# Detecção de Mudanças e Classificação de Uso do Solo

## 1 Visão Geral

A análise de mudanças em dados de sensoriamento remoto multi-imagem e multi-data nos ajuda a descobrir e entender condições globais. Este desafio utiliza características geográficas derivadas de imagens de satélite. Os dados foram processados usando técnicas de visão computacional e estão prontos para exploração usando métodos de aprendizado de máquina.

O objetivo deste desafio é classificar uma área geográfica dada em seis categorias.

## 2 Classes do Desafio

O objetivo é classificar cada polígono em uma das seguintes 6 classes:

- **Demolition** (Demolição): 0
- **Road** (Estrada): 1
- **Residential** (Residencial): 2
- **Commercial** (Comercial): 3
- **Industrial** (Industrial): 4
- **Mega Projects** (Mega Projetos): 5

## 3 Descrição dos Dados

Os dados de treinamento e teste estão nos arquivos `train.geojson` e `test.geojson`, respectivamente.

### 3.1 Características (Features)

As características geográficas incluem:

1. Um polígono irregular (`geometry`).
2. Valores categóricos descrevendo o status do polígono em cinco datas diferentes (ex: em construção no dia 0, completado nos dias seguintes).
3. Características urbanas da vizinhança (ex: região urbana densa, industrial).
4. Características geográficas da vizinhança (ex: perto de um rio ou colina).

### 3.2 Colunas do Dataset

As colunas disponíveis nos arquivos geojson são:

- `date0` até `date4`: Datas de observação (DD-MM-YYYY).
- `change_status_date0` até `change_status_date4`: Status do polígono em cada data.
- `urban_type`: Valores separados por vírgula mostrando tipos urbanos da vizinhança.
- `geography_type`: Valores separados por vírgula mostrando tipos geográficos da vizinhança.
- `geometry`: Representação vetorial dos polígonos.
- `change_type`: Rótulo a ser classificado (apenas no treino).

Além disso, para cada data, são fornecidas estatísticas de cor de imagens de satélite de 50cm:

- Médias: `img_red_mean_date1` até `img_red_mean_date5`, etc.
- Desvios Padrão: `img_red_std_date1` até `img_red_std_date5`, etc.

**Nota Importante:** Observe que as colunas de estatísticas de imagem usam sufixos de `date1` a `date5`, enquanto as colunas de data e status usam de `date0` a `date4`.

# 4 Pipeline Sugerido

O pipeline proposto segue a estrutura vista no curso:

1. **Pré-processamento de Dados**: Converter os dados para o formato apropriado.
2. **Engenharia de Features e Redução de Dimensionalidade**:
   - Explorar One-Hot Encoding para tipos urbanos e geográficos.
   - Criar features geométricas (área, perímetro) a partir dos polígonos.
   - Calcular intervalo de dias entre as datas consecutivas.
   - Seleção de features ou redução de dimensionalidade pode ser benéfica.
3. **Algoritmo de Aprendizado**:
   - Um baseline simples (k-NN) é fornecido em `skeleton_code.py` ($\approx$ 40% de performance). O código utiliza apenas a **área do polígono** como feature.
   - Testar outros classificadores: Regressão Logística, SVM, Árvores de Decisão, Redes Neurais, Ensemble Learning.
4. **Avaliação**: A métrica de avaliação é o **Mean F1-Score**.

# 5 Submissão e Avaliação

O trabalho deve ser feito em grupos de 3-4 alunos.

## 5.1 Entregáveis

1. **Submissão no Kaggle** (40 pontos):
   - Comando para baixar dados:
     `kaggle competitions download -c 2-el-1730-machine-learning-project-2026`
   - **Passo a Passo para Submissão:**
     (a) Executar o modelo treinado nos dados de teste (`test.geojson`).
     (b) Obter os rótulos de classe preditos para cada instância.
     (c) Criar um arquivo `sample_submission.csv` contendo o cabeçalho e duas colunas: `Id` e `change_type`.
     (d) Criar uma conta no Kaggle (uma por time) e realizar a submissão fazendo upload deste arquivo.
   - O Kaggle avaliará automaticamente:
     – Placar público (durante competição): baseado em $\approx$ 30% dos dados de teste.
     – Placar final (Private Leaderboard): baseado nos 70% restantes.
   - Limite diário: 10 submissões.
2. **Relatório no Edunao** (50 pontos):
   - Arquivo PDF apenas.
   - Nome do arquivo: `nomedotime_aluno1_aluno2_aluno3.pdf`.
   - Deve conter nomes completos e nome do time no Kaggle.
   - **Seção 1: Feature Engineering**: Motivação, experimentos, combinações testadas.
   - **Seção 2: Model Tuning e Comparação**: Comparar múltiplos classificadores, ajuste de hiperparâmetros, validação cruzada, discussão de modelos descartados.
3. **Código no Edunao** (10 pontos - Avaliação Geral):
   - Arquivo ZIP: `nome_do_seu_time.zip` (máx 512MB).
   - Código reproduzível.
   - Os 10 pontos também cobrem respeito às diretrizes e clareza.

