

Simulação de Poker baseada em Agentes

No Limit Texas Hold'em

Grupo 4

César Pinho - 201604039
João Barbosa - 201604156
Rui Guedes - 201603854

Parte 1



Descrição do problema de análise de dados

A personalidade *Pro* é uma das personalidades associadas aos jogadores presentes na simulação multi-agente de **NLTH** (No Limit Texas Hold'em), uma variante bastante popular de **Poker**. Esta corresponde a um agente reativo que adapta a sua jogabilidade conforme o tipo de adversários que defronta. Determinados métodos usados por este agente na sua tomada de decisão podem ser otimizados através da aplicação de modelos gerados por algoritmos de *machine learning* na área da aprendizagem supervisionada (classificação e regressão).

Neste âmbito, dos problemas possíveis a considerar, optou-se pela seleção de dois problemas que são, na nossa opinião, os mais adequados para este agente e melhor contextualizados nos objetivos do presente trabalho:

- **Problema de Regressão**

- Previsão dos valores de personalidade (*hand selection* e *aggression*) de um jogador com base nas suas jogadas anteriores, usando regressão para cada um destes dois valores.

- **Problema de Classificação**

- Previsão do resultado final de uma ronda da personalidade *Pro* (Ganho ou não de *chips*), tendo em conta a sua situação no *flop*.

Experiências realizadas

Experiência 1

➤ Objectivo

- Resolução do problema de classificação.

➤ Variáveis Independentes

- **(string) cards** - representa a combinação da mão do jogador com as 3 cartas presentes na mesa
- **(float) opponent's hand selection** - caracteriza a seletividade por parte dos seus oponentes.
- **(float) opponent's aggression** - caracteriza a agressividade por parte dos seus oponentes.

➤ Variáveis Dependentes

- **(boolean) final status** - valor que determina que se no final da ronda o jogador venceu ou não. No caso de vitória o valor desta variável é *True*, caso contrário é *False*.

➤ Algoritmos utilizados

- *Support Vector Machines, Stochastic Gradient Descent, Decision Trees, Gaussian Naïve Bayes, Nearest Neighbors, Random Forest and Neural Networks.*

Experiências realizadas

Experiência 2

➤ **Objectivo**

- Resolução do problema de classificação.
- Diminuição da dimensionalidade da experiência (Existem 33649 combinações de cartas possíveis).

➤ **Variáveis Independentes**

- ***(float) effective hand strength (ehs)*** - caracteriza a força da mão do jogador com base na combinação das suas cartas com as cartas que se encontram na mesa.
- ***(float) opponent's hand selection*** - caracteriza a seletividade por parte dos seus oponentes.
- ***(float) opponent's aggression*** - caracteriza a agressividade por parte dos seus oponentes.

➤ **Variáveis Dependentes**

- ***(boolean) final status*** - valor que determina se no final da ronda o jogador venceu ou não. No caso de vitória o valor desta variável é *True*, caso contrário é *False*.

➤ **Algoritmos utilizados**

- *Support Vector Machines, Stochastic Gradient Descent, Decision Trees, Gaussian Naive Bayes, K-Nearest Neighbors, Random Forest and Neural Networks.*

Experiências realizadas

Experiência 3

➤ Objectivo

- Resolução dos problemas de regressão.

➤ Variáveis Independentes

- **(string) bet actions** - últimas cinquenta jogadas efetuadas pelo oponente (*All in, Bet, Call, Check, Fold, Raise*).
- **(float) call value sum** - somatório das quantias provenientes de jogadas do tipo *Call*.
- **(float) bet value sum** - somatório das quantias provenientes de jogadas do tipo *Bet*.
- **(float) raise value sum** - somatório das quantias provenientes de jogadas do tipo *Raise*.

➤ Variáveis Dependentes

- **(float) hand selection** - selectividade de um dado oponente.
- **(float) aggression** - agressividade de um dado oponente.

➤ Algoritmos utilizados

- *Support Vector Machines, Decision Trees, K-Nearest Neighbors, Random Forest.*

Estatísticas sobre os dados recolhidos

Para os problemas de classificação e regressão, os dados recolhidos foram devidamente guardados num ficheiro CSV para facilitar a sua posterior importação e manipulação. Apesar de o conteúdo diferir nos datasets dos diferentes problemas estes apresentam uma estrutura semelhante na medida em que as suas colunas iniciais correspondem às variáveis independentes e a coluna final à variável dependente.

