


| | | |
|---|--|---|
|  | UNIVERSIDADE FEDERAL DA PARAÍBA | |
| | CENTRO DE INFORMÁTICA | |
| | Disciplina | Aprendizagem de Máquina |
| | Semestre | 2021.2 |
| | Professores | Bruno Jefferson de Sousa Pessoa Gilberto Farias de Sousa Filho |

PROJETO DA DISCIPLINA

1. Introdução

O projeto final da disciplina tem o objetivo de aplicar os conceitos de aprendizagem de máquina (AM) abordados durante o curso a problemas práticos enfrentados pela sociedade. A ideia é que o aluno utilize um *dataset* oriundo de algum problema no qual está trabalhando ou então um *dataset* disponível na Internet que possua uma aplicabilidade prática relevante. O discente então implementará os modelos de Aprendizagem de Máquina SVM, Redes Neurais e Árvores de Decisão, sobre o *dataset* escolhido, a fim de realizar classificações/predições em um conjunto de dados de testes. Além disso, um relatório deve ser elaborado contendo informações sobre a implementação dos modelos e uma comparação dos resultados obtidos para cada um deles.

2. Datasets

O discente está livre para escolher *datasets* das mais variadas fontes, desde que os dados se refiram a problemas práticos enfrentados pela sociedade e não sejam simplesmente gerados para fins de implementação de modelos de AM. Desse modo, os alunos são encorajados a buscar informações em sites do governo como o IBGE, ministério da saúde, órgãos que concentram informações sobre a educação pública (INEP), segurança pública, aspectos econômicos e sociais do país, entre outros. *Datasets* que contenham informações sobre outros países também podem ser utilizados. A ideia é que o aluno realize algum tipo de análise exploratória e preparação dos dados antes da implementação dos modelos de AM, o que renderá um bônus extra em sua avaliação. O aluno também pode optar por *datasets* disponíveis em sites como o Kaggle e o UCI, cujos dados já estão estruturados. Nesse caso, não havendo esforço de engenharia de dados, o aluno não fará jus ao bônus mencionado.

3. Instruções gerais

- O presente projeto deve ser desenvolvido de forma individual.
- Cada aluno deve trabalhar com *dataset* único na turma. Logo, escolha seu *dataset* e informe nesta planilha onde todos os alunos terão acesso:

Link: https://docs.google.com/spreadsheets/d/1xBGz7mKscsofJ0guJ_6in-hkN8GgnhW8CCAeaguTeOA/edit?usp=sharing

- O projeto deverá ser enviado ao professor (bruno@ci.ufpb.br) até o dia **21/06/2022** e apresentado no dia **22/06/2022** em algum dos horários dos seguintes horários:

| Horários para defesa do projeto | |
|---------------------------------|---------|
| 22/06/2022 | |
| 14:00 h | 16:00 h |
| 14:30h | 16:30h |
| 15:00 h | 17:00 h |
| 15:30 h | 17:30 h |

Escolha seu horário de apresentação e agende na planilha indicada acima

- Deverá ser enviado um arquivo zipado, contendo o código fonte e o relatório, no formato descrito a seguir:
 - AM-Projeto-Autor.zip
 - Ex.: AM-Projeto-Jose_Oliveira.zip
- Os seguintes itens serão avaliados: *dataset*, pré-processamento de dados, eficácia do modelo, código-fonte, tratamento da generalização do aprendizado, regularização/validação, aplicabilidade prática do projeto e apresentação.

4. Instruções para implementação e relatório

- O projeto deve ser desenvolvido em Python.
- As principais bibliotecas a serem utilizadas são Pandas, Numpy, Scikit-Learn, Matplotlib, Keras e TensorFlow.
- O projeto pode ser desenvolvido em um Jupyter Notebook ou em uma IDE de livre escolha.
- O aluno deverá elaborar um relatório descrevendo em palavras (não basta apenas colar o código) como realizou a implementação de cada atividade descrita a seguir, apresentando os resultados quando solicitados.

4.1. Tratamento de dados básico

Com relação à etapa de tratamento de dados, as seguintes etapas devem ser realizadas e deverão constar no relatório:

- Tratar os dados e construir a matriz X e o vetor y de entrada para o classificador;
- Identificar o número de amostras (N) e número de parâmetros (p);
- Separar os dados (X, y) entre treinamento e teste, justificando a quantidade de exemplos de cada grupo de dados. Não mexer no teste até a fase final de computação das métricas de aprendizado;
- Aplicar normalização ou padronização dos dados, justificando a escolha do método.

4.2. Implementação do modelo baseado em Rede Neural

No que diz respeito ao modelo baseado em Rede Neural, as seguintes etapas devem ser realizadas e abordadas no relatório:

- Definir a arquitetura da rede neural (número de camadas e o número de neurônios por camada) a partir do número de exemplos disponíveis no *dataset* e da “Regra de Ouro” que é consequência da Teoria da Generalização do Aprendizado. Use o cálculo da dimensão VC para justificar a escolha da arquitetura mencionada. **Obs:** Não esqueça da existência do bias em cada neurônio e use o Teorema da Aproximação Universal;
- Computar o E_{in} e E_{out} para analisar a existência de *overfitting*;
- Ocorreu *overfitting*? Se sim, isso ocorreu a partir de que época? Ilustrar sua resposta com um gráfico;
- Justificar as escolhas para os parâmetros batch size e o número de épocas;
- Computar as métricas de qualidade da melhor rede construída (acurácia, precisão, recall, f1 score).

4.3. Construção do modelo de Árvore de Decisão

As seguintes etapas devem ser realizadas e abordadas no relatório:

- Construir uma árvore de decisão com a instância de treino;
- Plotar a árvore e computar E_{in} e E_{out} para analisar a existência de *overfitting*;
- Regularizar o valor de α utilizando o algoritmo de *Minimal Cost-Complexity* já implementado na classe *DecisionTreeClassifier*, para encontrar a árvore que minimize a relação:

$$Pureza(T) + \alpha \cdot \#folhas(T).$$

Esse processo deve ser realizado com a técnica de *cross validation*, onde o tamanho do *fold* deve ser definido pela dimensão do conjunto de treino;

- Plotar a imagem e computar as métricas de qualidade da melhor árvore construída.

4.4. Construção do modelo SVM

Com relação à elaboração do modelo SVM, as seguintes atividades devem ser realizadas e tratadas no relatório:

- Construir um modelo de SVM, regularizando dos parâmetros C e γ (gama) através de *cross validation*, onde o tamanho do *fold* deve ser definido pela dimensão do conjunto de treino;
- Computar o valor de E_{in} , E_{out} e o valor de E_{out} esperado, baseado no número de vetores de suporte utilizados na solução, para analisar a existência de *overfitting*;
- Computar as métricas de qualidade de classificação do melhor modelo encontrado.

4.5. Escolha do melhor modelo

Escolher o melhor modelo, treinado nas fases anteriores, para o dataset utilizando a instância teste através de uma estratégia de validação para escolha de modelos.