

Dashboard:

## Trafego Urbano

### Avaliação do Modelo "Random Forest Regressor" para Previsão de Tráfego

**Nome do Modelo:** Random Forest Regressor **Melhores Hiperparâmetros:**  
n\_estimators: 300, min\_samples\_split: 2, min\_samples\_leaf: 1, max\_depth: 20

---

#### 1. Métricas de Erro Absoluto: MSE e RMSE

- **MSE (Mean Squared Error):** 182.817
- **RMSE (Root Mean Squared Error):** 13.521

**Análise:** O RMSE é a raiz quadrada do MSE e está na mesma unidade da variável alvo (travel\_time). Um RMSE de 13.52 significa que, em média, as previsões do seu modelo desviam dos valores reais em cerca de 13.52 unidades de tempo.

Assumindo que travel\_time esteja em minutos (o mais comum para esse tipo de problema), este RMSE indica que o modelo ainda tem uma margem considerável de erro.

---

#### 2. Métrica de Desempenho Relativo: $R^2$ (Coeficiente de Determinação)

- **$R^2$ :** 0.2486 (aproximadamente 24.86%)
- **Cross-Validation  $R^2$ :** 0.1756 (aproximadamente 17.56%)

**Análise:** O  $R^2$  mede a proporção da variância na variável dependente que é explicável pelas variáveis independentes do modelo. Ele varia de 0 a 1, onde 1 indica um ajuste perfeito (o modelo explica toda a variância).

- Um  $R^2$  de 0.2486 é **bastante baixo**. Isso significa que o seu modelo de Random Forest explica apenas cerca de **25% da variância total** no tempo de viagem. Os outros 75% da variância permanecem inexplicados pelo modelo e pelas features que ele utiliza.
- O Cross-Validation  $R^2$  ainda menor (0.1756) sugere que a performance do modelo é consistentemente baixa em diferentes subconjuntos dos dados de treinamento, reforçando a ideia de que o modelo não está capturando bem os padrões.

**Conclusão sobre o  $R^2$ :** Este é o indicador mais crítico. Um  $R^2$  tão baixo sugere que o modelo tem um **desempenho pobre** e não é muito eficaz em prever o tempo de

viagem. Ele está sobajustando (underfitting) os dados, ou seja, não está aprendendo as relações complexas ou importantes presentes nos dados.

---

### 3. Análise dos Hiperparâmetros

- **n\_estimators: 300:** Um bom número de árvores na floresta, geralmente suficiente para estabilizar as previsões.
- **max\_depth: 20:** Uma profundidade razoável que permite às árvores aprenderem padrões complexos sem ir ao extremo de superajustar o ruído (overfitting) por si só.
- **min\_samples\_split: 2:** O número mínimo de amostras necessárias para dividir um nó interno. O valor mínimo (2) significa que o modelo pode tentar dividir quase qualquer nó, permitindo árvores mais detalhadas.
- **min\_samples\_leaf: 1:** O número mínimo de amostras necessárias para que um nó seja uma folha. O valor mínimo (1) também permite folhas muito específicas.

**Análise:** Os hiperparâmetros escolhidos são típicos para um Random Forest e parecem sensatos. No entanto, o baixo  $R^2$  com esses parâmetros sugere que o problema não reside primariamente na otimização fina desses hiperparâmetros, mas em questões mais fundamentais como as features de entrada ou a adequação do modelo ao problema.

---

### Conclusão Geral e Próximos Passos Sugeridos

Com base nessas métricas, o modelo atual **não está performando bem** para a previsão do tempo de viagem. O  $R^2$  extremamente baixo é a principal preocupação, indicando que o modelo não está capturando os fatores que realmente influenciam o tempo de viagem.

#### Recomendações para Melhoria:

1. **Engenharia de Features é Fundamental:** Esta é, de longe, a área com maior potencial de melhoria.
  - **Mais Variáveis Temporais:** Além de 'hour', 'day\_of\_week', considere 'mês', 'dia do ano', 'semana do ano'.
  - **Variáveis Cíclicas:** Para hour e day\_of\_week, considere codificação seno/cosseno para capturar a natureza cíclica (ex: 23h é "próximo" de 01h).

