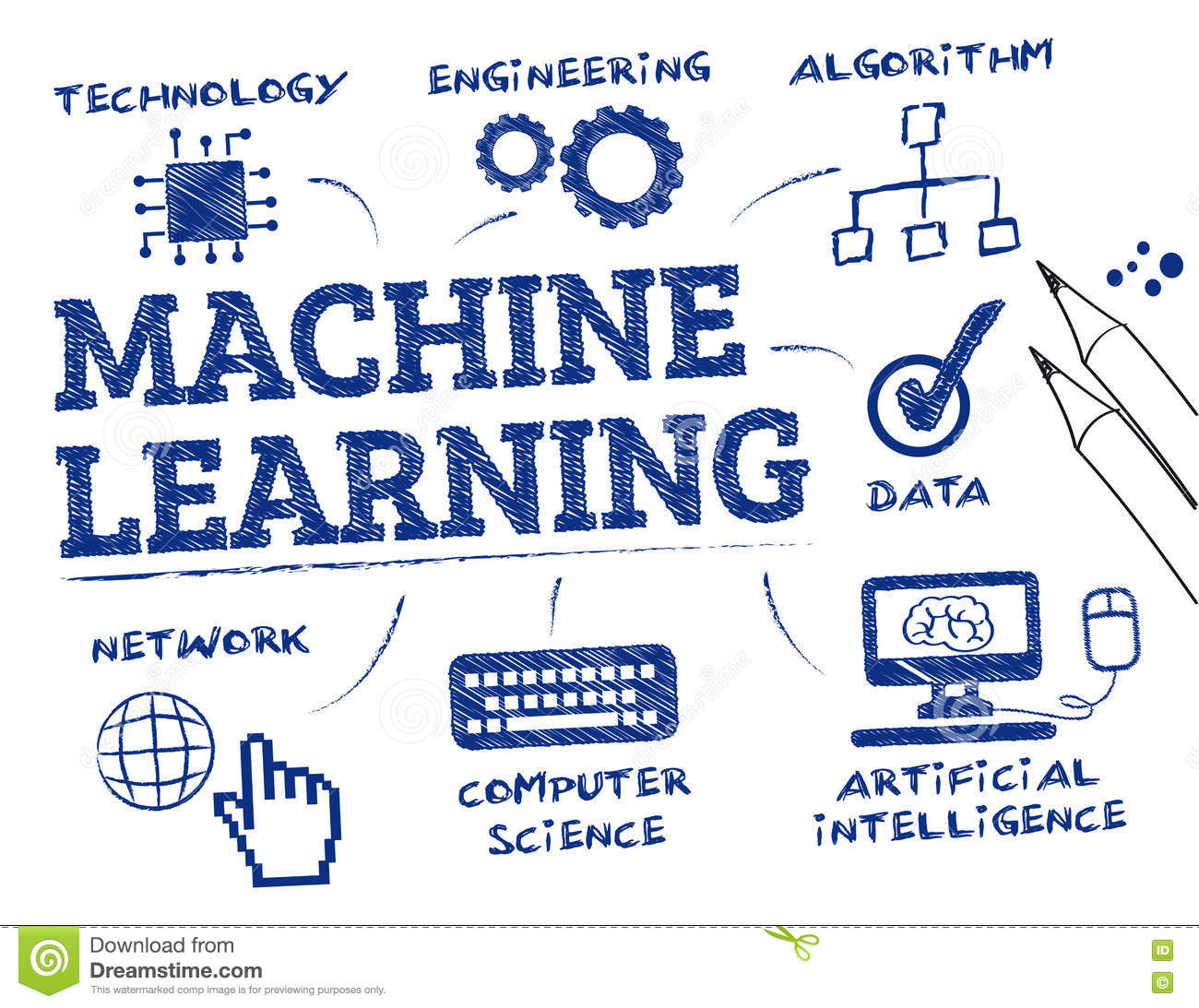
Engenharia do Conhecimento

**Projeto 2**



Grupo: 41

Francisco Henriques – 56348

Guilherme Marques – 55472

Miguel Seabra – 56344

Docentes: Sofia Teixeira, Cátia Pesquita

Data: 17/05/2023

* **Registo de horas de trabalho:**

|  |  |
| --- | --- |
| Aluno | Horas de Trabalho |
| Francisco Henriques  Guilherme Marques  Miguel Seabra | 22 Horas  5 Horas  0 Horas |