Dataset de regressão

- Ficheiro - personality.csv
- Cada linha do dataset segue o seguinte formato:
 - [`<bet_action_X>`],[50]`<hand_selection>`,`<aggression>`
- O tamanho do dataset é relativamente pequeno, contando com 1329 casos aleatoriamente gerados.
- Apesar da sua aleatoriedade, o dataset foi sempre gerado segundo a mesma configuração. No entanto esta configuração encontrava-se diversificada de modo a incluir o maior de número possível de jogadores com diferentes personalidades.

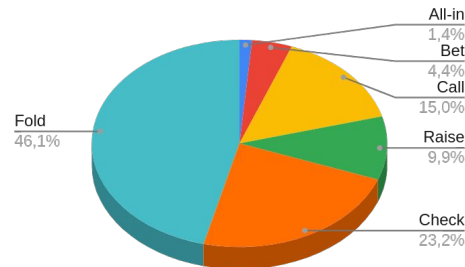


Figura 1 - Distribuição dos vários tipos de jogadas presente no dataset de regressão.

Estatísticas sobre os dados recolhidos

Dataset de classificação

- Ficheiro - round.csv
- Cada linha do dataset segue o seguinte formato:
 - `<ehs>,<hand_selection_X>,<aggression X>,<final_status>`
- O tamanho do dataset é relativamente pequeno, contando com 1171 casos aleatoriamente gerados.
- O dataset é balanceado na medida em que a percentagem do número de casos em que o jogador não sai vencedor no final de uma ronda é aproximadamente igual aos casos em que é vitorioso.
- Através da figura 3 é possível verificar que, conforme expectável, o valor médio da *feature effective hand strength* quando o jogador saiu vitorioso no final da ronda (0,46) é superior ao valor médio em situações contrárias (0,34).

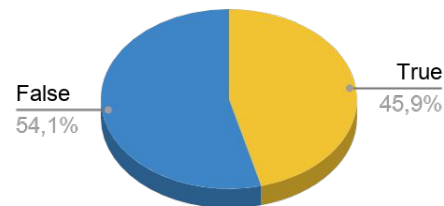


Figura 2 - Distribuição do *final_status* dataset de classificação

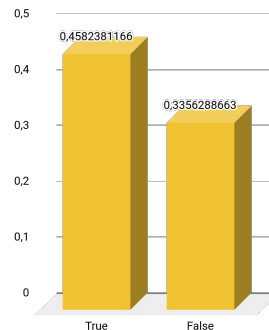


Figura 3 - Valor médio da Effective Hand Strength para cada label possível

Análise dos dados – Classificação

Na experiência 1 e 2, ambas referentes ao problema de classificação, foram utilizados diversos algoritmos de forma a resolver o problema em questão da forma mais eficiente possível. Estes algoritmos já se encontram previamente mencionados nas respectivas experiências.

De modo a analisar qualitativamente os algoritmos em questão recorreu-se à análise de diversas métricas: [classification report](#), [confusion matrix](#), [accuracy score](#), [learning curve](#), [roc curve](#) e [precision recall curve](#). Independentemente da métrica utilizada, todas elas possuem a sua respectiva importância no que diz respeito à comparação entre os diferentes algoritmos.

Na primeira experiência, todos os algoritmos apresentaram baixas performances, com a métrica de *accuracy score* a rondar os 55%. Deduzimos que o fator principal para este resultado negativo foi a utilização da combinação das 2 cartas do jogador com as 3 cartas na mesa como variável independente, tendo em conta que existem 33 649 combinações de cartas possíveis, intratáveis pelos algoritmos de treino dado o tamanho reduzido do dataset, em relação a este número de combinações.

Na segunda experiência, de forma a mitigar o problema acima referido, substituiu-se a variável independente *cards* pelo *effective hand strength*, uma métrica usada pelo simulador de Poker para estimar a força de uma mão em relação à mesa. Esta alteração gerou resultados positivos, destacando-se as redes neurais ([MLPClassifier](#)) e as máquinas de suporte vetorial ([SVC](#)) como os algoritmos que obtiveram melhores resultados, com uma *accuracy score* a rondar os 70%. Os restantes algoritmos sofreram também melhorias, no entanto mínimas, rondando os 55% a 60% de *accuracy score*.

Análise dos dados – Regressão

Na experiência 3, referente ao problema de regressão, foram utilizados diversos algoritmos de forma a resolver o problema em questão da forma mais eficiente possível. Assim, e como os algoritmos de classificação, estes algoritmos já se encontram previamente mencionados na respectiva experiência.

