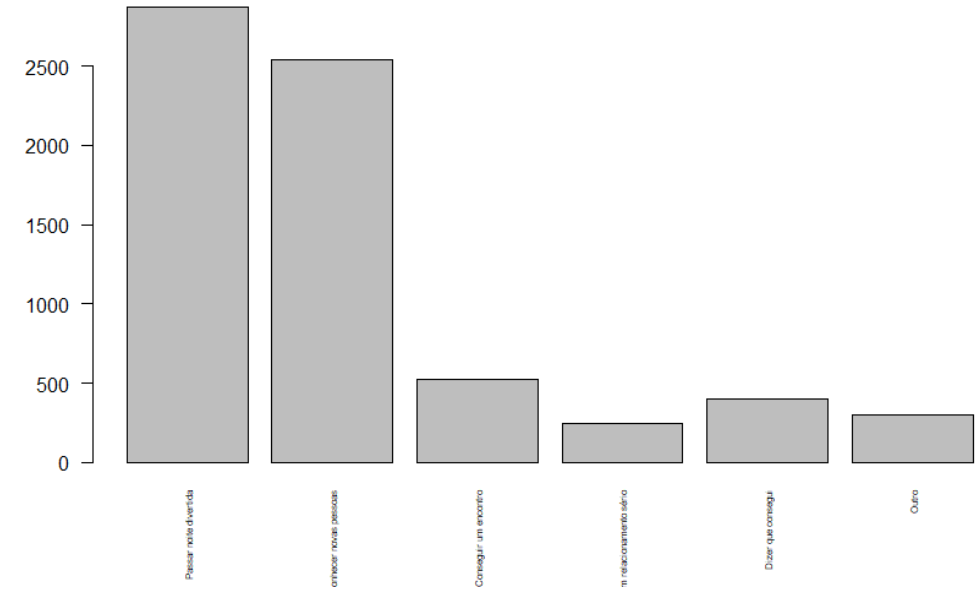
ID3

O método ID3 Iterative Dichotomiser 3 desenvolvido por J. Ross Quinlan é um algoritmo especifico para a construção de arvores de decisão tal como muitos outros que especificam uma das perguntas em cada nível podendo percorrer sub árvores completamente diferentes consoante a resposta dada . Porem este método em especifico escolhe cuidadosamente a ordem pela qual apresenta as diferentes perguntas ás quais tem acesso optando por partir sempre daquela que estará demarcada como a mais importante, mesmo de um ponto de vista humano parece óbvio para neste caso especifico de speed dating nem todas as perguntas teram o mesmo impacto na resposta final, podemos comparar perguntas como a (Goal) e a (int.corr), será óbvio que mesmo que o seu objetivo fosse apenas passar uma noite divertida se tiverem interesses muito parecidos muito provavelmente a sua escolha será um match de forma que o parâmetro int.corr tenha uma maior influencia sobre o resultado final que o goal. A razão pela qual o ID3 percorre as perguntas por essa ordem é para poder diminuir ao máximo a densidade e altura da árvore a partir da sua natureza de divide-and-conquer, uma vez que a primeira pergunta será a mais importante se a resposta for muito negativa provavelmente o match poderá ser instantaneamente marcado a 0 e marcado a 1 se for muito positiva ignorando assim alguns processamentos e obtendo uma resposta mais rapidamente, tendo isso em conta a pergunta que fica marcada como mais importante é aquela que consiga fazer a separaçao mais equilibrada de respostas para um acontecimento, ou seja tenha um valor quase igual de dados que levem a um sim ou um nao instantaneamente e outra que leve a uma revisao dos restantes parametros. A arvore de decisão obtida pelo algoritmo aparece representada a baixo:

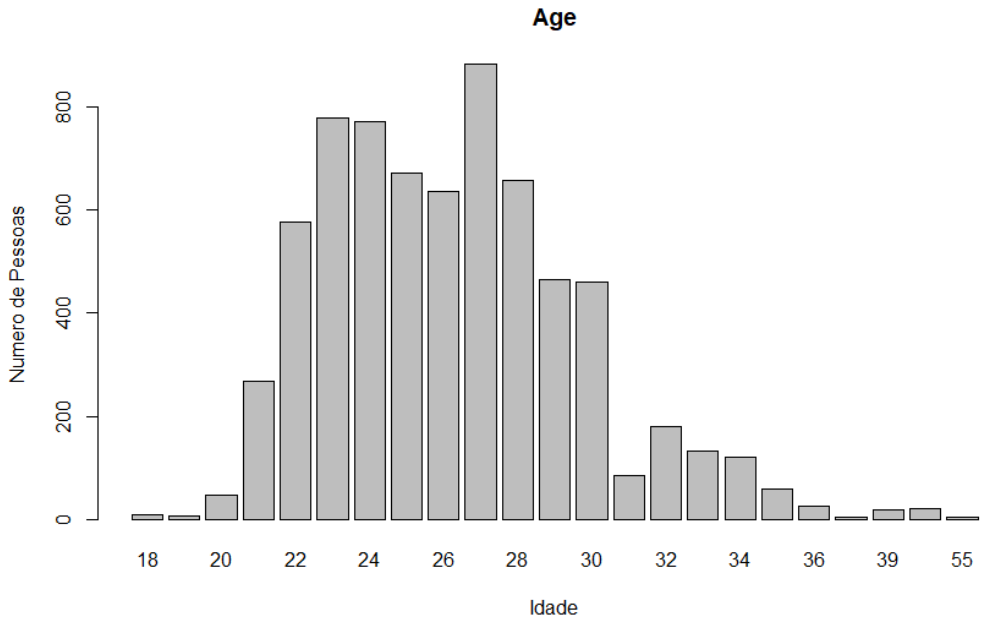


Naive Bayes

Também pusemos em execução o método de aprendizagem supervisionada Naive Bayes, neste método julgamos afincadamente todos os parâmetros individualmente, tornando assim possível calcular a probabilidade de um certo desfecho tanto com base num único parâmetro como com a junção de vários, sendo que os diferentes tipos de dados são processados de formas distintas.



Como podemos retirar da analise do grafico acima os dados categóricos são guardados em qualquer um dos seus valores de *match* (sim ou não) (e.g. se existirem 100 desfechos sim e em 15 deles o goal for igual a "outros", selecionar essa opção ira contribuir em 15% para a probablidade de a resposta final ser sim)que ocorreram quando essa opção foi selecionada ou seja a partir do grafico global demarcado em cima são criados dois sub graficos parecidos sendo que um ira mostar a dispersao de respostas para o resultado sim e o outro para a resposta nao,



enquanto que nos casos numéricos ele calcula as médias e desvio padrão de cada parâmetro quando a resposta é sim passando pelo mesmo processo quando a resposta final é não. Indicando um exemplo pratico os blocos de idades do gráfico de cima São divididos por dois gráficos para ter acesso á media de idades tanto quando a resposta é sim como não juntamente com o respectivo desvio. Desta forma o algoritmo é capaz de responder a perguntas concretas como sabendo que uma certa variável tem um valor x (e.g sabendo que o par tem idade 22 qual é a probabilidade de dar match) será capaz de responder se é mais provável a resposta ser sim ou não através da seguinte formula matemática:

$$P(X_k|Y_j) = \frac{1}{\sigma_{kj}\sqrt{2\pi}} e^{-\frac{(X_k-\mu_{kj})^2}{2\sigma_{kj}^2}}$$

Porem mesmo que sejamos capazes de chegar a pequenas conclusões com dados isolados isso não será suficiente para obter correctamente o valor da resposta final, para poder chegar a esse resultado será preciso calcular a probabilidade para cada valor apresentado que representa ambos os desfechos (sim ou não) e de seguida multiplicar todas as probabilidades ligadas na mesma reposta, calculadas essas duas probabilidades teremos de compara-las optando por aquela que tiver um maior valor.

