

Sistema de Previsão de Qualidade do Sono dos Alunos

Um modelo preditivo para avaliar a qualidade do sono utilizando redes neurais.



TABELA DE CONTEÚDO



01

introdução

02

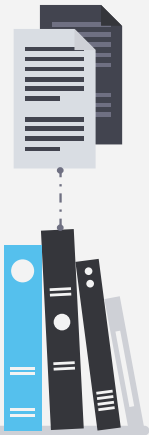
**Problema e
Solução**

03

**Bibliotecas
utilizadas**

04

**Estrutura do
código**



Introdução

O objetivo deste projeto é desenvolver um modelo preditivo que utilize redes neurais para prever a qualidade do sono dos usuários com base em seus padrões de comportamento e estilo de vida.

O modelo é treinado com dados históricos de padrões de sono para identificar relações entre variáveis como tempo de estudo, atividade física, ingestão de cafeína e duração do sono

Uma previsão precisa da qualidade do sono pode fornecer aos usuários informações valiosas, ajudando a melhorar seus hábitos e tomar decisões informadas para promover um sono mais saudável e reparador.





Problemática

A qualidade do sono está diretamente relacionada ao bem-estar e ao desempenho diário, mas muitos estudantes sofrem com sono inadequado devido a hábitos irregulares, uso de dispositivos eletrônicos e outros fatores.

Publico-alvo

Estudantes universitários que enfrentam problemas relacionados à qualidade do sono, influenciados por rotinas acadêmicas e estilo de vida.

01



Características

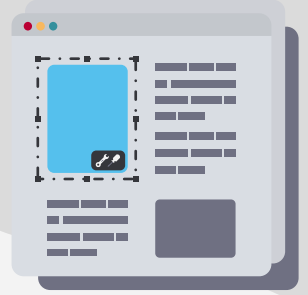
Uso de dados de padrões de sono, ingestão de cafeína, atividade física e tempo de tela para analisar e prever a qualidade do sono.

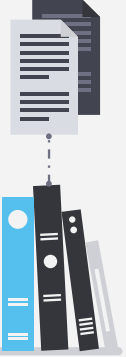
Objetivo

Desenvolver um modelo de rede neural MLP para prever a qualidade do sono com base em hábitos e comportamentos de estudantes, auxiliando-os a melhorar sua saúde do sono.



02 Problema e Solução





Problema

Estudantes sofrem com a falta de qualidade do sono, resultando em problemas de saúde e rendimento acadêmico.

Solução

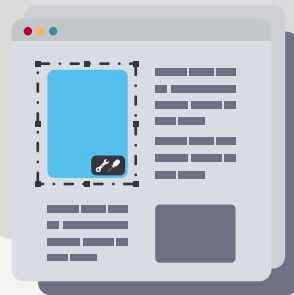
Um sistema de previsão baseado em IA (MLP) que utiliza variáveis como duração do sono, ingestão de cafeína e tempo de tela para prever a qualidade do sono, permitindo intervenções mais personalizada





03

Bibliotecas





Pandas

- Manipulação e análise de dados tabulares.



Scikit-learn

- Implementação da rede neural MLP, divisão de dados em treino e teste, e escalonamento das features



Joblib

- Salvamento do modelo e do scaler para reutilização nas previsões.





Flask

- Framework web para construir a API que permite interações externas com o modelo de IA.

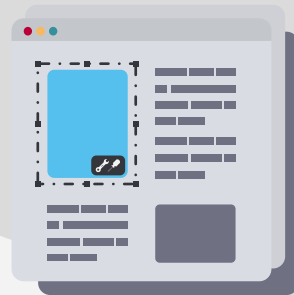
Flask-CORS:

- Liberação do acesso cross-origin para que diferentes frontends possam acessar a API.



04

Estrutura do Código





MODEL.PY



O arquivo **model.py** é onde criamos o modelo de **Machine Learning (ML)** para prever a qualidade do sono.

Ele faz três coisas principais:

- **Prepara os Dados:** Carrega os dados dos alunos e os organiza.
- **Treina o Modelo:** Ensina o modelo a entender os dados para fazer previsões.
- **Salva o Modelo:** Guarda o modelo treinado para uso futuro.



