

Dormindo como uma pedra: Explorando fatores que moldam a qualidade do sono com modelos preditivos

Henrique César Higino Holanda Cordeiro
Rafael do Nascimento Moura

Introdução

Contexto

A importância do sono



zz



Estilo de vida moderno



Problemas relacionados



Objetivo do projeto

Objetivos

Geral e específicos



Verificar a viabilidade em desenvolver um modelo preditivo baseado em aprendizado profundo para análise e previsão da qualidade do sono com base em alguns fatores comportamentais e fisiológicos.



- Identificar padrões e relações que influenciam a qualidade do sono.
- Implementar e treinar modelos de aprendizado (GBM e MLP) para prever a qualidade do sono.
- Avaliar o desempenho dos modelos utilizando métricas adequadas.
- Identificar os fatores mais relevantes que afetam a qualidade do sono .

Justificativa

Contribuições



- Combinação da robustez dos modelos preditivos com a praticidade das soluções tecnológicas modernas para promover um estilo de vida saudável.
- Desenvolvimento de ferramentas de auto-gestão do sono.
- Integração em aplicativos móveis e sistemas de monitoramento com feedback em tempo real e recomendações personalizadas.

Analise Exploratória dos Dados

Banco de Dados

	User ID	Age	Gender	Sleep Quality	Bedtime	Wake-up Time	Daily Steps	Calories Burned	Physical Activity Level	Dietary Habits	Sleep Disorders	Medication Usage
0	1	25	f	8	23:00	06:30	8000	2500	medium	healthy	no	no
1	2	34	m	7	00:30	07:00	5000	2200	low	unhealthy	yes	yes
2	3	29	f	9	22:45	06:45	9000	2700	high	healthy	no	no
3	4	41	m	5	01:00	06:30	4000	2100	low	unhealthy	yes	no
4	5	22	f	8	23:30	07:00	10000	2800	high	medium	no	no
...
95	96	43	m	7	00:45	07:15	6500	2400	medium	medium	no	no
96	97	33	f	8	23:15	06:15	8500	2600	high	medium	no	no
97	98	46	m	4	01:30	07:00	3000	2000	low	unhealthy	yes	yes
98	99	25	f	9	22:15	06:45	9500	2700	high	healthy	no	no
99	100	41	m	6	00:30	07:00	5000	2200	medium	unhealthy	no	no

