🚀 Quick Start Guide - Wildfire Analysis Pipeline

Step 1: Install (2 minutes)

```bash
# Create and activate virtual environment
python -m venv wildfire-env
source wildfire-env/bin/activate  # Windows: wildfire-env\Scripts\activate

# Install all dependencies
pip install -r requirements.txt
```

Step 2: Verify Setup (1 minute)

```bash
# Run automated tests
python test_script.py
```

Expected output:

```
✓ Package Imports      - PASSED
✓ Data Files           - PASSED
✓ Data Extraction      - PASSED
✓ Transformer Model    - PASSED
✓ Training Pipeline    - PASSED
✓ Full Pipeline        - PASSED

🎉 All tests passed!
```

Step 3: Run Pipeline (5-10 minutes)

Option A: Full Pipeline (Transformer)

```bash
python main_pipeline.py --step full --model transformer
```

Option B: Quick Test (Random Forest)

```bash
```

```
python main_pipeline.py --step full --model random_forest
```

Step 4: Check Results

bash

```
# View generated files
ls outputs/

# Should see:
# - wildfire_features.csv       (extracted features)
# - wildfire_predictions.csv    (model predictions)
# - transformer_model.pth       (trained model)
```

Step 5: Analyze Results

python

```
import pandas as pd

# Load predictions
df = pd.read_csv('outputs/wildfire_predictions.csv')

# Check accuracy
accuracy = (df['confidence_num'] == df['predicted_confidence']).mean()
print(f"Model Accuracy: {accuracy:.1%}")

# View sample predictions
print(df[['latitude', 'longitude', 'frp', 'confidence', 'predicted_confidence']].head())
```

Common Commands

bash

```
# Extract features only
python main_pipeline.py --step extract

# Train model only (uses cached features)
python main_pipeline.py --step train --model transformer

# Generate predictions only
python main_pipeline.py --step predict

# Force re-extract data
python main_pipeline.py --step full --force-reextract
```

Project Files Overview

File	Purpose
`data_extraction.py`	Extract features from VIIRS, ERA5, and DEM data
`transformer_model.py`	Transformer model implementation
`main_pipeline.py`	Main execution pipeline
`config_module.py`	Configuration settings
`test_script.py`	Automated testing

Output Files

File	Contents
`wildfire_features.csv`	Extracted features (lat, lon, frp, wind, elevation, slope)
`wildfire_predictions.csv`	Predictions added to features
`transformer_model.pth`	Trained model checkpoint

Quick Customization

Edit `config_module.py`:

```
python
```

```
# Change geographic area
bounds.lat_min = 33.5
bounds.lat_max = 34.5
bounds.lon_min = -119.0
bounds.lon_max = -118.0

# Adjust model size
model.d_model = 256      # Larger = more capacity
model.num_layers = 6     # More layers = deeper network
model.nhead = 16         # More heads = more attention

# Change training
model.test_size = 0.3    # 30% validation data
```

Troubleshooting

"Module not found"

bash

```
pip install -r requirements.txt
```

"File not found"

- Ensure data files are in project directory
- Check filenames match exactly

"Out of memory"

- Reduce batch size in config
- Use smaller model dimensions
- Process fewer samples

"CUDA error"

- Model auto-uses CPU if no GPU
- No action needed

What's Next?

1. **Experiment**: Try different hyperparameters
2. **Visualize**: Create interactive maps (see `blueprint.md`)

3. **Deploy**: Build web API (see Phase 3 in blueprint)

Getting Help

- Check `test_pipeline.log` for detailed errors
- Review README.md for full documentation
- See `blueprint.md` for website development

Total setup time: ~10 minutes
First successful prediction: ~15 minutes
Production-ready model: ~1 hour

Let's build something great! 🔥🌲