A Classe SleepQualityModel



```
class SleepQualityModel:
    def __init__(self, data_path):
        self.data_path = data_path
        self.model = None
        self.scaler = None

    def load_data(self):
        # Carregando os dados
        data = pd.read_csv(self.data_path)

        # Mapeando gênero para valores numéricos
        gender_mapping = {'Male': 0, 'Female': 1, 'Other': 2}
        data['Gender'] = data['Gender'].map(gender_mapping)

        # Selecionando as features e rótulo
        X = data[['Age', 'Study_Hours', 'Screen_Time', 'Caffeine_Intake', 'Physical_Activity', 'Sleep_Duration', 'Gender']]
        y = data['Sleep_Quality']

        return X, y
```



Função de Treinamento em **model.py**



```
def train(self):
    X, y = self.load_data()

    # Dividindo os dados
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Pré-processamento
    self.scaler = StandardScaler()
    X_train_scaled = self.scaler.fit_transform(X_train)
    X_test_scaled = self.scaler.transform(X_test)

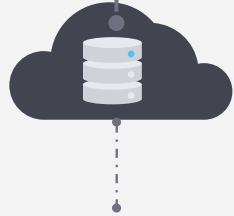
    # Ajuste fino do modelo MLP
    self.model = MLPClassifier(hidden_layer_sizes=(30, 30), activation='relu', max_iter=1500, random_state=42)
    self.model.fit(X_train_scaled, y_train)

    # Avaliando o modelo
    y_pred = self.model.predict(X_test_scaled)
    print(classification_report(y_test, y_pred))
```

You, há 48 minutos



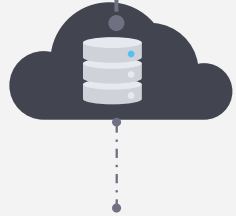
Funções de Salvamento, Carregamento e Previsão do Modelo



```
def save_model(self):  
    # Salvando o modelo e o scaler  
    joblib.dump(self.model, 'mlp_model.pkl')  
    joblib.dump(self.scaler, 'scaler.pkl')  
  
def load_model(self):  
    # Carrega o modelo e o scaler  
    self.model = joblib.load('mlp_model.pkl')  
    self.scaler = joblib.load('scaler.pkl')  
  
def predict(self, symptoms):  
    symptoms_scaled = self.scaler.transform([symptoms])  
    prediction = self.model.predict(symptoms_scaled)  
    return prediction[0]
```



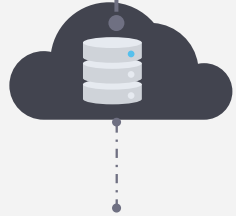
Treinamento e Salvamento **TRAIN_MODEL.PY**



```
1 from models.model import SleepQualityModel
2
3 # Crie caminho para os dados
4 model = SleepQualityModel(data_path='data/student_sleep_patterns.csv')
5
6 # Treine o modelo
7 model.train()
8
9 # Salve o modelo treinado e o scaler
10 model.save_model()
11
```




Configuração da API APP.PY



```
from flask import Flask, request, jsonify
from flask_cors import CORS
from models.model import SleepQualityModel

app = Flask(__name__)
CORS(app)

# Inicializando o modelo
model = SleepQualityModel(data_path='data/student_sleep_patterns.csv')
model.train() # Treine o modelo ao iniciar
model.save_model() # Salve o modelo treinado
```