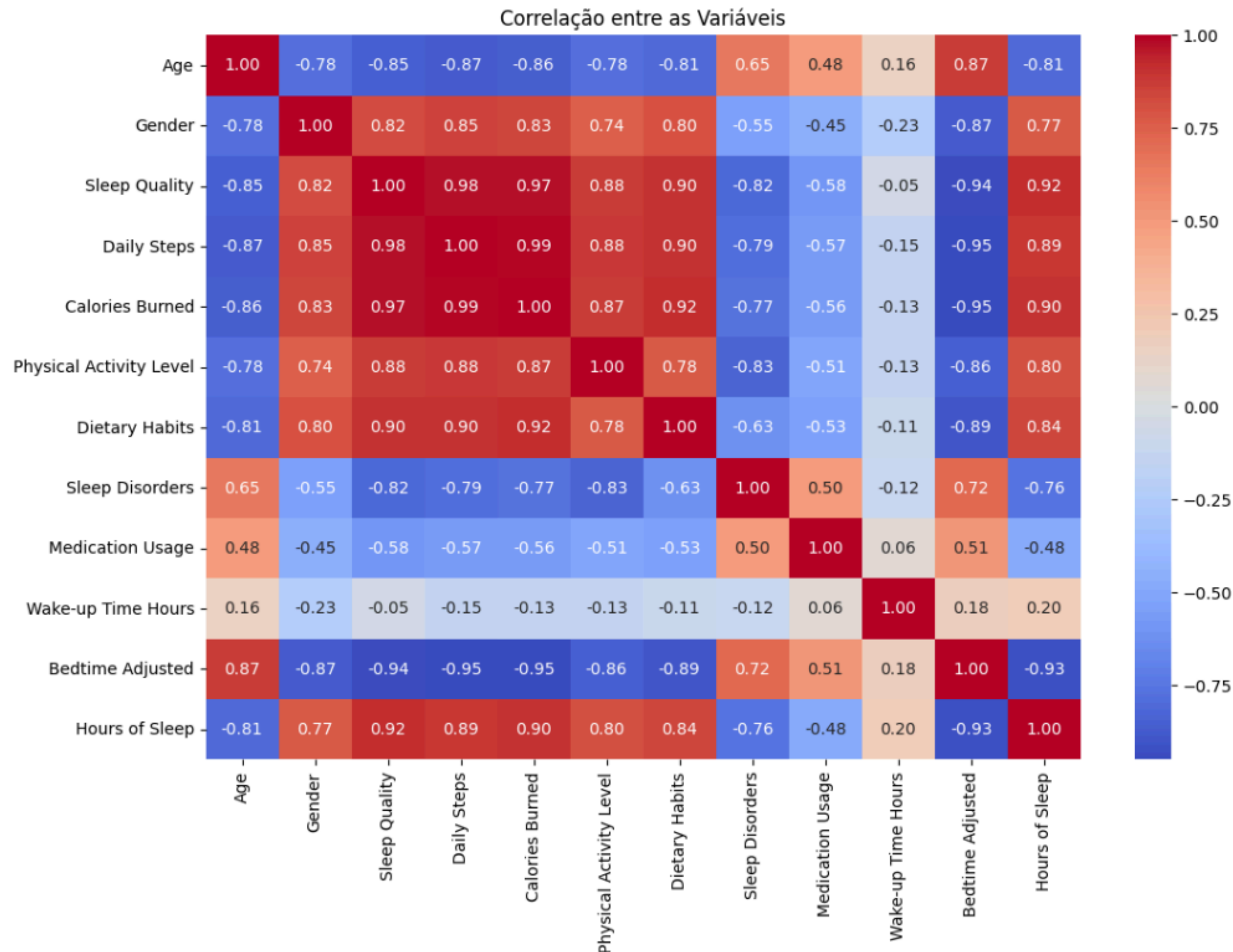
Health and Sleep Statistics

Pré-processamento

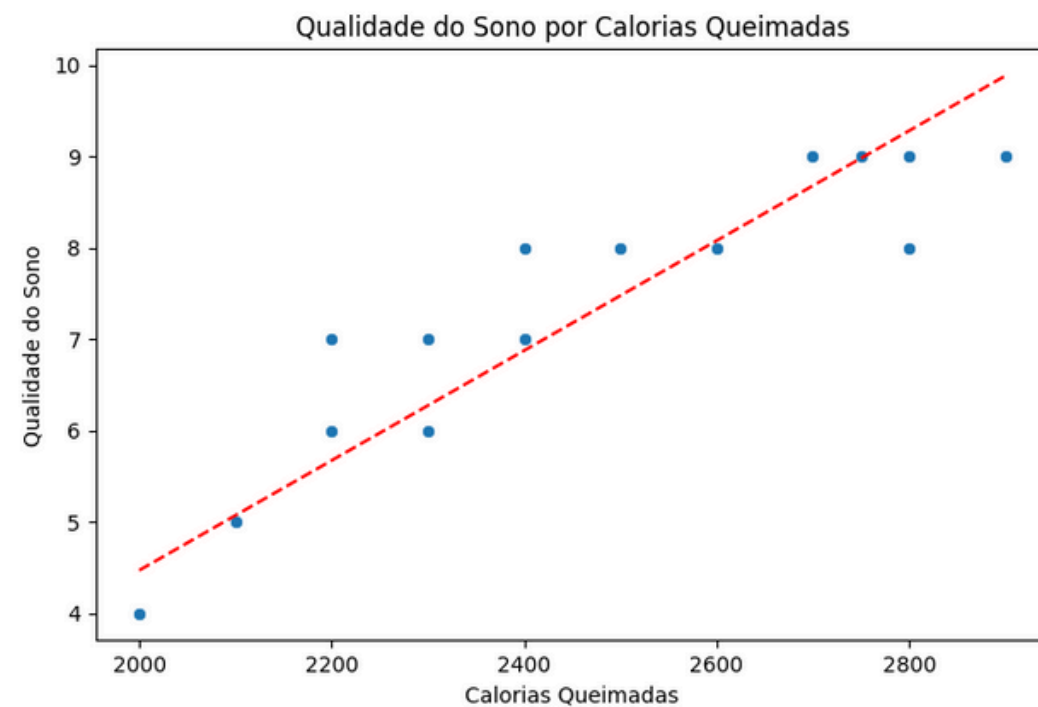


- Tratamento de valores ausentes
- Codificação de variáveis categóricas
- Normalização
- Divisão dos dados
- Análise Exploratória