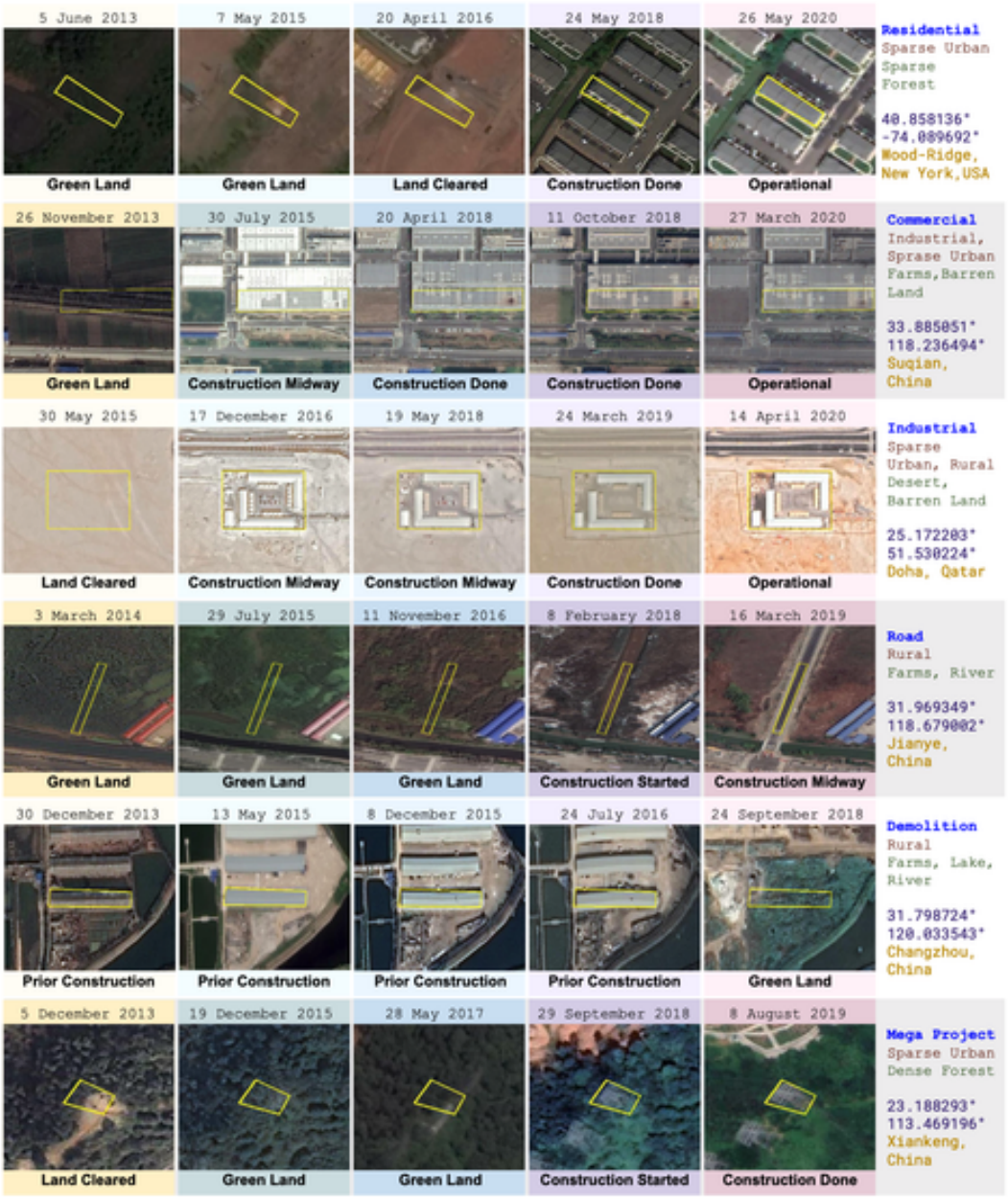


Figure 1. Samples from 'test.geojson' showing different **change type**, **change status** on `different dates`, neighborhood label(s), and geography label(s). Latitude-longitude of the change polygon is shown along with city name. First row shows construction of a residential property in suburban area of New York, USA. Second row shows a commercial building in an industrial region which used to be farm lands of a fast growing second tier city in China. Third row shows an industrial construction in desert of Doha, Qatar which went from rural barren desert to a sparse urban area in a time period of 5 years. Fourth row shows construction of a road crossing a river in farm lands of rural China. Fifth row shows special case of urban change, demolition of a farm storage in the fast growing city Changzhou in China. Last row shows construction of a power grid unit which comes under mega project type.

Figura 1: Amostras do 'test.geojson' mostrando diferentes tipos de mudanças, status, datas, e características urbanas/geográficas.

## 6 Referências

O conjunto de dados faz parte do paper *"QFabric: Multi-Task Change Detection Dataset"* (CVPR 2021W) de Sagar Verma et al.