Função predict APP.PY

```
@app.route('/predict', methods=['POST'])
def predict():
    try:
        data = request.json
        # Valida os dados recebidos
        if not all(key in data for key in ['Age', 'Study_Hours', 'Screen_Time', 'Caffeine_Intake', 'Physical_Activity', 'Sleep_Duration', 'Gender']):
            return jsonify({"error": "Dados incompletos"}), 400

        symptoms = [
            int(data['Age']),
            float(data['Study_Hours']),
            float(data['Screen_Time']),
            int(data['Caffeine_Intake']),
            int(data['Physical_Activity']),
            float(data['Sleep_Duration']),
            int(data['Gender'])
        ]

        model.load_model() # Carrega o modelo para fazer previsões
        prediction = model.predict(symptoms)

        return jsonify({"prediction": int(prediction)})

    except Exception as e:
        return jsonify({"error": str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

INTERFACE

Diagnóstico de Qualidade do Sono para estudantes

* Idade ?

Selecione seu gênero ? ▾

* Horas de Estudo: ?

* Tempo de Tela: ?

* Ingestão de Cafeína: ?

* Atividade Física (minutos): ?

* Duração do Sono: ?

Enviar

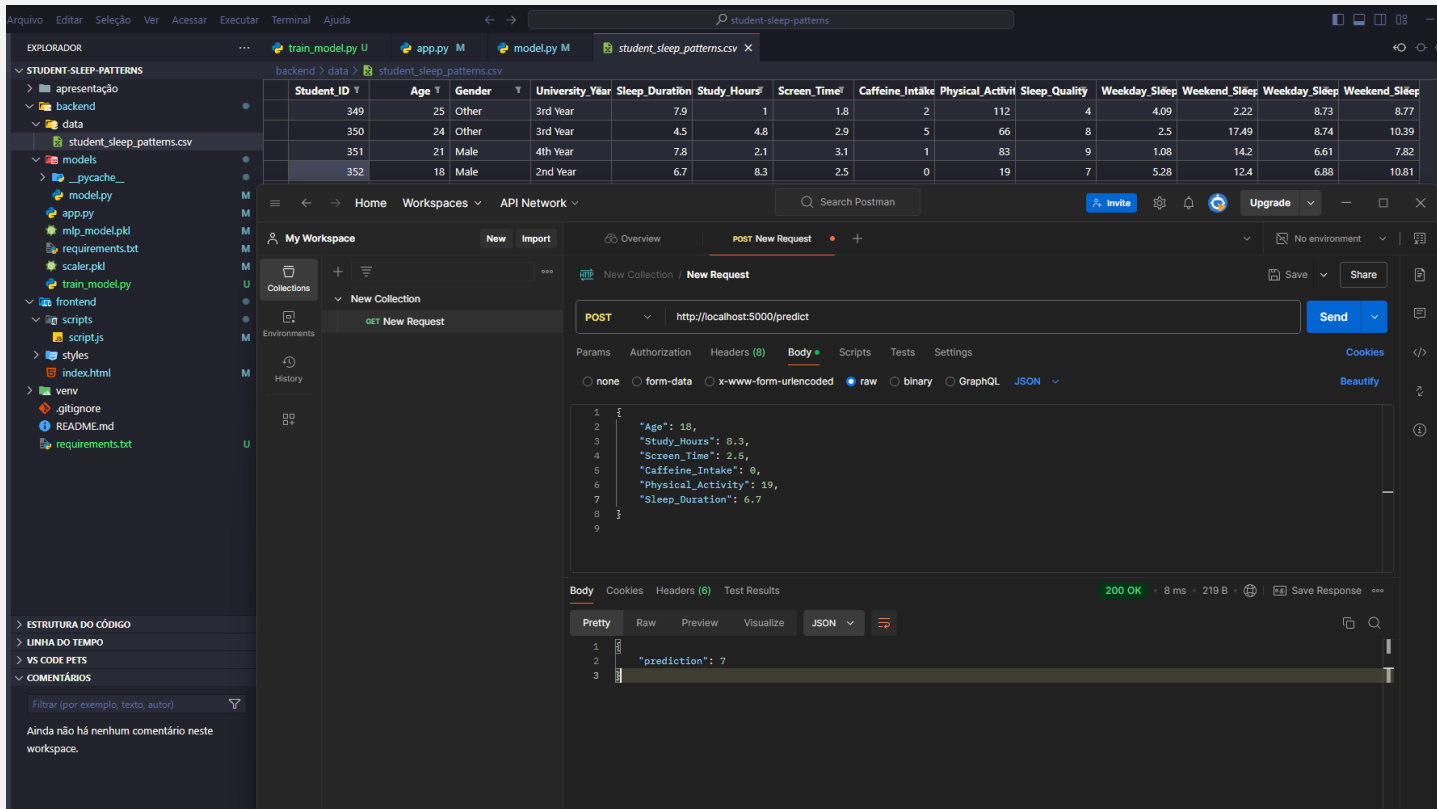
Diagnóstico: 1

Seu padrão de sono está muito abaixo do ideal. Recomenda-se aumentar a quantidade de sono e reduzir a cafeína.



Exportar como Imagem

TESTES



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure for 'STUDENT-SLEEP-PATTERNS' with folders like 'apresentação', 'backend', 'data', 'models', 'frontend', and 'scripts'. The 'data' folder is expanded, showing 'student_sleep_patterns.csv'. The main editor area displays the contents of this CSV file, which contains student sleep data. Below the CSV, a REST client request is configured in the 'New Request' tab. The request is a POST to 'http://localhost:5000/predict' with a JSON body containing student attributes. The response is shown as '200 OK' with a 'prediction' of 7.

Student_ID	Age	Gender	University_Year	Sleep_Duration	Study_Hours	Screen_Time	Caffeine_Intake	Physical_Activity	Sleep_Quality	Weekday_Sleep	Weekend_Sleep	Weekday_Sleep	Weekend_Sleep
349	25	Other	3rd Year	7.9	1	1.8	2	112	4	4.09	2.22	8.73	8.77
350	24	Other	3rd Year	4.5	4.8	2.9	5	66	8	2.5	17.49	8.74	10.39
351	21	Male	4th Year	7.8	2.1	3.1	1	83	9	1.08	14.2	6.61	7.82
352	18	Male	2nd Year	6.7	8.3	2.5	0	19	7	5.28	12.4	6.88	10.81