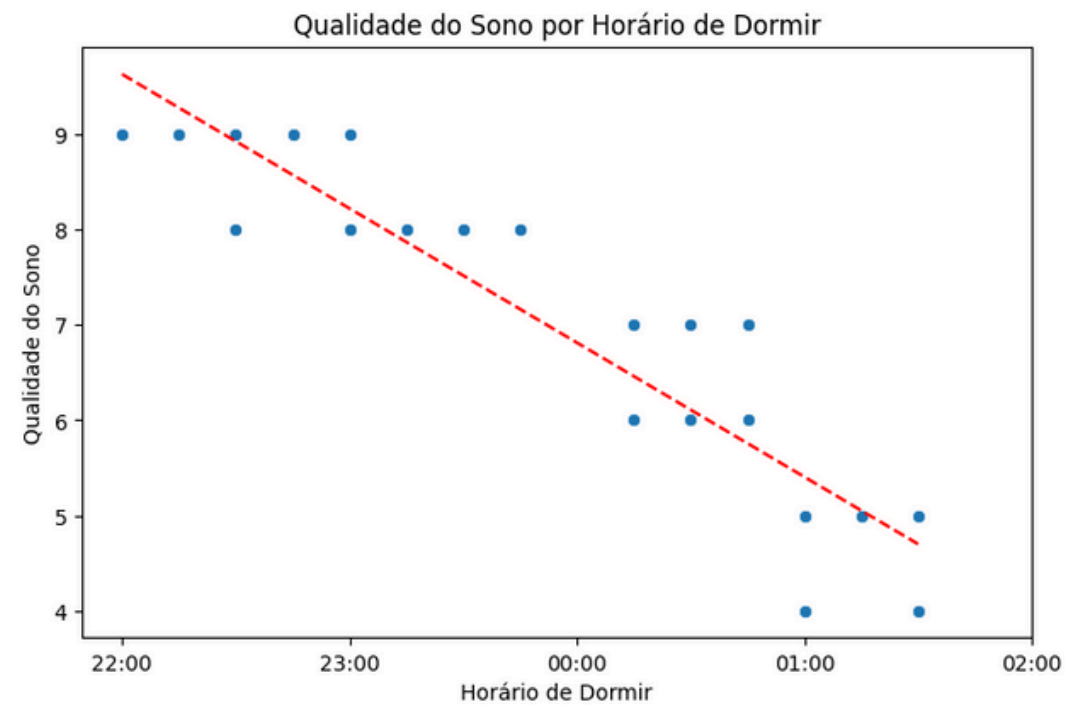
Análise Explanatória



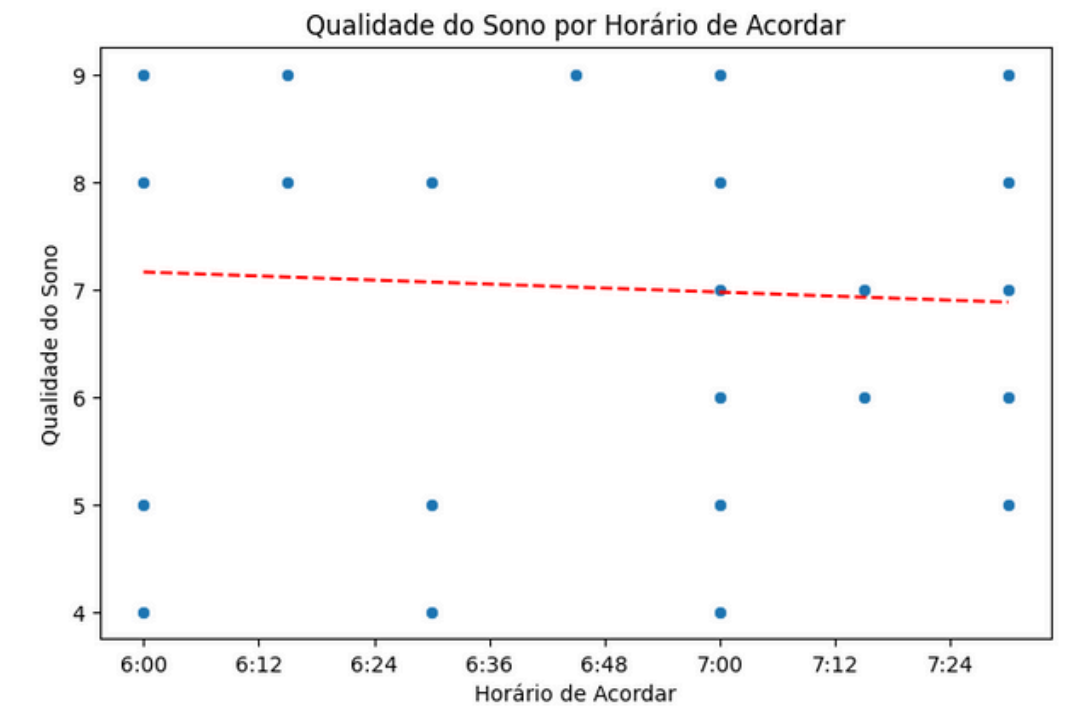
Análise Explanatória



