1. Introdução e Objetivos

Este projeto visa desenvolver o melhor modelo de classificação para determinar se um objeto é biodegradável. A base de dados utilizada contém variáveis categóricas e contínuas não normalizadas, com valores em falta, que serão tratados adequadamente, e colunas que podem ser descartáveis.

Pretendemos também explorar diferentes Hiper parâmetros para o nosso melhor modelo, numa tentativa de otimizar a sua performance. No entanto, é importante lembrar que, mantendo todas as outras condições semelhantes, os **modelos mais simples devem ser preferidos**.

Outro objetivo importante é identificar as características mais significativas que influenciam a biodegradabilidade de um objeto, de forma a melhorar a nossa compreensão acerca dos fatores mais relevantes.

O objetivo final é a criação de um modelo de classificação preciso, oferecendo uma compreensão mais aprofundada dos fatores que influenciam a biodegradabilidade de um objeto.

2. Processamento de Dados

2.1 Variáveis Categóricas em Contínuas

No início do nosso processo, deparámo-nos com a necessidade de **converter variáveis categóricas em contínuas**. Este passo é crucial pois muitos dos algoritmos usados no nosso caso, especialmente os baseados em cálculos matemáticos, requerem *inputs* numéricos e não conseguem processar diretamente dados categóricos.

Foi usada a técnica, *one-hot*, onde **cada categoria é convertida numa coluna própria** com um novo conjunto de dados binários (como no caso da coluna *Biodegradable*). Esta conversão permite-nos incorporar informações importantes contidas nas variáveis categóricas no nosso modelo de classificação, aumentando a sua precisão e eficácia.

Na mesma seção do Notebook é possível observar um gráfico que nos permite identificar as variáveis categóricas **nominais** e **ordinais**, sendo assim possível fazer a distinção corretamente.

2.2 Separar Conjunto de Treino e Teste

Durante a modelação, os dados são divididos em dois conjuntos: **treino** e **teste**. O conjunto de treino é utilizado para construir e ajustar o modelo, enquanto o conjunto de teste é usado para avaliar o seu desempenho. Esta divisão assegura que o modelo se generaliza adequadamente para novos dados, evitando o *overfitting*.

Deste modo, utilizamos sempre os dados de treino para ajustar o modelo e os de teste para prever e avaliar a eficácia final.

2.3 Cross-validation

Outro passo empregue nesta fase foi a **Validação Cruzada**. Esta técnica é utilizada para avaliar a capacidade de um modelo em prever dados que **não** foram utilizados no treino, ajudando a detetar situações de p.ex. *overfitting*.

A validação cruzada aplicada permitiu verificar se a *performance* do modelo era melhorada em cada etapa de manipulação dos dados. Com esta abordagem, foi possível avaliar o impacto de cada passo e decidir, de forma informada, sobre como prosseguir com o processamento de dados.

Desta forma, conseguimos assegurar que **cada etapa do processamento de dados contribuiu para a melhoria da *performance* do nosso modelo**, permitindo-nos identificar e corrigir quaisquer etapas que pudessem estar a prejudicar essa mesma *performance*.

2.4 Valores em Falta *(NaN)*

No nosso conjunto de dados identificámos que algumas variáveis tinham valores em falta *(NaN)*. Optámos por uma abordagem em **2** fases para lidar com estes dados considerados incompletos.

Inicialmente, decidimos remover as variáveis que tinham mais de **14,5%** dos seus valores em falta. Este valor foi escolhido com base numa avaliação do equilíbrio entre a manutenção da quantidade máxima de informação útil e a minimização do impacto potencialmente distorcido dos valores em falta. Se uma variável tem uma grande proporção de valores em falta, a informação que ela fornece pode ser menos confiável e potencialmente induzir em erro o nosso modelo.

Após a remoção destas variáveis, aplicámos um **método de imputação** para lidar com os valores em falta restantes. Esta fase está detalhada na seção **2.5**.

2.5 Imputer

Agora que temos uma variável sem as colunas com mais de 14,5% dos seus valores em falta *(NaN)* e outra que tem essas colunas, podemos verificar qual das **2** obteve melhor desempenho.

Para preencher esses valores em falta foi necessário usar métodos de **imputação**, pelo que tivemos de verificar qual o melhor método a usar, neste caso, entre o ***Simple*** e ***KNN Imputer***.

Decidimos usar **diferentes estratégias dependendo do tipo de dados**, *mean* e *most_frequent*, escolhidas com base nas características dos dados em falta – a substituir pela **média** dos valores presentes na coluna, **ou** pelo valor mais frequente **– moda**, respetivamente.