* **Introdução e Objetivos:**

O objetivo deste projeto baseia-se em desenvolver o melhor modelo de classificação possível para prever a variável *Biodegradable,* utilizando uma versão aumentada e editada do conjunto de dados: *QSAR biodegradation Data Set.* Disponibilizado pelos docentes da UC. A tarefa consistiu em 3 passos gerais como: Processar os dados disponibilizados; Examinar diferentes Hiper parâmetros para cada modelo; Selecionar as features mais importantes; Selecionar o melhor modelo;

Para atingir este objetivo, foi utilizado um JupyterNotebook de modo a ser mais fácil a execução do código necessário e da organização do output do mesmo. Esse ficheiro está em anexo junto deste relatório.

* **Processamento de dados:**

Tendo em conta que o conjunto de dados era relativamente grande (cerca de 42 colunas) e que não estava normalizado, existindo células em branco nos dados, começámos por carregar o csv disponibilizado para um panda, sendo depois utilizado o método KNNInputer para **preencher as células vazias**, este método tem em conta um K numero de vizinhos, em que neste caso foi escolhido K=5, de modo a não termos overfitting nem underfitting, com estes 5 vizinhos mais próximos o KNNInputer atribui um valor médio desses vizinhos à célula vazia.

Tendo o conjunto de dados completo e tendo em conta que a variável alvo (Biodegradable) era uma variável binária (RB ou NRB), procedemos à sua conversão em número binário 1 para RB e 0 para NRB.

Depois foram feitos testes para determinar qual o scaler que melhor se adaptaria aos dados, para isso foi utlizada uma função do notebook da TP04, que imprimia estatísticas para diferentes scalers um modelo, e um certo conjunto de dados. Nos testes observou-se que para a maioria dos modelos, o StandartScaler era o que trazia melhores resultados tanto a nível de accuracy como de precision, recall, etc. Os resultados dos testes foram os seguintes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Scaler** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **MCC** |
| Logistic Regression | StandartScaler | 0.968 | 0.9634 | 0.9914 | 0.9772 | 0.8441 |
| RobustScaler | 0.9575 | 0.9611 | 0.9898 | 0.9753 | 0.8304 |
| MinMaxScaler | 0.9416 | 0.9425 | 0.9914 | 0.9663 | 0.7612 |
| Decision Trees | StandartScaler | 0.9628 | 0.9736 | 0.9827 | 0.9781 | 0.8552 |
| RobustScaler | 0.9582 | 0.9734 | 0.9772 | 0.9753 | 0.8387 |
| MinMaxScaler | 0.9575 | 0.9741 | 0.9757 | 0.9749 | 0.837 |
| Random Forests | StandartScaler | 0.9735 | 0.9805 | 0.9882 | 0.9844 | 0.897 |
| RobustScaler | 0.9735 | 0.9813 | 0.9874 | 0.9844 | 0.8972 |
| MinMaxScaler | 0.9735 | 0.982 | 0.9867 | 0.9843 | 0.8975 |
| SVM | StandartScaler | 0.9688 | 0.9716 | 0.9922 | 0.9817 | 0.877 |
| RobustScaler | 0.9608 | 0.962 | 0.9929 | 0.9772 | 0.844 |
| MinMaxScaler | 0.9502 | 0.9531 | 0.9898 | 0.9711 | 0.7992 |

Como podemos ver em 3 dos 4 modelos testados apenas no modelo RandomForests obtivemos um resultado muito idêntico entre o RobustScaler e o StandartScaler, sendo o Recall um pouco melhor no StandartScaler e a precision um pouco pior, face às pequenas diferenças nos valores consideramos que não vale a pena ser utilizado o RobustScaler.

