Lift Management - Parte 2

AIAD 18/19 - Grupo 50

António Almeida (up201505836)
Diogo Torres (up201506428)
João Damas (up201504088)

Descrição do problema - Enquadramento

- 1ª parte: sistema gestão de uso de elevadores num edifício; sistema parametrizado de acordo com fatores internos e externos
- 2ª parte: utilizar o sistema desenvolvido para fazer uma previsão da taxa de uso média dos elevadores, dentro de um conjunto predefinido de classes representativas do nível de uso: baixo, médio, alto - **problema de classificação**

Variáveis independentes	
Número de pisos no edifício	Probabilidade chamada/piso (Call Strategy)
Número elevadores	Tipo chamada inicial ao elevador (Lift Strategy)
Capacidade máxima/elevador	Frequência de novos pedidos (Call Frequency)
Tempo transporte entre pisos (Lift speed)	Tempo espera entrada/saída pessoas (Stop time)

Variável dependente
Taxa de uso média/elevador, codificada de acordo com o seguinte critério: 0 <= tx < 30% -> “baixo” 30 <= tx < 45% -> “medio” tx >= 45% -> “alto”

Descrição do dataset utilizado

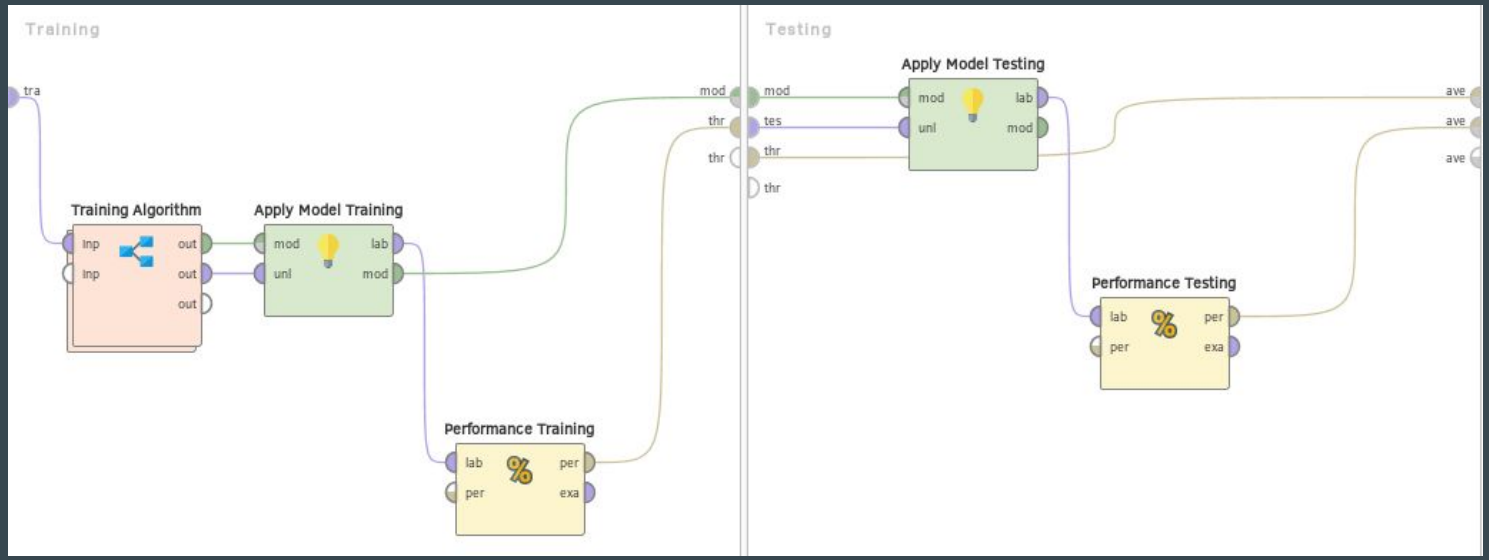
- 2750 entradas, sem conjuntos de input duplicados

Variável	Lift Capacity	#Floors	#Lifts	Lift speed	Lift Strategy	Call Strategy	Call Frequency	Lift Stop Time
Valor min	5	50	5	8	0	0	25	1
Valor max	15	200	40	20	2	1	200	5
Valor medio	9.98	125.34	18.38	12.75	1.01	0.50	113.41	1.90
Desv. pad.	3.15	43.54	7.65	2.71	0.82	0.50	50.98	0.86

- Frequência das classes: baixo - 893; medio - 1018 (*baseline most_frequent* ~37%); alto - 839