Resultados

- Valores obtidos com 20% dos dados para treino e 80% dados para teste

Percentagem de Acertos Gauss = 82.48228239142286

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	4330	225
Positivo	739	209

Percentagem de Acertos Entropy = 76.0312556787207

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	3892	663
Positivo	656	292

- Valores obtidos com 50% dos dados para treino e 50% dados para teste

Percentagem de Acertos Gauss = 81.97150334399535

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	2666	195
Positivo	425	153

Percentagem de Acertos Entropy = 76.21401570223902

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	2432	429
Positivo	389	189

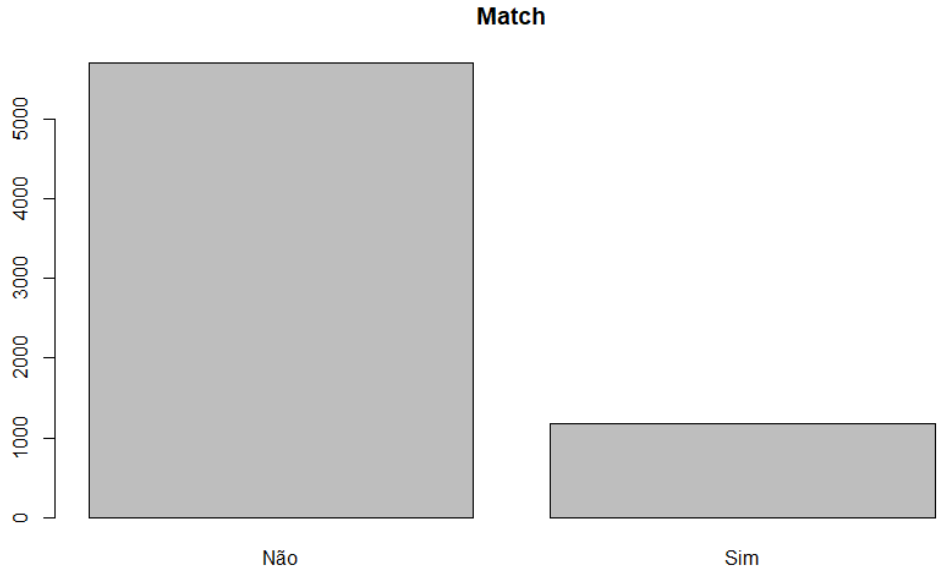
- Valores obtidos com 80% dos dados para treino e 20% dados para teste

Percentagem de Acertos Gauss = 80.95930232558139

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	1055	76
Positivo	186	59

Percentagem de Acertos Entropy = 75.36337209302324

Valores Reais	Valores Previstos	
	Negativo	Positivo
Negativo	973	158
Positivo	181	64



Análise de Resultados

Normalmente num estudo deste género o aumento de casos de estudo/treino aumentariam a taxa de acerto sobre os casos de teste, porem neste caso como é possível

observar através das probabilidades exibidas acima podemos constatar que o mesmo não acontece neste caso, devendo-se ao facto de que para este estudo os dados fornecidos estão muito pouco equilibrados sendo que cerca de 82% desses casos estão todos associados à resposta não como aliás é possível observar pela figura acima, desta forma o aumento dos casos de treino ira tornar mais propensa a escolha da resposta sim que como podemos prever o mais certo é que seja uma resposta errada diminuindo assim o seu acerto que sofrera uma variação muito grande por lhe serem feitos poucos testes.

Comparaçao

Analisando atentamente os dois métodos podemos denotar algumas principais diferenças, como por exemplo o facto de o método naive ser capaz de realizar pequenas suposições mesmo sem conter o valor de todas as variáveis enquanto o ID3 funciona única e exclusivamente se tiver acesso às respostas de todas as variáveis, por outro lado quando ambos tem acesso a toda a informação que lhes é disponibilizada o ID3 será mais rápido por ser capaz, mesmo necessitando dos dados completos para ser viável ignorar algumas das informações chegando a respostas com um menor processamento por fazer suposições como "se a resposta à pergunta x é tão unilateral então não será necessário ter em conta as restantes respostas", enquanto que naive como vê os parâmetros como iguais precisará de calcular o valor de todos uma vez que a falta de uma única variável será suficiente para que o resultado seja diferente, tendo em conta essas diferenças na analise de parâmetros juntamente com as matrizes de confusão representadas anteriormente nos "Resultados", podemos perceber que a razão pela qual o método ID3 perde face ao Naive deve-se ao facto do ID3 arriscar muito mais na resposta sim, o que nos leva a perceber que existem um numero significativo de dados no qual a variável mais importante obtida pelo algoritmo "Like" contem um valor muito propício para um resposta positiva levando o primeiro método a marcar automaticamente a sua escolha, enquanto que todos ou pelo menos a maioria dos restantes dados serão negativos que depois de serem analisados mais tentamente pelo Naive são associados à sua verdadeira resposta que seria o não.

Conclusão

Uma vez terminada a explicação de como os diferentes métodos dividem e interpretam todos os dados que lhes são fornecidos, e analisadas as suas capacidades de acerto, podemos concluir que o método ID3 pelo facto de ter uma abordagem mais greedy ao problema será a melhor escolha se o objectivo for encontrar uma aproximação que não seja muito exata especialmente se o quantidade de dados for muito elevada uma vez que é capaz de abdicar da analise de alguns dados, já por sua vez o Naive seria a opção indicada se estivermos mais focados em obter valores mais exatos, conseguindo assim obter por norma uma maior taxa de acerto nas decisão que toma com os dados que lhe são atribuídos inicialmente.

Referências

- Naive Bays:
  - https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
- Arvores de Decisão:
  - https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html