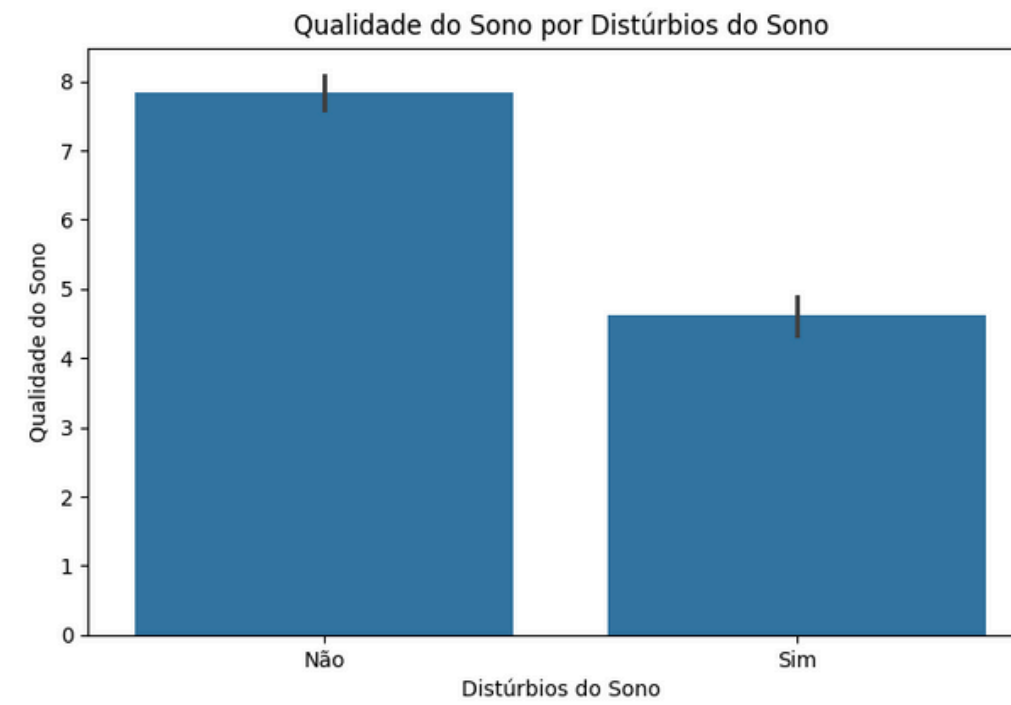
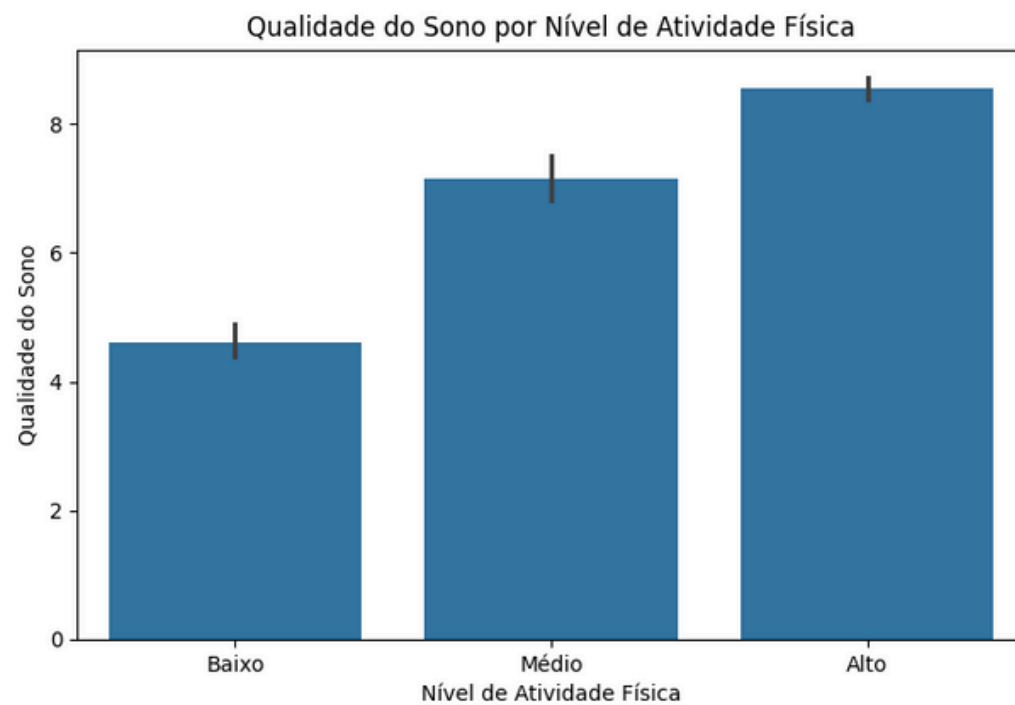
Coef. de Correlação: 0.97
Coef. de Regressão: 0.01
Coef. de Determinação: 0.95



