Drought Condition Prediction Model Analysis

Introduction

This project develops a machine learning pipeline for six-class drought severity classification (scores 0–5) using daily meteorological and land-surface features. We combine insights from preliminary baselines (AdaBoost, Decision Tree, Linear Regression) with a robust preprocessing workflow (outlier removal, feature selection via RFE), resampling (SMOTE, NearMiss, Cleaning Rule), and model training (Decision Tree, KNN, Random Forest). Finally, we explore dimensionality reduction (PCA, LDA) and construct a soft-voting ensemble. The best single Random Forest achieved **85.1%** accuracy; the ensemble yielded **78.9%** accuracy with improved stability.

Previous Work & Baselines

- **Initial Experiments:** AdaBoost, Decision Tree, and Linear Regression on raw data.
- **Baseline Accuracy:** Up to **71%** with minimal preprocessing.
- **Lessons Learned:** Feature scaling and outlier handling are critical before advanced modeling.

Dataset Analysis

Data set used-https: //www.kaggle.com/datasets/cdminix/us-drought-meteorological-data/data

Class Distribution

The dataset shows a significant class imbalance:

- Class 0: 1,652,230 instances (no drought)
- Class 1: 466,944 instances
- Class 2: 295,331 instances
- Class 3: 196,802 instances
- Class 4: 106,265 instances
- Class 5: 39,224 instances (severe drought)

Features

The dataset contains multiple meteorological measurements including:

- PRECTOT: Precipitation total
- PS: Pressure at surface
- QV2M: Specific humidity at 2m
- T2M: Temperature at 2m
- T2MDEW: Dew point temperature at 2m
- T2MWET: Wet bulb temperature at 2m
- T2M_MAX: Maximum temperature at 2m
- T2M_MIN: Minimum temperature at 2m
- T2M_RANGE: Temperature range at 2m
- TS: Surface temperature
- WS10M: Wind speed at 10m
- WS10M_MAX: Maximum wind speed at 10m
- WS10M_MIN: Minimum wind speed at 10m
- WS10M_RANGE: Wind speed range at 10m
- WS50M: Wind speed at 50m
- WS50M_MAX: Maximum wind speed at 50m
- WS50M_MIN: Minimum wind speed at 50m
- WS50M_RANGE: Wind speed range at 50m

Additionally, temporal features were extracted from the date:

- Year
- Month
- Day

Data Preprocessing

Data Cleaning

- Removed NaN values
- Parsed date into year, month, and day components
- Removed outliers using the 3σ (three standard deviation) rule for all continuous features

Feature Scaling

- Applied StandardScaler for normalization of features

Handling Imbalanced Data

Multiple approaches were tested:

1. **SMOTE** (Synthetic Minority Over-sampling Technique): Used to upsample minority classes
2. **Neighbourhood Cleaning**: Reduced noisy majority samples
3. **NearMiss**: Downsampled majority classes by proximity

Feature Selection

Used RFE (Recursive Feature Elimination) with RandomForestClassifier to select the top 15 features.

The following columns were dropped based on feature importance analysis:

- PRECTOT
- T2MWET
- WS10M_MAX
- WS10M_MIN
- WS50M_MIN
- month

Feature Engineering

Two dimensionality reduction techniques were applied:

1. **PCA** (Principal Component Analysis): 5 components were selected that explained over 90% of variance
2. **LDA** (Linear Discriminant Analysis): 5 components were used

Machine Learning Models

Multiple classification models were developed and evaluated:

1. Decision Tree Classifier

- Tested with various resampling techniques and dimensionality reduction
- Hyperparameter tuning using GridSearchCV for optimal max_depth and max_features

2. K-Nearest Neighbors (KNN)

- Tested with and without SMOTE upsampling
- Hyperparameter tuning for finding optimal n_neighbors

3. Random Forest Classifier

- Hyperparameter tuning for n_estimators, max_depth, bootstrap, and max_features
- Best performing base model with accuracy of 85.1%

Hyperparameter tuning details

Model	**Hyperparameters**	**CV**
Decision Tree (DT)	max_depth, min_samples_leaf, max_features	4-fold
KNN	n_neighbors (1–9), Euclidean	3-fold
Random Forest (RF)	n_estimators, max_depth, max_features, bootstrap	3-fold

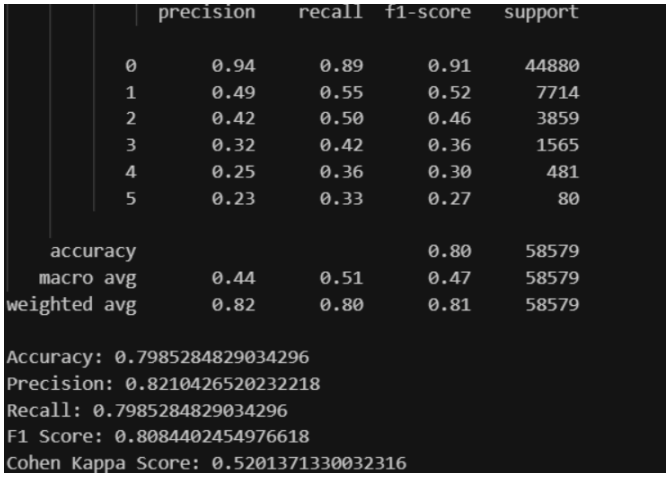
4. Bagging Ensemble

- Combined multiple trained models using VotingClassifier
- Used soft voting strategies
- Achieved 78.9% accuracy with better balanced metrics