```
POST http://localhost:5000/predict

{
  "Age": 18,
  "Study_Hours": 8.3,
  "Screen_Time": 2.5,
  "Caffeine_Intake": 0,
  "Physical_Activity": 19,
  "Sleep_Duration": 6.7
}
```

Body Cookies Headers (6) Test Results

200 OK · 8 ms · 219 B

```
1 {
2   "prediction": 7
3 }
```

TESTES



127.0.0.1:5500/frontend/index.html?

Horas de Estudo:

9.1

Tempo de Tela:

2.8

Ingestão de Cafeína:

5

Atividade Física (minutos):

14

Duração do Sono:

8.6

Enviar

Predição: 1

Predição

1.0

0.9

0.8

0.7

0.6

Workspaces

More

POST New Request

http://localhost:5000/predict

Send

Params Auth Headers (8) Body Scripts Tests Settings

raw JSON

Comment

```
"Age": 19,  
"Study_Hours": 9.1,  
"Screen_Time": 2.8,  
"Caffeine_Intake": 5,  
"Physical_Activity": 14,  
"Sleep_Duration": 8.6,  
"Gender": 0 // 0 para Male, 1 para Female, 2 para Other
```

Body

200 OK 9 ms 219 B

Pretty Raw Preview Visualize JSON

```
{  
  "prediction": 1  
}
```



Conclusão!

Resultados esperados: Melhoria na compreensão dos padrões de sono dos estudantes e possibilidades de intervenções para otimizar a qualidade do sono.

Próximos passos: Aperfeiçoamento do modelo e potencial uso em larga escala para acompanhar hábitos de sono.

Reflexões: O impacto positivo de previsões personalizadas baseadas em IA para a saúde dos estudantes.



Referências

- <https://www.kaggle.com/datasets/arsalanjamal002/student-sleep-patterns>Cms concept illustration in
- [Projeto com modelo de classificação perceptron multi-camadas \(MLP\) | by Ana Paula Ferrari Januário | Medium](#)
- https://www.youtube.com/watch?v=cO59qwb3Haw&ab_channel=LIITUFG
- https://www.youtube.com/watch?v=BNjRnIDQqOc&ab_channel=Caf%C3%A9Computa%C3%A7%C3%A3o
- https://www.youtube.com/watch?v=eiQkrCSV8w&ab_channel=Jos%C3%A9Ricardo