Coef. de Correlação: -0.94
Coef. de Regressão: -1.41
Coef. de Determinação: 0.89



Coef. de Correlação: -0.05
Coef. de Regressão: -0.19
Coef. de Determinação: 0.00



Análise Explanatória

Coeficientes de Correlação:

Atributo	Coef. de Correlação
Daily Steps	0.976277
Calories Burned	0.973557
Hours of Sleep	0.918909
Dietary Habits	0.902498
Physical Activity Level	0.875203
Wake-up Time Hours	-0.048526
Medication Usage	-0.579284
Sleep Disorders	-0.818806
Age	-0.848356
Bedtime Adjusted	-0.941617

Coeficientes de Determinação (R²):

Atributo	Coef. de Determinação
Daily Steps	0.953117
Calories Burned	0.947813
Bedtime Adjusted	0.886643
Hours of Sleep	0.844393
Dietary Habits	0.814502
Physical Activity Level	0.765981
Age	0.719708
Sleep Disorders	0.670443
Medication Usage	0.335570
Wake-up Time Hours	0.002355

Coeficientes de Regressão (R):

Atributo	Coef. de Regressão
Physical Activity Level	1.934426
Dietary Habits	1.881564
Hours of Sleep	1.369410
Calories Burned	0.006010
Daily Steps	0.000678
Age	-0.174085
Wake-up Time Hours	-0.187126
Bedtime Adjusted	-1.409172
Medication Usage	-2.500000
Sleep Disorders	-3.222453

Criar e Treinar os Modelos

Conjuntos de Treino, Teste e Validação

```
# Preparar os dados para os modelos
# Selecionar as colunas para o treinamento
features = ['Age', 'Gender', 'Daily Steps', 'Calories Burned', 'Hours of Sleep', 'Physical Activity Level', 'Dietary Habits', 'Sleep Disorders', 'Medication Usage', 'Bedtime Adjusted']
x = df[features]
y = df['Sleep Quality']
```

- Dividir os dados em conjuntos de Treino, Validação e Teste para MLP
- Normalizar as variáveis numéricas
- Aplicar One-Hot Encoding nas variáveis categóricas
- Dividir os dados em conjuntos de Treino, Validação e Teste para CatBoost

Hiperparâmetros do MLP

```
# Definir os hiperparâmetros para ajuste
param_grid = {
    'hidden_layer_sizes': [(25, 50), (50, 25), (50, 50), (100, 50), (50, 100), (100, 100)],
    'activation': ['relu', 'tanh'],
    'alpha': [0.0001, 0.001],
    'solver': ['adam', 'sgd', 'lbfgs'],
    'max_iter': [500, 750, 1000]
}

# Configurar o GridSearchCV
grid_search = GridSearchCV(MLPRegressor(random_state=42), param_grid, cv=3, scoring='neg_mean_squared_error', verbose=2, n_jobs=-1)

# Ajustar o modelo
grid_search.fit(X_train, y_train)
```

Hiperparâmetros otimizados:

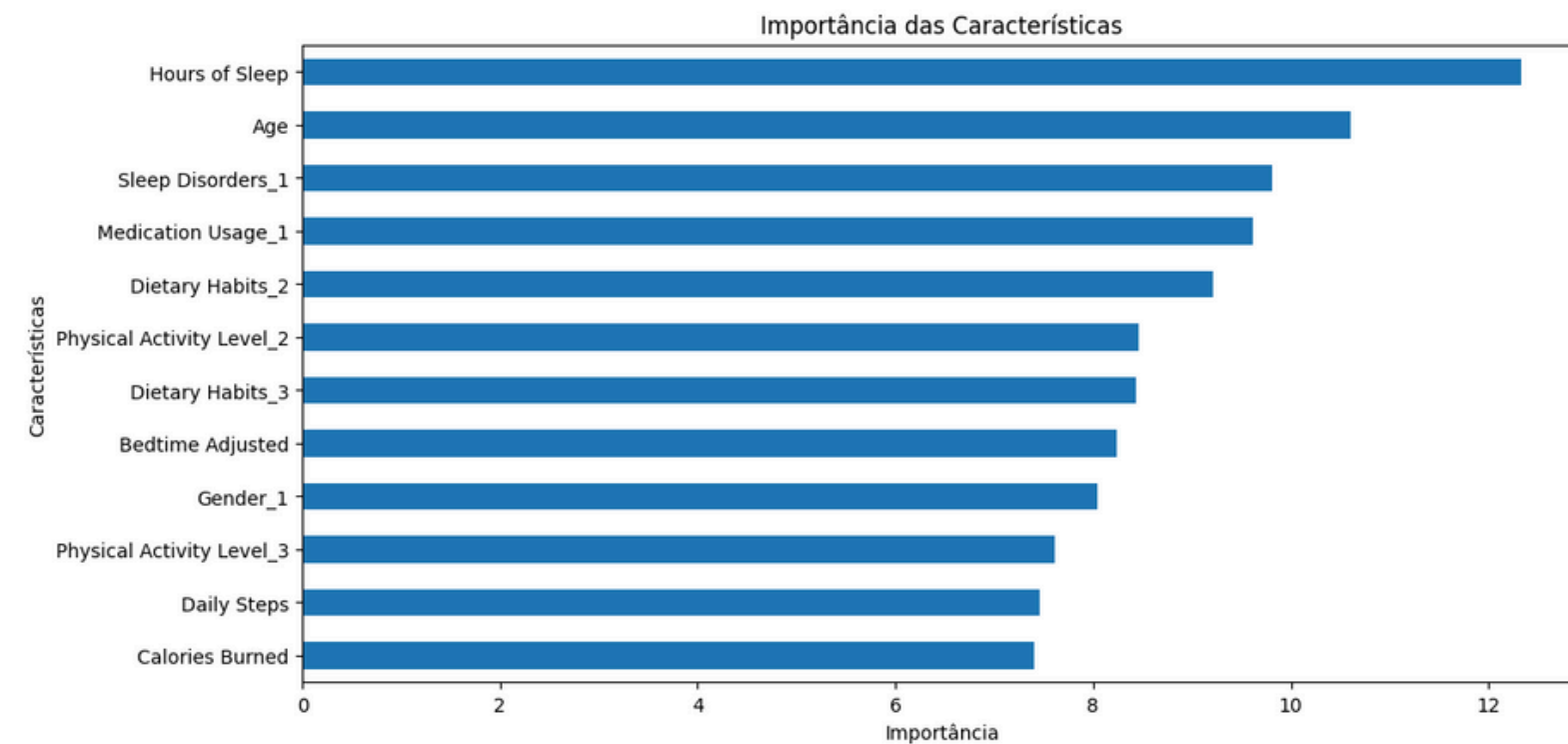
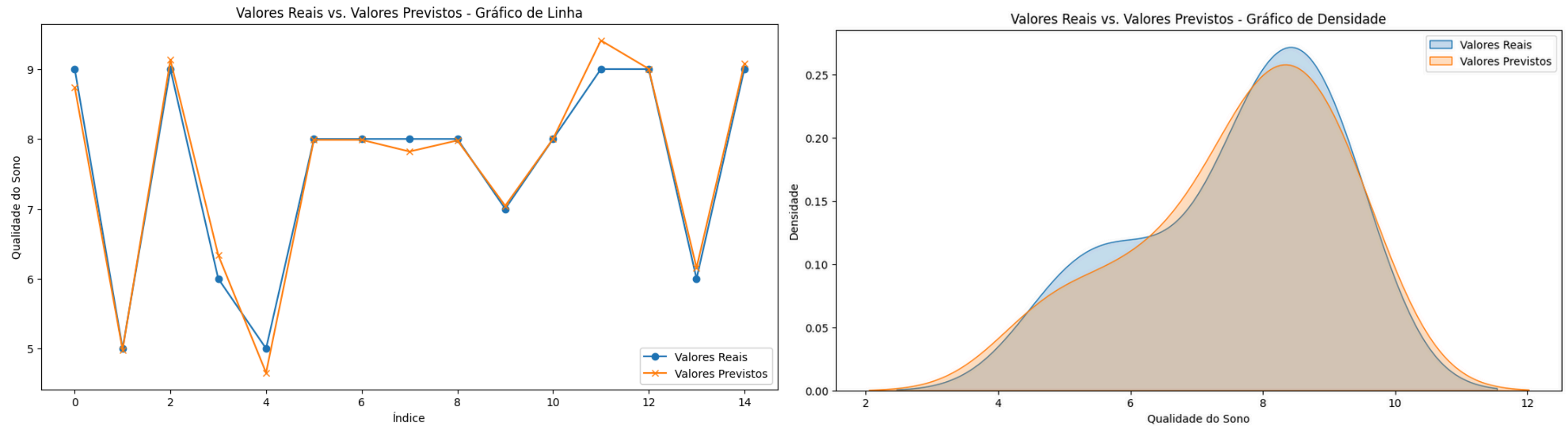
```
{'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes': (100, 50), 'max_iter': 1000, 'solver': 'lbfgs'}
```