Experiências realizadas - Processo RapidMiner

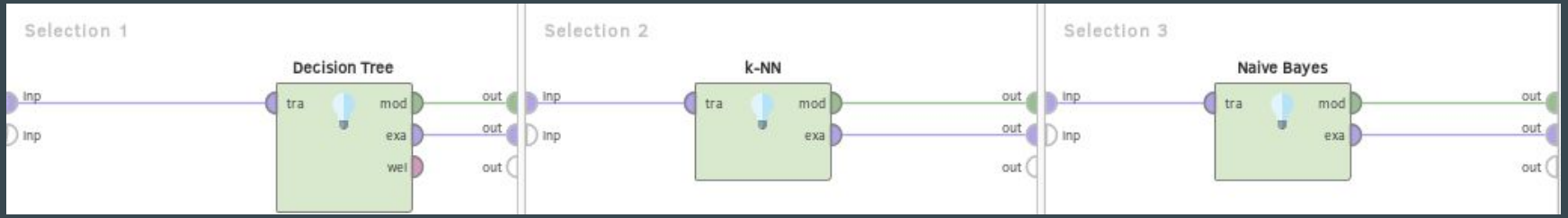
Para realizar as experiências, foi desenvolvido um processo no RapidMiner utilizando *split validation*, com um rácio 70/30.



Quer a performance de treino, quer a performance de teste são medidas.

Experiências realizadas - Training Algorithm

Training algorithm é um operador *Select Subprocess* permitindo fácil alternação entre diferentes algoritmos de treino.



Foram recolhidas estatísticas sobre a performance utilizando algoritmos de árvores de decisão (para 3 critérios de escolha de atributo para *splitting - accuracy, information_gain* e *gain_ratio -* e, para cada um desses 3, variação do *minimum leaf size*), k-NN (com variação do valor de k) e Naive Bayes.

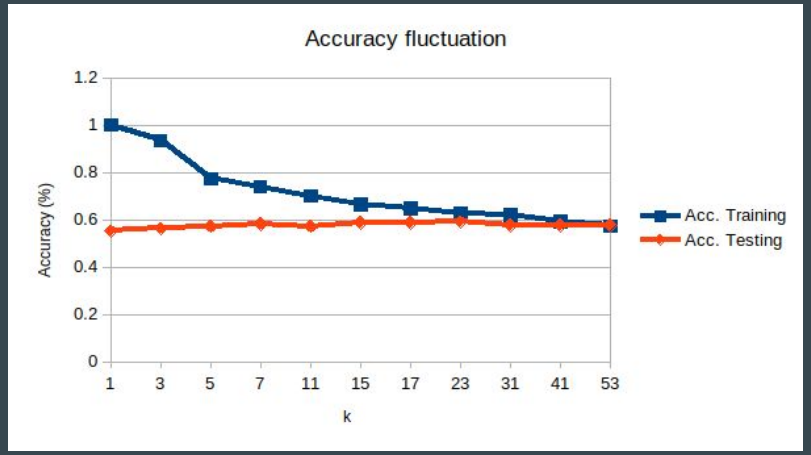
Experiências realizadas - Naive Bayes

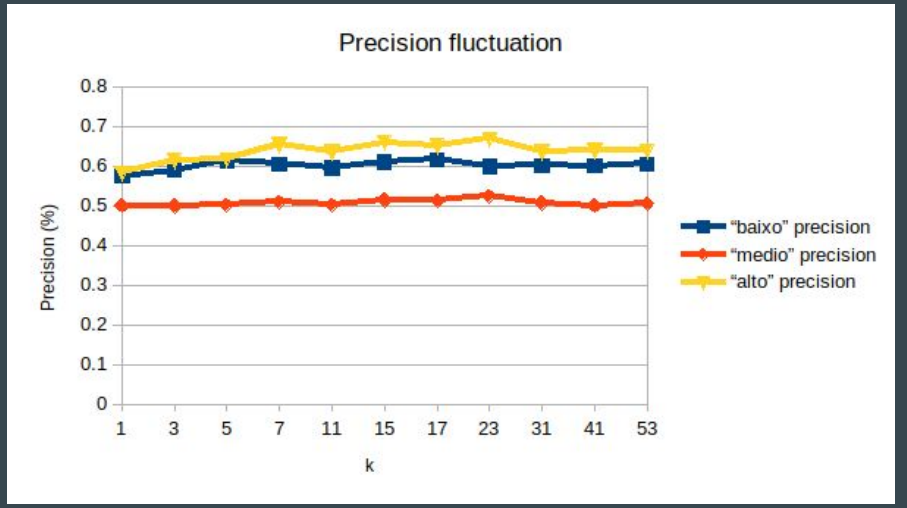
Resultados da experiência efetuada utilizando Naive Bayes como algoritmo de treino:

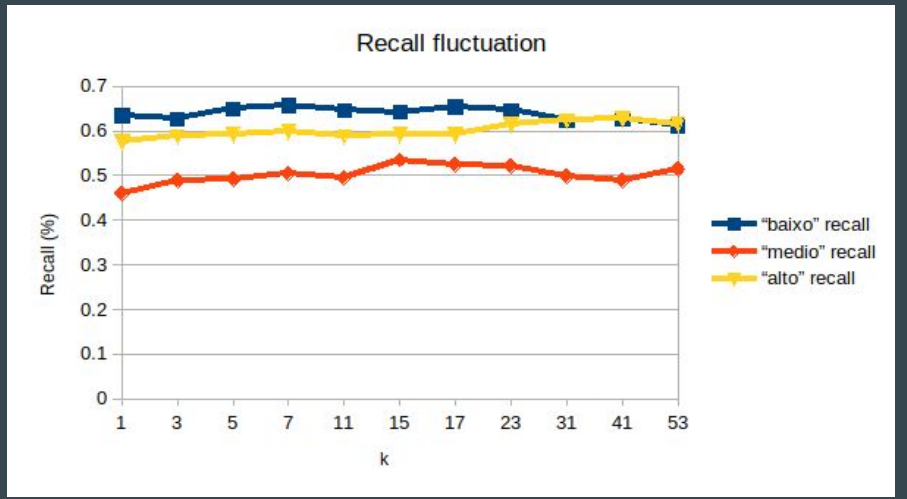
Acc. Train	Acc. Test	"Baixo" recall	"Medio" recall	"Alto" recall	"Baixo" precision	"Medio" precision	"Alto" precision
0.586	0.546	0.5858	0.4361	0.619	0.6331	0.4539	0.5493

- **Accuracy global** apresenta alguma melhoria relativamente ao *baseline*, embora algo baixa, quer em treino, quer em teste
- Taxas de **recall** e **precision** da classe "medio" não muito satisfatórias (menos de metade de previsões positivas corretas)
- **Conclusão**: Naive Bayes não será o algoritmo mais apropriado para construir um modelo preditivo para este *dataset*, talvez porque as variáveis independentes não são todas completamente independentes entre si