- **Interações:** Crie features de interação, como `hour * is_holiday` ou `day_of_week * weather_rainy`.
- **Horário de Pico:** Crie uma feature binária para indicar se é horário de pico (rush hour).
- **Distância/Segmento da Estrada:** Se os dados permitirem, inclua informações sobre a distância percorrida ou o segmento específico da estrada.
- **Dados Externos:** Integre dados de eventos reais (acidentes, obras) se disponíveis.

## 2. Experimentar Outros Modelos Mais Potentes:

- **Gradient Boosting Machines (GBM):** Modelos como **XGBoost**, **LightGBM** ou **CatBoost** são frequentemente os melhores em problemas de regressão com dados tabulares. Eles são mais eficazes em capturar relações complexas e interações de features.

## 3. Gerar Mais Dados (testar com dados reais)

### Gráficos:

#### Random Forest: Real vs Previsto

**O que esses gráficos representam:** Esses gráficos de dispersão comparam os valores reais (Valor Real no eixo X) com os valores previstos pelo modelo (Valor Previsto no eixo Y) para o conjunto de teste. Em um modelo de regressão perfeito, todos os pontos estariam alinhados em uma diagonal de 45 graus (onde Valor Previsto = Valor Real). Quanto mais dispersos os pontos estiverem dessa linha, pior é o desempenho do modelo.

### Análise:

- **Dispersão:** Ambos os gráficos mostram uma **alta dispersão** dos pontos. Os pontos não estão agrupados perto de uma linha diagonal, mas sim espalhados por uma grande área. Isso é um forte indicativo de que **nenhum dos modelos está fazendo previsões precisas**.
- **Padrão:** É possível observar uma leve tendência de alta (conforme o valor real aumenta, o previsto também tende a aumentar), mas essa tendência é muito fraca.
- **Concentração em Faixas:** Observamos que as previsões (e talvez os valores reais) parecem se agrupar em faixas horizontais e verticais em alguns lugares. Isso pode ser um artefato dos dados (ex: tempo de viagem

tende a ser em múltiplos de 5 ou 10 minutos, ou variáveis categóricas discretizam muito o espaço).

- **Subestimação/Superestimação:**

- Para o **Random Forest**, a dispersão é similar, talvez um pouco mais densa no centro, mas também não demonstra uma capacidade preditiva forte.

**Relação com as Métricas:** Esses gráficos visuais **confirmam as baixas métricas de  $R^2$**  (0.2486 para Random Forest e 0.2486 para Linear Regression, usando o  $R^2$  que você viu na sua avaliação, embora no gráfico apareça o RF melhor com 0.2486 e o LR com 0.2486, eles estão no mesmo patamar, talvez o LR fosse ligeiramente pior). A grande dispersão e a falta de alinhamento diagonal são a representação visual de um modelo que explica muito pouco da variância real.

### **Histograma: "Distribuição do Erro (Real - Previsto)"**

**O que este gráfico representa:** Este histograma mostra a distribuição dos erros de previsão, calculados como Real - Previsto. Um modelo ideal teria seus erros centrados em zero e com uma distribuição aproximadamente normal (formato de sino), indicando que o modelo não tem um viés sistemático (não está consistentemente superestimando ou subestimando) e que a maioria dos erros são pequenos.

#### **Análise:**

- **Centro:** A distribuição está **centrada aproximadamente em zero**, o que é um bom sinal e indica que o modelo não tem um viés sistemático forte (não está superestimando ou subestimando consistentemente os tempos de viagem na média).
- **Amplitude/Dispersão:** A distribuição dos erros é **bastante ampla**. Os erros variam de aproximadamente -40 a +70 (assumindo que o eixo X representa as mesmas unidades do travel\_time, provavelmente minutos). Isso significa que, embora a média seja próxima de zero, há muitos erros grandes, tanto positivos (modelo subestimou muito) quanto negativos (modelo superestimou muito).
- **Forma:** A distribuição parece ser um pouco assimétrica à direita, com uma "cauda" mais longa para erros positivos (onde Real - Previsto é positivo, significando que o modelo previu um valor menor do que o real). Isso pode indicar que o modelo tem mais dificuldade em prever com precisão tempos

de viagem mais longos ou em situações de congestionamento inesperado, subestimando-os com mais frequência.