De modo a analisar qualitativamente os algoritmos em questão, recorreu-se a diversas métricas: [explained variance score](#), [max error](#), [mean absolute error](#), [mean squared error](#), [median absolute error](#), [r2 score](#), [mean poisson deviance](#), [mean gamma deviance](#) and [learning curve](#). Independentemente da métrica utilizada, todas elas possuem a sua respectiva importância no que diz respeito à comparação entre os diferentes algoritmos.

Nesta terceira experiência, os algoritmos que apresentaram melhores resultados foram as *decision trees* ([DecisionTree Regressor](#)) e o *random forest* ([RandomForest Regressor](#)), com destaque para este último que apresentou valores de *explained variance score* de 90%. Já as *decision trees*, estas apresentaram o valor de 84% aproximadamente. Os restantes algoritmos utilizados neste problema apresentaram uma performance pobre, com o valor de *explained variance score* a rondar os 50%.

Dadas as capacidades do scikit-learn, procedeu-se ao cálculo da [feature importance](#), a partir dos modelos referidos acima, que melhores resultados obtiveram. Com a exceção da *feature* de *fold* os resultados obtidos demonstram que as *features* mais importantes não estão relacionadas com a sua abundância no dataset. Na verdade a ação de *Check*, a segunda mais abundante, é a sexta *feature* mais importante evidenciando assim a ausência da relação direta entre abundância e importância.

Conclusões

Este segundo projecto da unidade curricular de Agentes e Inteligência Artificial Distribuída tinha como objectivo analisar os dados provenientes do primeiro projecto através de um problema de classificação e de um problema de regressão.

No início do desenvolvimento deste projecto, o grupo especificou devidamente os problemas a abordar, definindo para tal o modo como os datasets referentes a estes problemas iriam ser formatados e qual o seu conteúdo. Durante a execução do planeamento inicial, deduziu-se a possibilidade de obtenção de melhores resultados na classificação alterando as variáveis independentes deste problema, dedução verificada pelas experiências realizadas. No caso da classificação, verificou-se pela análise da *learning curve* que se poderiam obter melhores resultados com um dataset maior. Relativamente à aplicação dos algoritmos, foi possível identificar algoritmos cuja performance apresenta valores satisfatórios perante os problemas associados a estes. Em suma, e considerando todos estes factores, o grupo considera que o resultado final é positivo e promissor, e que o objetivo do projeto foi alcançado com sucesso.

Como trabalho futuro sugere-se não só o aumento do dataset utilizado para o problema de classificação, tendo sempre em atenção possíveis cenários de over-fitting, bem como uma otimização do treino dos modelos, através da eliminação de features baseado na *feature importance*; um *tuning* dos parâmetros dos algoritmos (recorrendo, por exemplo, ao [GridSearchCV](#) do scikit-learn) de modo a gerar modelos mais bem adequados aos problemas; e a integração dos melhores modelos desenvolvidos no simulador de Poker, de modo a colocar em prática a resolução destes problemas e avaliar o seu desempenho fora do ambiente de desenvolvimento dos modelos.

Parte 2

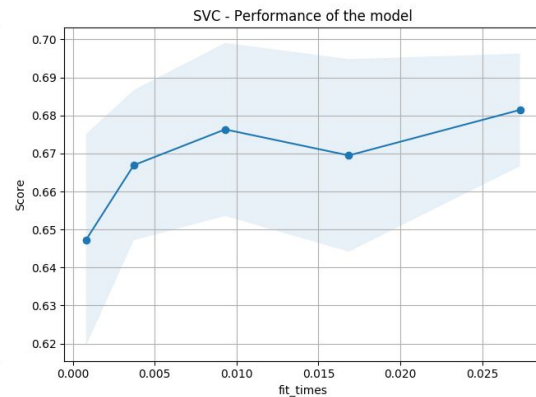
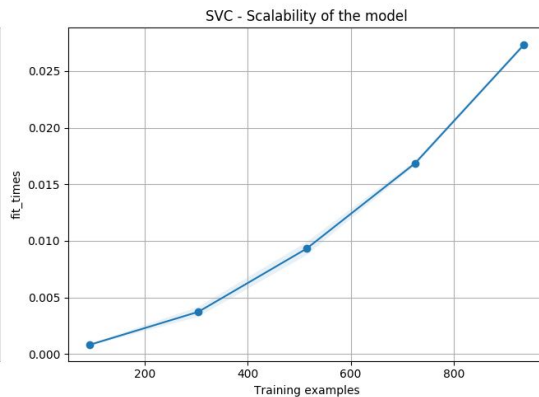
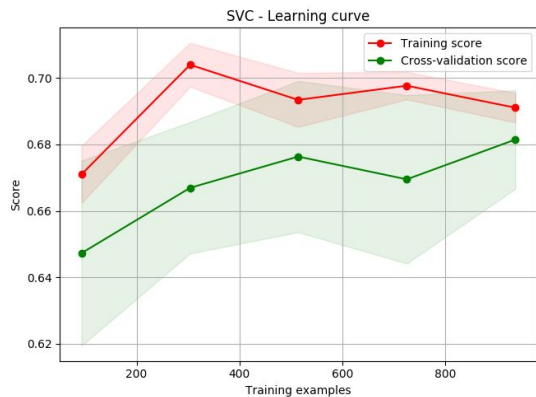