As estratégias de imputação foram **avaliadas usando o método de validação cruzada** - verificando as estatísticas (*precision, recall, F1 score e Matthews correlation coefficient*).

Foram **comparados os desempenhos** para os modelos *Decision Tree* e *Logistic Regression*, de forma, a ter uma rápida avaliação da decisão tomada. Tal foi efetuado para cada fase de processamento de dados do nosso problema.

Para cada um dos modelos, verificámos as estatísticas usando o *dataset* incluindo as colunas com valores ausentes - mencionado anteriormente - e sem estas colunas, de modo a verificar se perdemos informação ao retirar as mesmas.

Os valores das estatísticas referentes a cada método de Imputação estão apresentados na secção 2.5 do Notebook e organizados nas tabelas 1 e 2:

		Precision	Recall	F1 score	MCC
DecisionTree	Sem Retirar NaN	0.9515	0.9806	0.9658	0.7658
	Retirando NaN	0.9546	0.9860	0.9701	0.7949
LogisticRegression	Sem Retirar NaN	0.9553	0.9853	0.9700	0.7952
	Retirando NaN	0.9553	0.9853	0.9700	0.7952

Tabela 1 - Desempenho do método Simple Imputer

		Precision	Recall	F1 score	MCC
DecisionTree	Sem Retirar NaN	0.9451	0.9884	0.9663	0.7644
	Retirando NaN	0.9467	0.9849	0.9654	0.7596
LogisticRegression	Sem Retirar NaN	0.9564	0.9872	0.9716	0.8058
	Retirando NaN	0.9564	0.9868	0.9714	0.8045

Tabela 2 - Desempenho do método KNN Imputer

Foi possível detetar que a estratégia de imputação *KNN Imputer* foi a melhor, dado que gera valores de estatística ligeiramente melhores relativamente ao *Simple Imputer*.

Concluímos que a melhor técnica de imputação foi o *KNN* dado que, comparando os resultados das estatísticas de *KNN Imputer* - com e sem os valores ausentes *(NaN)* -, verificamos que os resultados diferem muito pouco, pelo que as colunas que apresentam mais de 14,5% dos seus valores ausentes não interferem com os resultados.

Sendo assim, e **usando a estratégia de imputação *KNN***, decidimos retirar essas colunas e preencher os valores ausentes das restantes colunas usando este método, o que nos permite ter um dataset simplificado.

2.6 Scaler

Depois de todos os valores estarem preenchidos, foi necessário verificar se existem diferenças que pudessem prejudicar o desempenho dos modelos.

Foi feito o *Scaling* dos dados e analisamos o desempenho do conjunto de treino usando dois métodos diferentes: o *Standard* ***Scaler*** e o ***MinMax*** *Scaler*.

Dados categóricos, binários e ordinais não foram escalados pois o seu resultado não tem nenhum impacto significativo nesse tipo de dados.

É possível verificar os resultados obtidos na secção 2.6 do Notebook. Ao observarmos esses resultados, conseguimos perceber que a métrica MCC difere bastante entre *Decision Tree* e *Logistic Regression*, bem como entre os dois métodos de *Scaling*.

Já os valores das restantes métricas são relativamente semelhantes entre si. No entanto, as que no geral apresentam **melhores resultados utilizam o *Logistic Regression* com *Standard Scaling***, pelo que **decidimos utilizar este método**.

3. Seleção de Variáveis

Nesta secção, **focamo-nos na seleção das variáveis mais relevantes para o modelo** de biodegradabilidade, através de métodos como *stepwise* e *correlation*. Avaliamos o tratamento de valores em falta e planeamos a normalização dos dados, garantindo a preparação adequada para a modelagem.

Esta etapa é crucial para otimizar a *performance* do modelo e reduzir a complexidade dos dados.

3.1 Pela Variância

Outro critério que adotamos na seleção de variáveis foi a **análise de variância**. As variáveis com baixa variância podem não adicionar informação significativa para a tarefa de classificação, uma vez que apresentam valores muito semelhantes para todas as observações - isto é, se os valores numa coluna são quase os mesmos, essa coluna pode não ser muito útil para o nosso modelo.

Deste modo, consideramos eliminar as colunas com variância muito baixa. Estabelecemos um limiar para a variância (0.01) e eliminamos as colunas cuja variância estava abaixo desse limiar.

Ao fim desta análise, concluímos que esta abordagem é eficaz e que vale **a pena retirar variáveis com baixa variância**, **simplificando** o nosso modelo **sem comprometer** significativamente a sua **precisão**.