**Relação com as Métricas:** A ampla dispersão do histograma de erros é consistente com o alto RMSE (13.52) e o baixo  $R^2$  (0.2486). Embora o modelo esteja "sem viés" na média, a magnitude dos erros individuais é grande, resultando em uma precisão geral baixa.

## **Classificação de Saúde Pública**

### **Informações do Modelo:**

- **Nome do Modelo:** Decision Tree
- **Melhores Hiperparâmetros:** max\_depth: 15, min\_samples\_split: 2
- **Acurácia Geral:** 0.77

### **Análise Detalhada:**

#### **1. Acurácia Geral (0.77) e Relatório de Classificação (Tabela)**

A acurácia geral de 0.77 (77%) indica que o modelo classifica corretamente 77% das instâncias. Para um problema de classificação, 77% pode ser considerado um desempenho razoável, mas precisamos olhar as métricas por classe para entender melhor, especialmente se houver desequilíbrio de classes.

#### **Análise do Relatório:**

- **Suporte:** As classes (alto, baixo, moderado) têm um suporte muito similar (cerca de 400 instâncias cada). Isso é excelente, pois indica que o dataset é bem balanceado, o que faz da acurácia geral (77%) uma métrica mais confiável. Se as classes fossem muito desbalanceadas, a acurácia por si só poderia ser enganosa.
- **Precisão (Precision):**
  - **alto: 76%** - Das vezes que o modelo previu "alto", 76% estavam corretas.
  - **baixo: 74%** - Das vezes que o modelo previu "baixo", 74% estavam corretas.
  - **moderado: 81%** - Das vezes que o modelo previu "moderado", 81% estavam corretas. O modelo é mais preciso na previsão da classe moderado.

- **Recall (Revocação):**
  - **alto: 81%** - Dos casos "alto" reais, o modelo identificou 81%.
  - **baixo: 69%** - Dos casos "baixo" reais, o modelo identificou 69%.
  - **moderado: 81%** - Dos casos "moderado" reais, o modelo identificou 81%. O modelo tem um bom recall para "alto" e "moderado", mas tem mais dificuldade em identificar todos os casos reais da classe "baixo". Isso significa que ele está perdendo cerca de 31% dos casos "baixo" reais.
- **F1-Score:**
  - **alto: 0.78**
  - **baixo: 0.71**
  - **moderado: 0.81** O F1-score é a média harmônica de precisão e recall. Ele é uma boa métrica quando se quer um equilíbrio entre os dois. A classe "baixo" tem o F1-score mais baixo, o que confirma que o modelo tem mais dificuldades com essa classe.

**Conclusão Parcial:** O modelo Decision Tree com 77% de acurácia é razoável. Ele performa melhor para as classes "alto" e "moderado", mas a classe "baixo" é o ponto fraco, com menor precisão e recall.

---

## 2. Matriz de Confusão

**O que este gráfico representa:** A matriz de confusão visualiza o desempenho de um algoritmo de classificação, mostrando o número de previsões corretas e incorretas para cada classe.

- **Linhas:** Representam as classes reais.
- **Colunas:** Representam as classes previstas pelo modelo.
- **Diagonal Principal (azul mais escuro):** Representa as previsões corretas.
- **Fora da Diagonal:** Representa os erros de classificação.

**Análise:**

- **Classe alto (primeira linha):**
  - **Corretamente classificado: 326 (81% do recall)**
  - **Confundido com baixo: 43**

- **Confundido com moderado: 34** Isso mostra que, quando o real é "alto", o modelo tende a confundir um número significativo com "baixo" ou "moderado".
- **Classe baixo (segunda linha):**
  - **Corretamente classificado: 274 (69% do recall)**
  - **Confundido com alto: 69**
  - **Confundido com moderado: 54** Este é o ponto fraco, como já identificado pelo recall. O modelo comete muitos erros ao classificar "baixo", confundindo-o bastante com "alto" (quase 70 vezes).
- **Classe moderado (terceira linha):**
  - **Corretamente classificado: 324 (81% do recall)**
  - **Confundido com alto: 38**
  - **Confundido com baixo: 38** Esta classe é a que o modelo lida melhor, com mais previsões corretas e menos confusões relativas.