Treinamento MLP

```
# Função de treino e avaliação
def train_and_evaluate_mlp(X_train, y_train, X_val, y_val):
    mlp_model = grid_search.best_estimator_
    mlp_model.fit(X_train, y_train)
    y_val_pred_mlp = mlp_model.predict(X_val)
    rmse_mlp = np.sqrt(mean_squared_error(y_val, y_val_pred_mlp))
    r2_mlp = r2_score(y_val, y_val_pred_mlp)
    importances = np.abs(mlp_model.coefs_[0]).sum(axis=1)
    feature_importances_mlp = pd.Series(importances, index=X_train.columns)
    return rmse_mlp, r2_mlp, y_val_pred_mlp, feature_importances_mlp

# Resultados do treinamento
rmse_mlp, r2_mlp, y_val_pred_mlp, feature_importances_mlp = train_and_evaluate_mlp(X_train, y_train, X_val, y_val)
```

Treinamento MLP

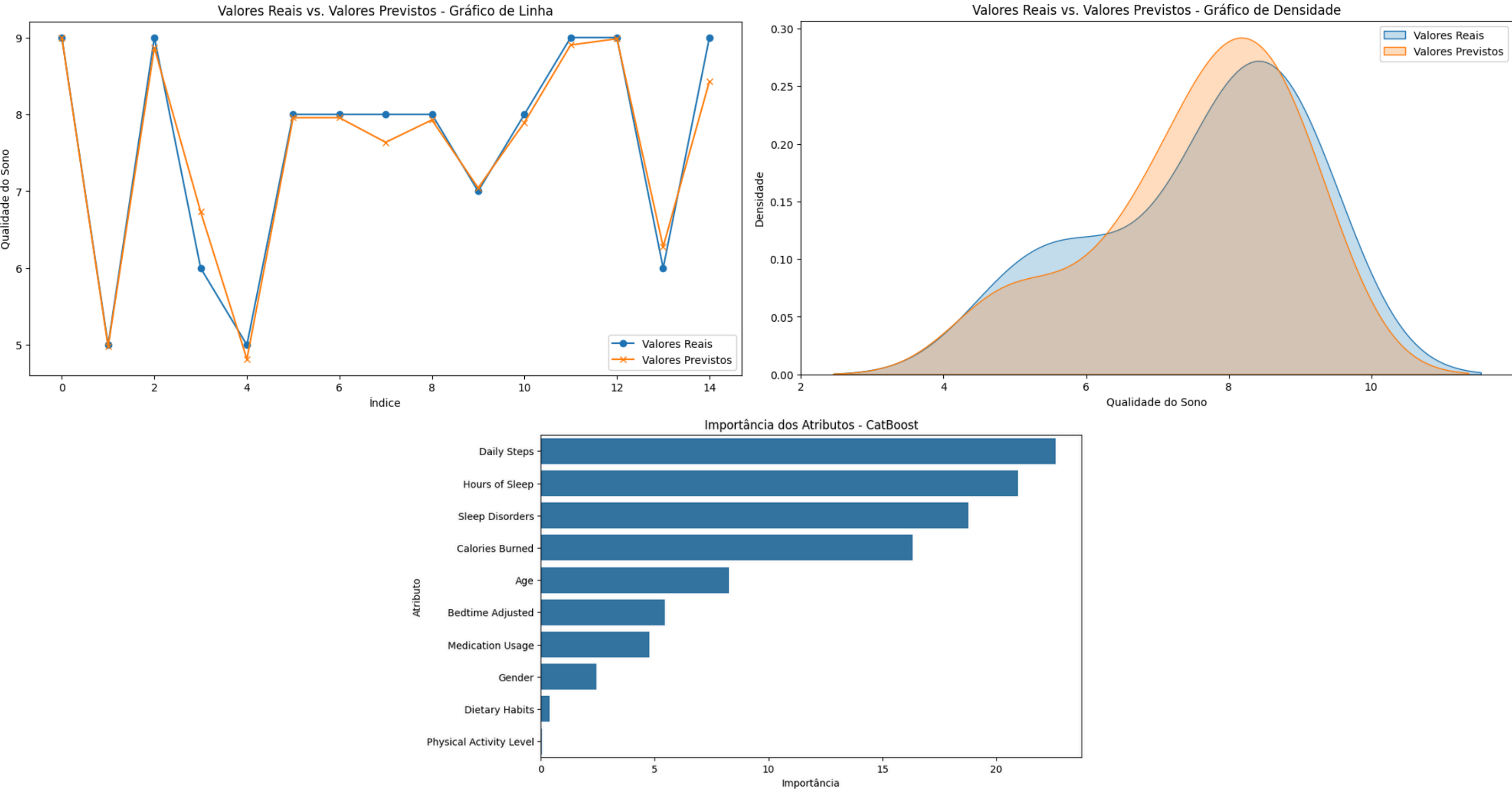


Treinamento CatBoost

```
# Função de treinamento do CatBoost
def train_and_evaluate_catboost(X_train, y_train, X_val, y_val):
    categorical_cols = ['Sleep Disorders', 'Medication Usage', 'Gender', 'Dietary Habits', 'Physical Activity Level']
    for col in categorical_cols:
        X_train[col] = X_train[col].astype(str)
        X_val[col] = X_val[col].astype(str)
    cat_features_indices = [X_train.columns.get_loc(col) for col in categorical_cols]
    cat_model = CatBoostRegressor(iterations=cat_model_init.get_best_iteration(), learning_rate=0.1, depth=6, loss_function='RMSE', verbose=10, random_seed=42)
    cat_model.fit(X_train, y_train, cat_features=cat_features_indices)
    y_val_pred_catboost = cat_model.predict(X_val)
    rmse_catboost = np.sqrt(mean_squared_error(y_val, y_val_pred_catboost))
    r2_catboost = r2_score(y_val, y_val_pred_catboost)
    feature_importances_cat = cat_model.get_feature_importance()
    return rmse_catboost, r2_catboost, y_val_pred_catboost, feature_importances_cat