3.2 Stepwise & Correlation

Para além da análise da variância, para realizar a seleção de variáveis e avaliar o seu impacto no desempenho dos modelos, aplicámos os métodos ***Stepwise*** e ***Correlation***.

Através dos resultados apresentados na secção 3.2 do Notebook, podemos observar diferenças no desempenho destes dois modelos.

Tanto no caso da *Decision Tree* e no caso da *Logistic Regression*, a correlação apresentou resultados ligeiramente superiores em termos de métricas do que o método de *Stepwise*.

No último método, a *Logistic Regression* apresentou melhores resultados, pelo que decidimos simplificar o modelo usando o método de *correlação* com Logistic Regression.

4. Resultados dos Modelos

O objetivo desta etapa é comparar o desempenho de vários modelos de classificação. Para tal, foram escolhidos - **DecisionTreeClassifier**, **LogisticRegression**, **SVC**, **KNeighborsClassifier** e **RandomForestClassifier**) - utilizando parâmetros *default* para um conjunto de dados.

Primeiro, os modelos são treinados utilizando o conjunto de treino. Em seguida, o desempenho de cada modelo é avaliado utilizando o conjunto de testes, calculando várias métricas de desempenho, como *accuracy*, *precision*, *recall* e *F1-score*.

4.1 Resultados

Para obtermos os melhores modelos calculamos a **média das métricas** para cada um deles. Assim, os dois modelos com as maiores médias serão os selecionados. Na seção 4.6 do código é possível visualizar um **gráfico de barras** que mostra, para cada modelo, a respetiva **média das métricas**.

Com base no gráfico - e lembrando que a seleção de métricas depende do problema em questão, e estando neste caso a prever se algo é **biodegradável** ou não -, tanto as métricas *precision* como a *recall* são importantes.

Queremos identificar corretamente tantos objetos biodegradáveis quanto possível (alto *recall*), enquanto tentamos garantir que, quando prevemos que um objeto é biodegradável, realmente o é (alta *precision*). Portanto, a pontuação *F1*, que equilibra esses dois aspetos, é uma métrica crucial para este problema.

Podemos, desta maneira, concluir que os dois melhores modelos são o **RandomForestClassifier** e o **SVC**.

5. Tuning de Hiper parâmetros

Escolhidos os dois melhores modelos, procedemos ao *tuning* dos seus parâmetros, com o objetivo de obter o melhor modelo possível. Para tal servimo-nos da classe *GridSearchCV*, de forma a testar rapidamente diversas combinações de parâmetros.

No *SVC*, testamos sete valores diferentes de *gamma*, compreendidos entre *1e-1* e *1e-7*, juntamente com 6 valores de *C*, no intervalo de *1* a *1e5*. Concluímos que o melhor par era constituído por *C=10* e *gamma=0.1*.

Relativamente ao *Random Forest*, foi testado com *50*, *100* e *150* estimadores, *5*, *10* e *15* níveis de profundidade e com dois critérios: *gini* e *entropy*. Através do teste concluímos que a melhor combinação de parâmetros envolvia o critério *gini*, com profundidade *15* e *150* estimadores.

Por fim, tendo avaliado ambos através dos critérios usados anteriormente para avaliar os modelos anteriores, concluímos que o modelo ótimo é o ***SVC***, com *C=10* e *gamma=0.1*, tendo obtido melhores pontuações em todas as métricas.

Estes resultados são visíveis na secção 5 do *Notebook*.

6. Discussão e Conclusões

Os resultados obtidos sugerem que o **modelo é capaz de prever corretamente se um objeto é Biodegradável ou não com alta taxa de acerto (96.68%),** e que quando o modelo prevê que um objeto é biodegradável, é verdadeiro na maioria das vezes (precisão=96.64%), identificando corretamente a maioria dos objetos – confirmado pelo elevado valor de *recall*.

O MCC indica também uma forte e significativa **correlação entre as previsões do modelo e as reais**.

Em resumo, este projeto alcançou o seu objetivo principal de desenvolver um modelo de classificação para prever a biodegradabilidade de objetos.

O modelo otimizado demonstrou um **elevado desempenho em termos de várias métricas de avaliação**.

No entanto, o **trabalho futuro** pode explorar **outras técnicas** de processamento de dados e seleção de variáveis, bem como outros modelos de classificação e técnicas de ajuste de Hiper parâmetros, para continuar a melhorar a precisão da previsão da biodegradabilidade.

Cada elemento do grupo dedicou cerca de 3 horas ao projeto.