**Relação com o Relatório:** A matriz de confusão visualiza claramente os números do relatório de classificação, destacando o problema com a classe "baixo" e as confusões específicas que o modelo está fazendo.

---

### **3. Gráfico de Barras: "Precisão por Classe"**

**O que este gráfico representa:** Este gráfico mostra a precisão para cada uma das classes (alto, baixo, moderado). A precisão indica a proporção de previsões positivas que foram realmente corretas para aquela classe.

**Análise:**

- **Confirma visualmente o que vimos no relatório:**
    - **A precisão para moderado é a mais alta (cerca de 0.81).**
    - **A precisão para alto é razoável (cerca de 0.76).**
    - **A precisão para baixo é a mais baixa (cerca de 0.74).**
- 

### **4. Gráfico de Barras: "Distribuição dos Rótulos: Real vs Previsto"**

**O que este gráfico representa:** Este gráfico compara a contagem de cada rótulo na base de dados real com a contagem de cada rótulo prevista pelo modelo. Isso ajuda a identificar se o modelo tem uma tendência a superestimar ou subestimar certas classes em termos de quantidade total prevista.

**Análise:**

- **Real (laranja):** Confirma o balanço quase perfeito das classes no dataset real (cerca de 400 instâncias para cada alto, baixo, moderado). Isso é ideal para treinamento e avaliação.
- **Previsto (verde):**
  - O modelo tende a superestimar a classe alto (prevê mais instâncias "alto" do que realmente existem).
  - O modelo tende a subestimar a classe baixo (prevê menos instâncias "baixo" do que realmente existem).
  - A previsão para a classe moderado está bem alinhada com o real.

**Relação com Análises Anteriores:** Essa tendência de superestimar "alto" e subestimar "baixo" é coerente com a matriz de confusão, onde vimos que muitos casos reais "baixo" foram classificados como "alto", e muitos casos reais "alto" foram classificados corretamente ou confundidos com "moderado", mas não subestimados na previsão total como a classe "baixo". A alta precisão e recall de "moderado" também se refletem na sua boa correspondência entre o real e o previsto.

## **Análise de Sentimentos**

**Análise Detalhada:**

### **1. Acurácia Geral (0.40) e Relatório de Classificação**

**Acurácia Geral (0.40):** Uma acurácia de 40% é muito baixa para um problema de classificação de 3 classes (negativo, neutro, positivo). Em um cenário de classificação aleatória para 3 classes balanceadas, a acurácia esperada seria de aproximadamente 33.3% (1/3). O modelo está apenas marginalmente melhor do que um palpite aleatório.

**Relatório de Classificação:**



- **Suporte:** Todas as classes (negative, neutral, positive) têm suporte de 1068 instâncias. Isso indica que o dataset de teste (e provavelmente o de treinamento) é perfeitamente balanceado. Isso é bom para a avaliação, mas a baixa acurácia não pode ser justificada por desequilíbrio de classes.
- **Precisão, Recall e F1-Score por Classe:**
  - **negative (Negativo):**
    - **Precisão: 0.44** - Das vezes que o modelo previu "negativo", 44% estavam corretas.
    - **Recall: 0.26** - Dos casos "negativo" reais, o modelo identificou apenas 26%. Isso é extremamente baixo, significando que o modelo está perdendo a vasta maioria dos sentimentos negativos reais.
    - **F1-Score: 0.33** - Muito baixo.
  - **neutral (Neutro):**
    - **Precisão: 0.38** - Das vezes que o modelo previu "neutro", 38% estavam corretas.
    - **Recall: 0.59** - Dos casos "neutro" reais, o modelo identificou 59%. Este é o melhor recall entre as classes, mas ainda assim moderado.
    - **F1-Score: 0.46** - É o melhor F1-score, mas ainda baixo.
  - **positive (Positivo):**
    - **Precisão: 0.42** - Das vezes que o modelo previu "positivo", 42% estavam corretas.
    - **Recall: 0.35** - Dos casos "positivo" reais, o modelo identificou 35%. Muito baixo, indicando que o modelo está perdendo uma grande parte dos sentimentos positivos reais.
    - **F1-Score: 0.38** - Baixo.

**Conclusão Parcial:** O desempenho é muito fraco em todas as classes, com a classe neutral sendo a "menos pior". O modelo tem uma capacidade muito limitada de identificar corretamente sentimentos negative e positive (baixo recall) e, quando o faz, a precisão ainda é sub-ótima.