# Resultados do treinamento
rmse_cat, r2_cat, y_val_pred_cat, feature_importances_cat = train_and_evaluate_catboost(X_train_cat, y_train_cat, X_val_cat, y_val_cat)
```

Treinamento CatBoost



Avaliação

Validação Cruzada

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)

# Listas para armazenar os resultados
results_mlp = []
results_catboost = []
predictions_mlp = []
predictions_catboost = []

# Loop de validação cruzada
for train_index, val_index in kf.split(x):
    X_train, X_val = x.iloc[train_index], x.iloc[val_index]
    y_train, y_val = y.iloc[train_index], y.iloc[val_index]

    # MLP
    rmse_mlp, r2_mlp, y_pred_mlp, importances_mlp = train_and_evaluate_mlp(X_train, y_train, X_val, y_val)
    results_mlp.append((rmse_mlp, r2_mlp))
    predictions_mlp.append((y_val, y_pred_mlp))

    # CatBoost
    rmse_catboost, r2_catboost, y_pred_cat, importances_cat = train_and_evaluate_catboost(X_train, y_train, X_val, y_val)
    results_catboost.append((rmse_catboost, r2_catboost))
    predictions_catboost.append((y_val, y_pred_cat))
```

Validação Cruzada

Resultados da Validação Cruzada - MLP:

Fold 1 - RMSE: 0.35, R^2 : 0.96

Fold 2 - RMSE: 0.48, R^2 : 0.90

Fold 3 - RMSE: 0.20, R^2 : 0.99

Fold 4 - RMSE: 0.28, R^2 : 0.98

Fold 5 - RMSE: 0.38, R^2 : 0.92

Validação Cruzada - RMSE médio: 0.34 ± 0.09

Validação Cruzada - R^2 médio: 0.95 ± 0.03

Resultados da Validação Cruzada - CatBoost:

Fold 1 - RMSE: 0.21, R^2 : 0.99

Fold 2 - RMSE: 0.37, R^2 : 0.94

Fold 3 - RMSE: 0.21, R^2 : 0.99

Fold 4 - RMSE: 0.15, R^2 : 0.99

Fold 5 - RMSE: 0.44, R^2 : 0.90

Validação Cruzada - RMSE médio: 0.28 ± 0.11

Validação Cruzada - R^2 médio: 0.96 ± 0.04

- Eficiência
- Viabilidade
- Potencial

Conclusão

Conclusão

- Atributos
- Modelos
- Considerações finais

Proposta

Proposta de uso

- Uso profissional



- Uso pessoal