Experiências realizadas - k-NN

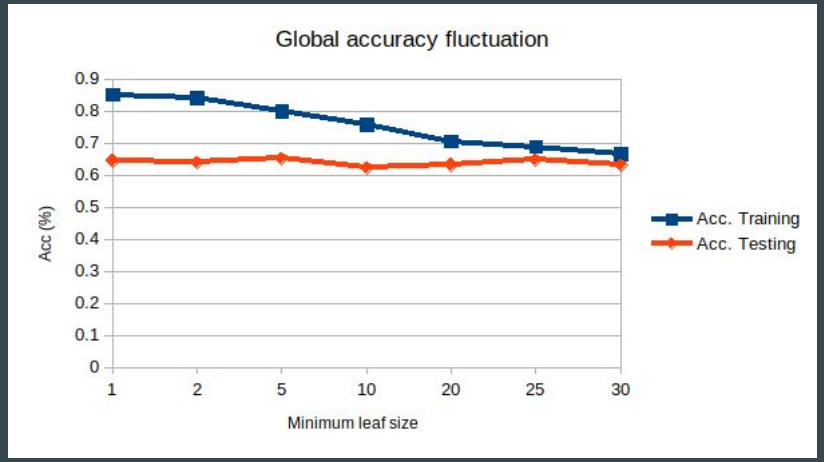


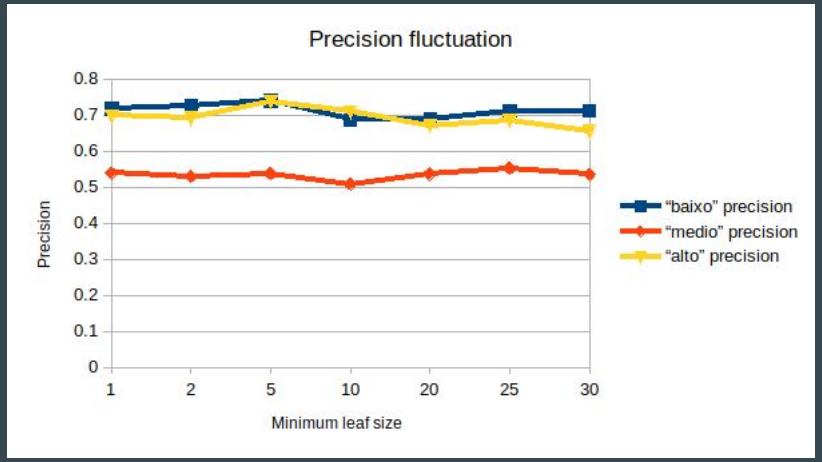




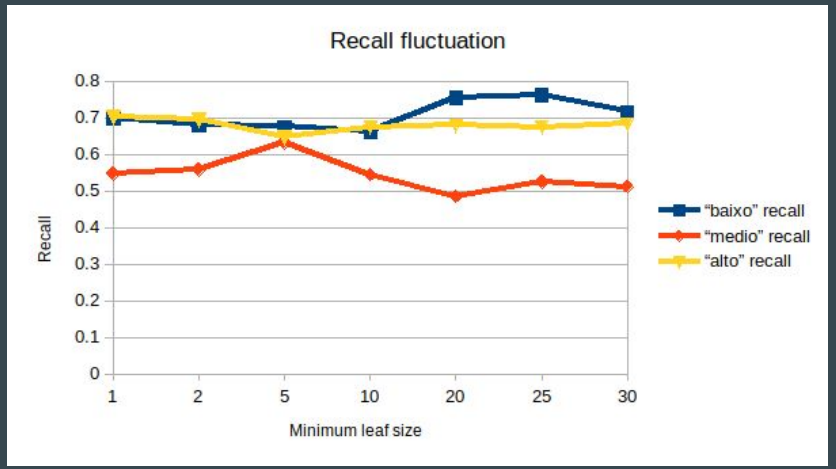
- **Accuracy global** novamente a ter valores de melhoria, mas não muito melhores do que utilizando Naive Bayes: o melhor valor é ~59% para k=23
- **Recall** e **Precision** parecem maximizar globalmente para k ~ 15/17
- Melhor, mas performance pode ter sido limitada por padrões locais

Experiências realizadas - Árvores decisão (*information_gain*)

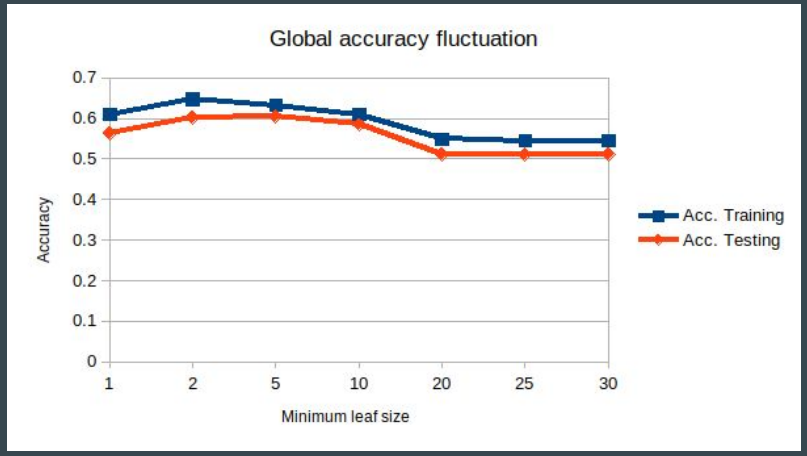


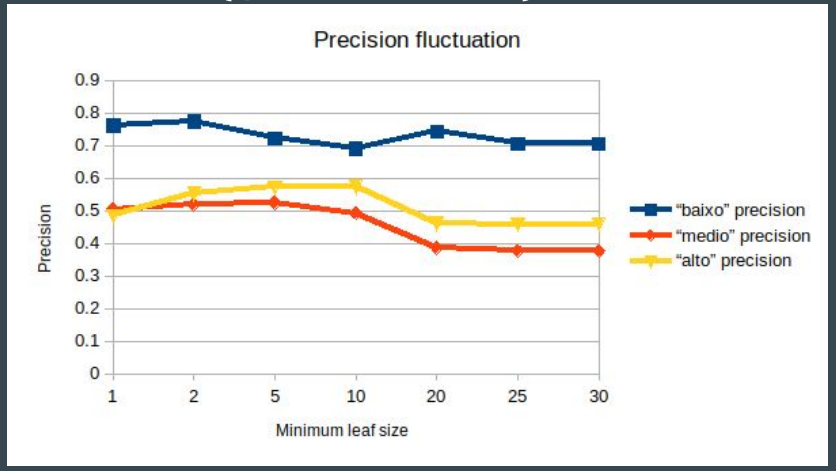


- **Information gain** - Atributo com menos entropia escolhido, com algum *bias* para atributos com mais valores
- Para min. leaf size = 5 precision global atinge valores máximos
- No mesmo contexto, recall atinge valores de >= 62% para todas as classes (ou seja, para cada, mais de 6 em cada 10 classificações positivas eram realmente positivas).