* **Seleção de Variáveis:**

Na hora de selecionar as variáveis, o objetivo é o de remover as variáveis irrelevantes. Para isso usamos o correlation selection method. Para encontrarmos as variáveis irrelevantes temos de procurar as variáveis que têm uma correlação perto de 0 pois perto de 1 significa que apoiam fortemente a feature alvo (Biodegradable) e perto de -1 rejeitam fortemente a feature alvo, daí as que têm uma correlacao com valor perto de 0 serem irrelevantes pois nem apoiam nem rejeitam a feature alvo não trazendo assim vantagens suficientes para continuarem no dataSet. Tendo em conta a correlação das features em relação a variável alvo obtivemos o seguinte gráfico:

Uma imagem com texto, captura de ecrã, file, Paralelo

Descrição gerada automaticamente

Como podemos ver pelo gráfico existem muitas features que praticamente não são relevantes para a previsão da variável Biodegradable (a nossa variável alvo) por este motivo é que devemos reduzir o número de colunas para obter somente as que são relevantes , reduzindo a dimensionalidade, o que vai ser muito importante para a performance dos nossos modelos e até logo a seguir quando formos avaliar quais são as features mais importantes.

Ao remover as variáveis irrelevantes (cerca de 20 colunas) e usando uma RandomForest e escalando os dados com StandartScaler, obtivemos o seguinte gráfico:

Uma imagem com texto, captura de ecrã, diagrama, file

Descrição gerada automaticamente

Neste gráfico facilmente percebe-se que não existem features com performance muito perto de 0 logo apenas estão incluídas as relevantes

* **Afinação com os parâmetros:**

Iniciamos a testar a ferramenta GridSearchCV ao usar o SVC. Com esta ferramenta conseguimos "afinar" o modelo SVM com os melhores hiperparametros, neste caso são : C : 10 e gamma : 0.1

Depois de termos encontrado, com RandomizedSearchCV (utlizando uma grid de parametros possiveis), os melhores hiperparametros para afinar o modelo de RandomForests (o nosso melhor modelo até agora) conseguimos perceber uma pequena melhoria no modelo em comparação com outro RandomForest sem os hiperparametros e testado com os mesmos dados

* **Testes dos modelos:**

Depois de termos os hiperparametros, iniciamos a construção dos diferentes modelos com os devidos hiperparametros.

Os modelos foram construídos através do Logistic Regression, Decision Trees, SVM e Random Forests. Criamos uma tabela para comparar os resultados dos modelos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Logistic Regression** | **Decision Trees** | **SVM** | **Random Forests** |
| Accuracy | 0.943597 | 0.945587 | 0.971466 | 0.974784 |
| Precision | 0.948679 | 0.961300 | 0.976025 | 0.977589 |
| Recall | 0.986656 | 0.974882 | 0.990581 | 0.992936 |
| F1 Score | 0.967295 | 0.968044 | 0.983249 | 0.985202 |
| MCC | 0.770878 | 0.785944 | 0.888194 | 0.901284 |

Como podemos observar pela tabela, tanto o modelo SVM como o RandomForest conseguiram melhores resultados em todas as métricas. Comparando então os dois modelos:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SVM** | **RF** | **RF – SVM** |
| Accuracy | 0.971466 | 0.974784 | 0.003318 |
| Precision | 0.976025 | 0.977589 | 0.001564 |
| Recall | 0.990581 | 0.992936 | 0.002355 |
| F1 Score | 0.983249 | 0.985202 | 0.001954 |
| MCC | 0.888194 | 0.901284 | 0.013090 |

Como podemos ver o modelo RandomForest é o melhor em todas as métricas sendo que no F1 Score temos uma pequena diferença mas no MCC acabamos por ter um décimo de percentagem a mais.

Ficou assim escolhido **o modelo RandomForests com os hiperparametros : 'n\_estimators': 300, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1, 'max\_features': 'sqrt', 'max\_depth': None, 'bootstrap': False**

* **Discussão e conclusões sobre os resultados obtidos:**

Em falta…