---

## 2. Matriz de Confusão

O que este gráfico representa: Visualiza o número de previsões corretas e incorretas para cada classe.

- Linhas: Classes reais.
- Colunas: Classes previstas.
- Diagonal Principal: Previsões corretas.

### Análise:

- Valores na Diagonal Principal são baixos:
  - negative: 279 (de 1068 reais) - Muito poucos negativos reais foram classificados corretamente.
  - neutral: 625 (de 1068 reais) - Mais neutros reais foram classificados corretamente, o que explica o recall de 0.59.
  - positive: 378 (de 1068 reais) - Muito poucos positivos reais foram classificados corretamente.
- Erros de Classificação são muito altos:
  - Tendência negative (linha 1): Quando o sentimento é realmente negativo, o modelo classifica erroneamente 500 como neutral e 289 como positive. Isso é problemático, pois está classificando sentimentos negativos como neutros ou, pior, positivos.
  - Tendência neutral (linha 2): Quando o sentimento é realmente neutro, o modelo confunde bastante com negative (223) e positive (220).
  - Tendência positive (linha 3): Quando o sentimento é realmente positivo, o modelo classifica erroneamente 474 como neutral e 216 como negative. Classificar positivo como negativo é um erro grave.
- O modelo está confundindo as classes em larga escala. Há um grande número de classificações cruzadas.

Relação com o Relatório: A matriz de confusão visualiza claramente o motivo da baixa acurácia e dos baixos F1-scores. O modelo tem uma dificuldade enorme em distinguir entre as classes, especialmente entre "negativo" e "positivo", e frequentemente as confunde com "neutro".

---

### 3. Gráfico de Barras: "Métricas por Classe (Precisão, Revocação e F1-Score)"

O que este gráfico representa: Uma visualização em barras das métricas Precision, Recall e F1-Score para cada classe.

Análise:

- Confirma visualmente a baixa performance geral e a fraqueza do recall para negative e positive.
  - A barra de "Recall" para neutral é visivelmente mais alta, enquanto para negative e positive é bem baixa.
  - Os F1-Scores são todos baixos, o que é um reflexo direto do baixo recall e precisão.
- 

### 4. Gráficos de Palavras Mais Comuns por Sentimento (negative, neutral, positive)

O que estes gráficos representam: Eles mostram as 10 palavras mais frequentes em cada uma das classes de sentimento, após o pré-processamento (remoção de stopwords e stemming). Isso pode dar insights sobre o vocabulário associado a cada sentimento e se o pré-processamento está funcionando como esperado.

Análise:

- Geral: As palavras parecem ser comuns e genéricas ("ru", "públic", "serviç", "cidad", "melhor", "problem"). O stemming está ativo, transformando palavras como "rua", "ruas" em "ru".
- Contexto: As palavras são bastante genéricas e podem aparecer em qualquer tipo de sentimento (ex: "serviç" pode ser "bom serviço" ou "mau serviço").
- Problema Potencial: Se as palavras mais comuns não são distintivas o suficiente para cada sentimento, isso dificulta a tarefa do classificador. Por exemplo, "problem" pode levar a "negative", mas se "problem" aparece em contextos neutros ou até mesmo em discussões sobre soluções ("resolvendo o problema"), pode criar ruído.
- Ausência de Palavras Fortes: Não vemos palavras com carga emocional muito forte que seriam esperadas para distinguir claramente entre positivo e negativo (ex: "excelente", "péssimo", "amor", "ódio"). Isso pode ser devido a:

- **Natureza dos Dados:** Os textos coletados podem ser muito factuais ou neutros, dificultando a distinção de sentimentos.
  - **Pré-processamento:** O stemming ou a remoção de stopwords pode estar removendo palavras-chave importantes ou nuances.
  - **TF-IDF:** Embora o TF-IDF ajude a ponderar termos, se a frequência de termos-chave é baixa ou se termos neutros dominam, o vetor resultante pode não ser discriminatório.
- 