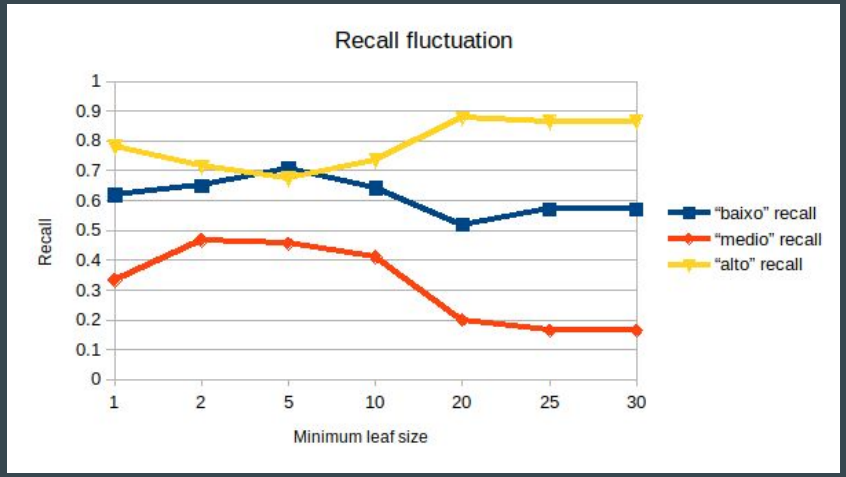


Experiências realizadas - Árvores decisão (*gain_ratio*)