Métricas de Classificação – SVC

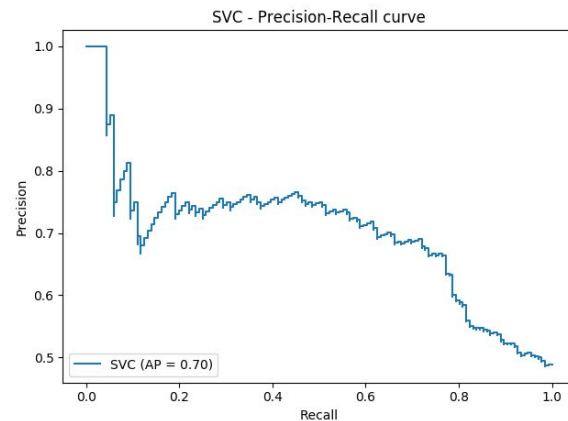
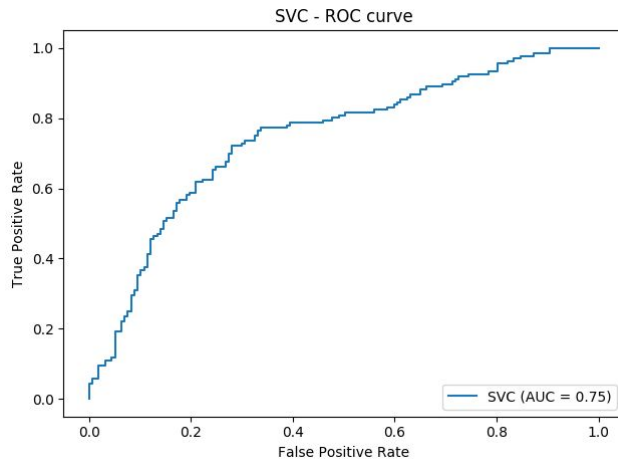
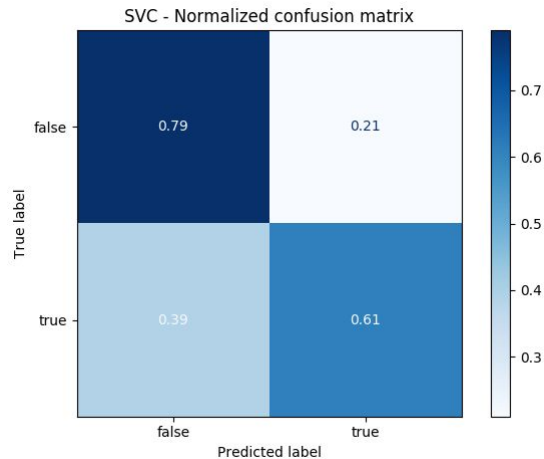
Classification report:

	precision	recall	f1-score	support
false	0.70	0.79	0.74	157
true	0.72	0.61	0.66	136
accuracy			0.71	293
macro avg	0.71	0.70	0.70	293
weighted avg	0.71	0.71	0.70	293

