## Treinamento da IA

Os dados que consideramos mais importantes para passar para a IA foram o dia, a hora, a categoria e a distância de uma corrida, esses fatores são os que mais influenciam no preço e portanto são os mais importantes para prever preços de corridas.

### Criando o treino e teste

Com a base já limpa e integrada nós criamos o treino e o teste da IA

```
[10] x = df.drop(['Preco', 'ID_Corrida'], axis=1)
    y = df['Preco']

scaler = StandardScaler()
scaled_x = scaler.fit_transform(x)

treino_x, teste_x, treino_y, teste_y = train_test_split(
    scaled_x,y,test_size=0.3, random_state=42)

display(teste_x)
display(teste_x)
```

## Analisando o desempenho dos modelos

Em seguida usamos esse treino e teste em alguns modelos diferentes para ver qual obteria a melhor performance

```
[16] # Função para fazer a avaliação de um modelo específicado def avaliacao(modelo, nome):
    modelo.fit(treino_x, treino_y)

    predicao = modelo.predict(teste_x)
        MAE = mean_absolute_error(teste_y, predicao)
        MSE = mean_squared_error(teste_y, predicao)
        RMSE = np.sqrt(MSE)
        R2 = r2_score(teste_y, predicao)

    print(f*Modelo: {nome}\n')
    print(f*MAE: {MAE:.2f}\n")
    print(f*MSE: {MSE:.2f}\n")
    print(f*MSE: {MSE:.2f}\n")
    print(f*NSSE: {RSE:.2f}\n")
    print(f*nNSE: {RSE:.2f}\n")
    print(f*nNSE: (R2:.2f}\n")
    print(f\n"\n")

    # Passando diversos modelos pela função de análise de desempenho
    avaliacao(NeighborsRegressor(n_esibhors=5), "KNN")
    avaliacao(NeighborsRegressor(n_esibhors=5), "KNN")
    avaliacao(QecisionTreeRegressor(n_esibhors=5), "KNN")
    avaliacao(MandomForestRegressor(n_estimators=100), "Random Forest")
```

Os modelos testados foram:

- LogisticRegression
- KNeighborsRegressor
- DecisionTreeRegressor
- RandomForestRegressor

## Definindo o modelo

O modelo que apresentou a melhor performance foi o RandomForest, então esse vai ser o modelo usado para o projeto

```
[13] # Escolhendo o modelo
modelo = RandomForestRegressor(n_estimators=100)

# Treino do modelo escolhido
modelo.fit(treino_x, treino_y)

Modelo: Regressão Logistica

AMAE: 20.89%
AMS: 120.89%
NSS: 15088.85
RMSE: 126.13
R? - 1.78

Modelo: KNN

AMAE: 81.08E
AMS: 98.95
R? - -0.44

Modelo: Árvore de Decisão

MAE: 62.1%
NSSE: 98.95
RPSE: 84.40
RPSE: 84.40
RPSE: 84.40
Modelo: Random Forest

MAE: 16.65%
MOdelo: Random Forest

MAE: 15.65%
MODELO: Random Forest

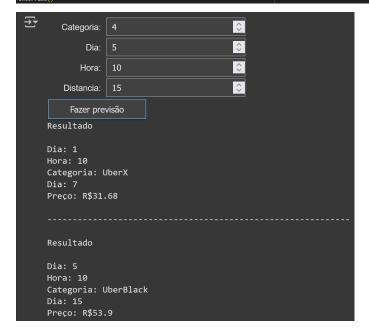
MODELO: Random Forest

MAE: 15.65%
MODELO: Random Forest
```

Ainda precisamos refinar mais o treinamento da IA para obter uma melhor performance, mas já temos um framework pronto para um MVP e para poder começar a evoluir

# Interface de previsão

Com o modelo já definido e treinado criamos uma pequena interface no próprio Colab para poder entrar com novos dados e fazer previsões novas



#### Link do Colab

O arquivo .ipnyb pode ser encontrado no diretório da entrega 2, segue o link para o notebook

co Treinamento IA Khipo