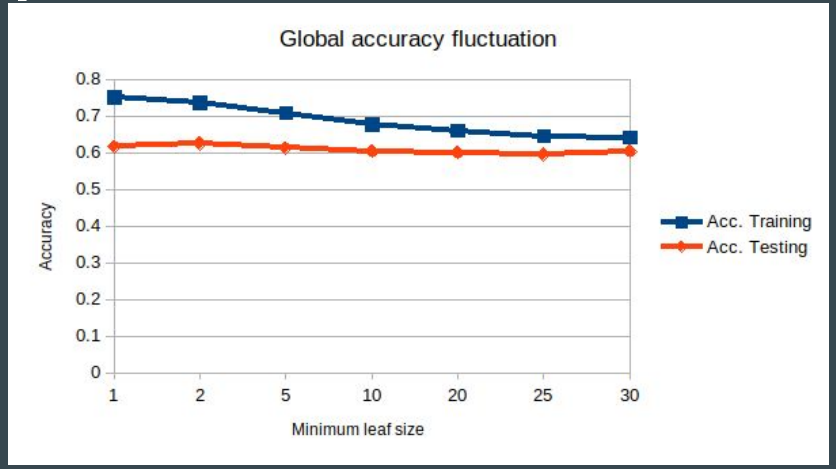


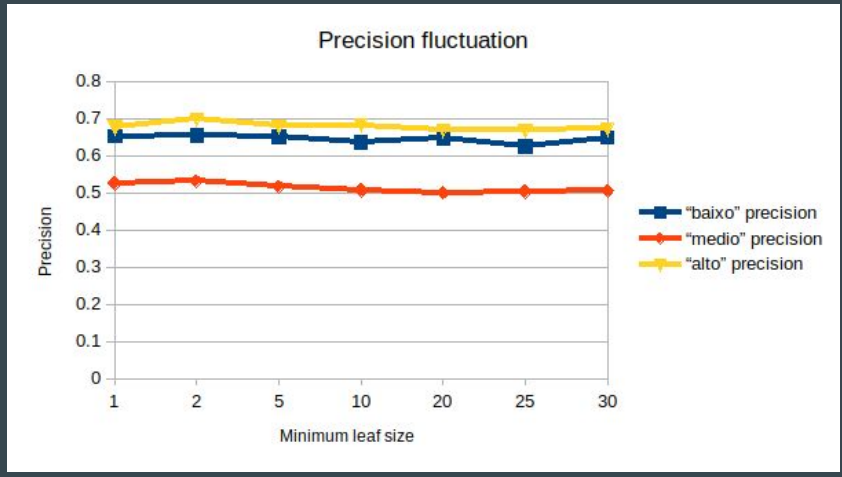


- **Gain ratio** - Semelhante a information gain, mas tenta determinar que ganho de informação será realmente relevante após o split
- Recall para a classe "medio" nunca superior a 50% com um decréscimo constante para valores de k >= 2
- Accuracy global decresceu relativamente ao critério anterior; o processamento extra deste critério não parece ser compensado nos resultados obtidos.

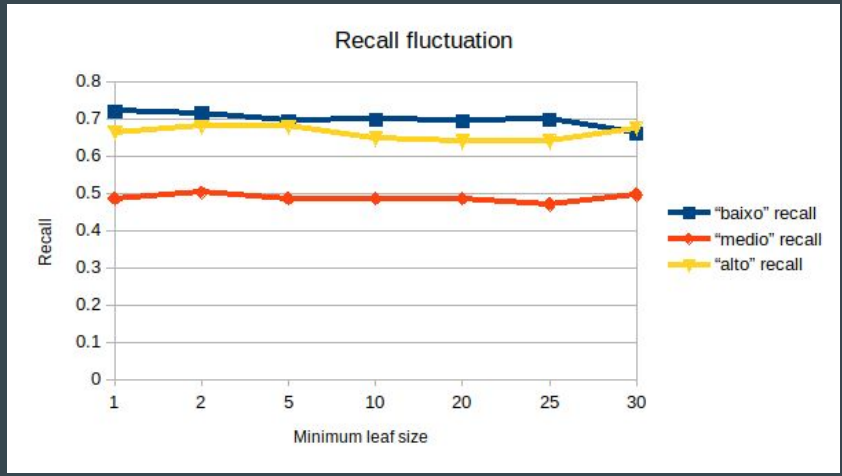


Experiências realizadas - Árvores decisão (*accuracy*)





- **Accuracy** - A escolha do atributo é feita de acordo com a maximização da accuracy global da árvore
- Resultados de certa forma semelhantes ao critério de ganho de informação, exceto recall que apresenta valores muito mais estáveis, embora inferiores



Análise de resultados e conclusões

Resultados experiências

- **Árvores de decisão** apresentam-se como o melhor algoritmo para treino do *dataset* e obtenção de um modelo que permita classificar exemplos futuros
- Nesse contexto, information gain foi o critério que apresentou melhores resultados, com uma performance global máxima aparente para min. leaf size = 5
- As matrizes de confusão para árvore decisão (information gain) apresentam-se em anexo no ficheiro **experiments.html**

Impacto variáveis independentes

- Através de uma análise visual no RapidMiner (árvore muito grande para colocar aqui), constatou-se que o atributo com maior influência é Call Frequency (root da árvore), o que era esperado porque apresentava maior diversidade de valores, algo favorecido em information gain
- Lift e Call Strategy (as mais importantes na 1ª parte) apresentam alguma influência, estando presentes em níveis intermédios da árvore
- Number of lifts parece ser a variável que permite retirar menos conhecimento, por ser quase sempre das últimas a ser escolhida para splitting

Análise de resultados e conclusões

Conclusão/Trabalho futuro

- Com este trabalho foi possível aplicar alguns métodos conhecidos de aprendizagem de forma a reconhecer padrões que um humano dificilmente conseguiria ver; no contexto do problema, a conclusão é que uma árvore de decisão com critério de ganho de informação e minimum leaf size 5 permitiria obter um modelo com resultados preditivos a um nível interessante para casos futuros.
- Algumas melhorias podem sempre ser realizadas. Neste caso, um pré-processamento mais complexo do conjunto de dados inicial poderia eventualmente ter sido explorado.

Processo Rapidminer

O ambiente de desenvolvimento no RapidMiner foi exportado no formato XML como descrito em https://community.rapidminer.com/discussion/37047/how-can-i-share-processes-without-rapidminer-server e encontra-se no arquivo enviado em conjunto com esta apresentação, no ficheiro **process.xml**. O processo de importação é semelhante ao de exportação.

Outras observações

Os dados recolhidos das experiências realizadas e utilizados para construção dos gráficos apresentados são enviados numa página HTML (resultado da exportação do excel), **experiments.html**, no arquivo enviado em conjunto com esta apresentação.