Accuracy score: 0.7064846416382252



Métricas de Classificação – SVC

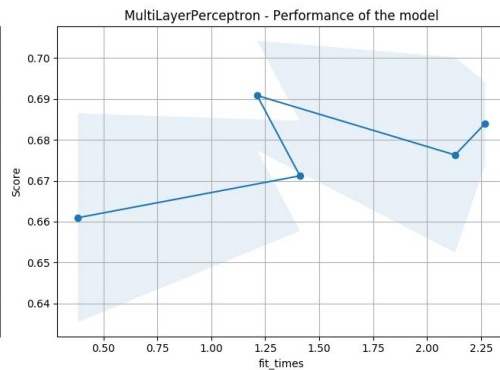
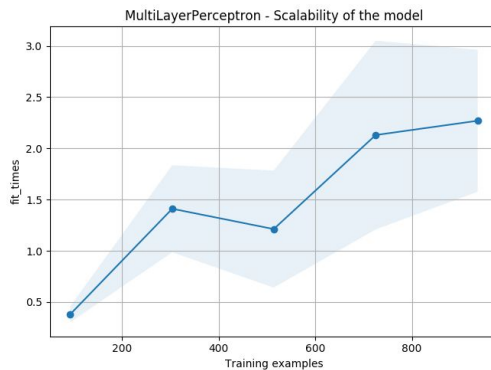
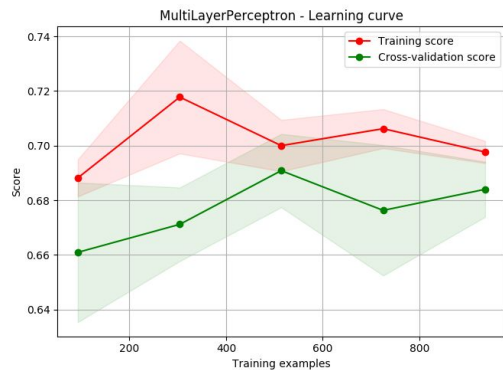


Métricas de Classificação – MLP

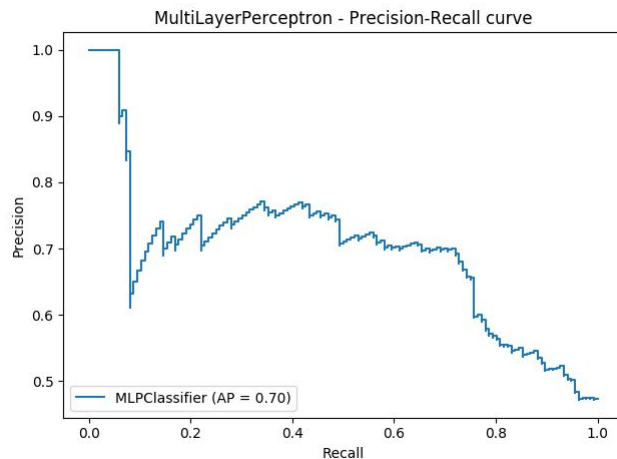
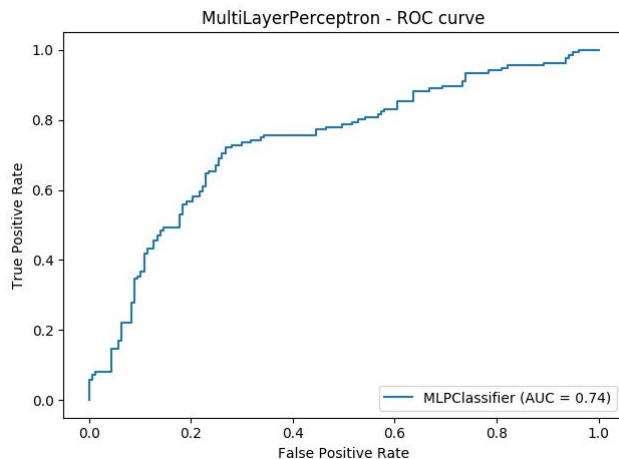
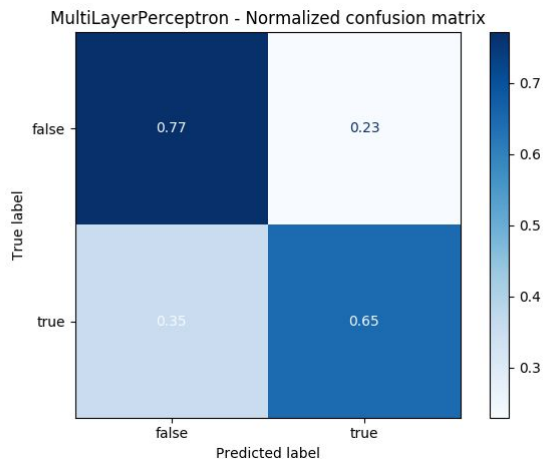
Classification report:

	precision	recall	f1-score	support
false	0.72	0.77	0.74	157
true	0.71	0.65	0.68	136
accuracy			0.71	293
macro avg	0.71	0.71	0.71	293
weighted avg	0.71	0.71	0.71	293