#### **Parâmetros e Configurações do Modelo:**

- **Vetorizador TF-IDF:**
    - **ngram\_range: [0:1, 1:2]** - Se isso significa unigramas e bigramas, é um bom ponto de partida, mas talvez explorar (1,3) para trigramas possa ajudar se frases curtas forem importantes.
    - **max\_df: 0.85, min\_df: 2** - Bons para remover termos muito frequentes e muito raros.
    - **sublinear\_tf: true** - Ajuda a suavizar a contagem de termos, o que é geralmente benéfico.
  - **Classificador: LogisticRegression(max\_iter=1000)** - Regressão Logística é um classificador linear simples, mas eficaz, e é um bom ponto de partida. **max\_iter=1000** garante convergência para datasets maiores.
- 

#### **Conclusão Geral e Próximos Passos Sugeridos:**

O modelo de Análise de Sentimentos, com uma acurácia de 40%, está performando muito mal. Ele está apenas um pouco melhor do que um chute aleatório. O problema principal é a incapacidade do modelo de distinguir efetivamente entre as classes, especialmente entre negative e positive, e a forte confusão com neutral.

#### **Recomendações Urgentes para Melhoria:**

1. **Revisão do Dataset e Rotulagem:**
  - **Qualidade dos Dados:** Os dados de sentimento são realmente adequados? Existem muitos textos ambíguos ou que são difíceis até para humanos rotularem?

- **Balanceamento:** Embora o suporte seja balanceado, talvez a dificuldade esteja na sobreposição semântica entre as classes nos textos.
- **Revisão da Rotulagem:** A rotulagem foi feita de forma consistente e clara?

## **2. Engenharia de Features/Pré-processamento:**

- **N-gramas:** Experimente `ngram_range=(1,3)` para capturar trigramas, que podem conter mais nuances de sentimento (ex: "muito bom", "nada feliz").
- **Análise de Sentimentos Específicos:** Considere o uso de léxicos de sentimento (dicionários de palavras com pontuações de sentimento) ou modelos pré-treinados para extração de features de sentimento que possam complementar o TF-IDF.
- **Processamento de Negação:** Implementar regras para lidar com negações (ex: "não bom" deve ser tratado como "ruim").
- **Tratamento de Pontuação/Emojis:** Emojis e pontuação podem ser fortes indicadores de sentimento e devem ser considerados (talvez não removê-los ou mapeá-los para tokens).
- **Stemming vs. Lemmatização:** `RSLPStemmer` pode ser muito agressivo e remover nuances. Considere um lematizador se houver um disponível para português (reduz palavras à sua forma base léxica, ex: "correndo" -> "correr").

## **3. Experimentar Outros Modelos de Classificação:**

- **Modelos mais robustos para texto:**
  - **Multinomial Naive Bayes (MultinomialNB):** Frequentemente um baseline muito forte para classificação de texto com TF-IDF.
  - **Support Vector Machines (SVM) com kernel linear:** `LinearSVC` ou `SVC(kernel='linear')` costumam ter ótimo desempenho com dados TF-IDF.
  - **Ensembles: Random Forest ou Gradient Boosting (XGBoost, LightGBM)** podem ser experimentados, embora modelos lineares como Logistic Regression e SVMs sejam frequentemente muito competitivos para texto vetorizado com TF-IDF.

- **Redes Neurais (Modelos de Embedding):** Para um desempenho superior em análise de sentimentos, considere usar modelos de embedding de palavras (Word2Vec, GloVe, FastText) combinados com redes neurais (RNNs, LSTMs, Transformers). Isso, no entanto, é um nível de complexidade muito maior.
- 4. Balanceamento Sintético (SMOTE):** Você mencionou SMOTE nas importações do script inicial. Se o problema for que o modelo não consegue aprender as fronteiras das classes negative e positive (mesmo que balanceadas, talvez elas sejam semanticamente difíceis de separar), aplicar SMOTE nas etapas de treinamento *antes* de treinar o classificador pode ajudar o modelo a "ver" mais exemplos dessas classes e aprender melhor suas características. Embora seus dados pareçam balanceados agora, SMOTE pode ajudar o modelo a diferenciar melhor.

### **Sistema de Recomendação**

#### **Conclusão Final sobre o Sistema de Recomendação:**

Baseado nas informações fornecidas, seu sistema de recomendação baseado em conteúdo parece estar funcionando muito bem para a tarefa proposta de encontrar itens similares.