Accuracy score: 0.7133105802047781



Métricas de Classificação – MLP



Métricas de Regressão – Random Forest

General Metrics

Explained variance score: 0.9078473255059962

Maximum residual error: 0.3315

Mean absolute error regression loss: 0.034834028667362404

Mean squared error regression loss: 0.005381873763150715

Median absolute error regression loss: 0.00349999999999990594

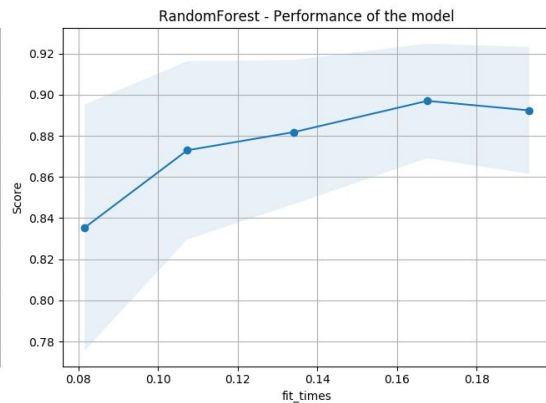
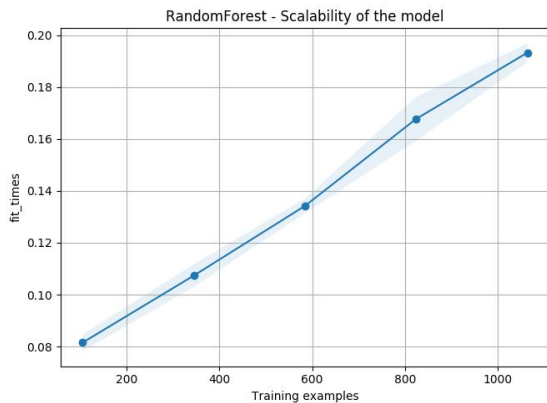
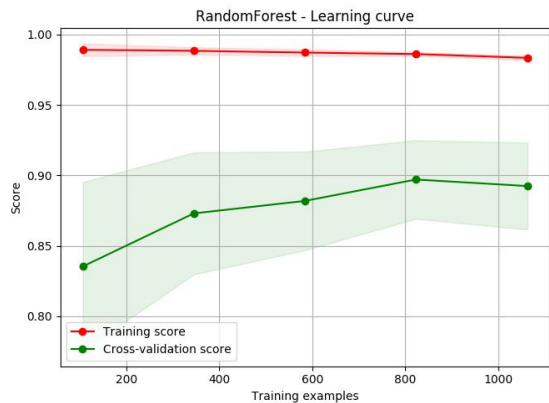
R² (coefficient of determination) regression score function: 0.9076313523255397

Mean Poisson deviance regression loss: 0.009532282236987061

Mean Gamma deviance regression loss: 0.017744174816177048

Feature importance

	Importance
Fold	0.780826
Call	0.087715
Raise	0.054074
Call_sum	0.023364
Raise_sum	0.016699
Check	0.013592
Bet_sum	0.009968
All_in	0.007327
Bet	0.006436



Outras observações

GridSearchCV

- No presente projecto, e como funcionalidade adicional, o grupo optou por incorporar as funcionalidades do *GridSearchCV* que o *scikit-learn* fornece com objectivo de facilitar a tomada de decisão no que diz respeito à melhor configuração a ser aplicada aos diferentes algoritmos utilizados.

Decision Trees

- Relativamente à implementação do algoritmo de *decision trees* encontram-se associadas duas funcionalidades extra:
 - *feature importance* - Determina a importância de cada uma das features utilizadas no modelo. Também disponível para o algoritmo de random forest.
 - *export* - Exporta a *decision tree* num ficheiro de forma a facilitar a sua visualização.

Estrutura do código

- O código encontra-se dividido essencialmente em dois packages, um referente ao problema de classificação e outro ao problema de regressão, nos quais as respetivas classes correspondem aos modelos utilizados e às métricas a estes associadas.
- Todo o código encontra-se devidamente documentado para facilitar a sua interpretação.
- Juntamente com o código encontra-se um ficheiro README.md para auxiliar no setup e execução do projecto.