Model Performance Analysis

Performance Metrics

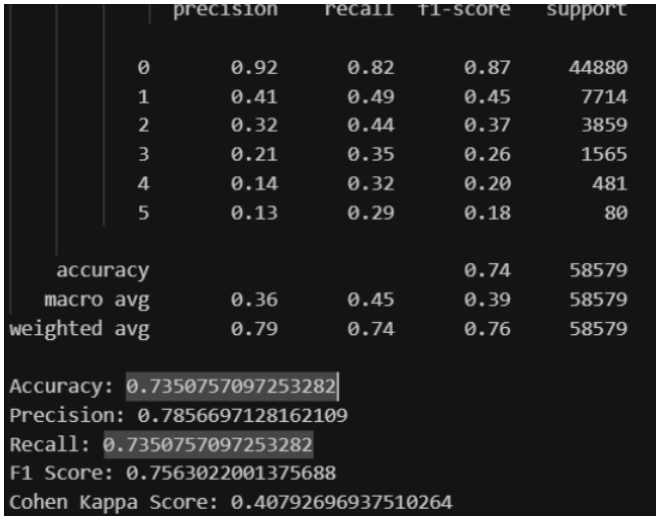
- Decision Tree using SMOTE upsampling



	precision	recall	f1-score	support
0	0.94	0.89	0.91	44880
1	0.49	0.55	0.52	7714
2	0.42	0.50	0.46	3859
3	0.32	0.42	0.36	1565
4	0.25	0.36	0.30	481
5	0.23	0.33	0.27	80
accuracy			0.80	58579
macro avg	0.44	0.51	0.47	58579
weighted avg	0.82	0.80	0.81	58579

Accuracy: 0.7985284829034296
Precision: 0.8210426520232218
Recall: 0.7985284829034296
F1 Score: 0.8084402454976618
Cohen Kappa Score: 0.5201371330032316

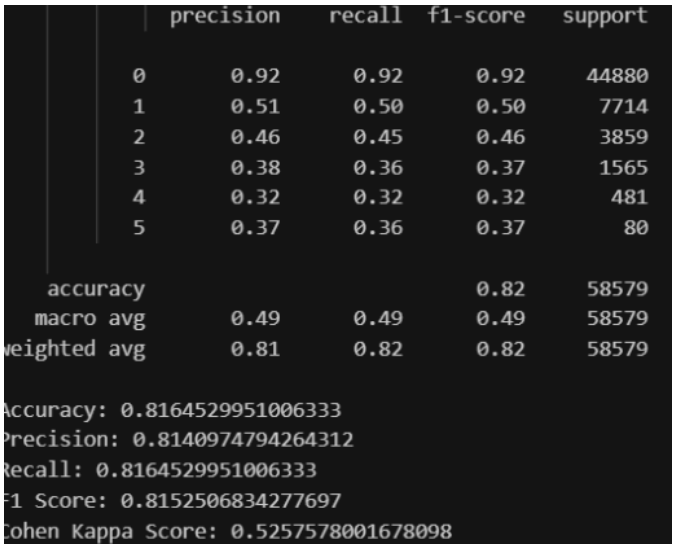
- Decision Tree using SMOTE upsampling and PCA



	precision	recall	f1-score	support
0	0.92	0.82	0.87	44880
1	0.41	0.49	0.45	7714
2	0.32	0.44	0.37	3859
3	0.21	0.35	0.26	1565
4	0.14	0.32	0.20	481
5	0.13	0.29	0.18	80
accuracy			0.74	58579
macro avg	0.36	0.45	0.39	58579
weighted avg	0.79	0.74	0.76	58579

```
Accuracy: 0.7350757097253282
Precision: 0.7856697128162109
Recall: 0.7350757097253282
F1 Score: 0.7563022001375688
Cohen Kappa Score: 0.40792696937510264
```

- Decision Tree using SMOTE upsampling and LDA



	precision	recall	f1-score	support
0	0.92	0.92	0.92	44880
1	0.51	0.50	0.50	7714
2	0.46	0.45	0.46	3859
3	0.38	0.36	0.37	1565
4	0.32	0.32	0.32	481
5	0.37	0.36	0.37	80
accuracy			0.82	58579
macro avg	0.49	0.49	0.49	58579
weighted avg	0.81	0.82	0.82	58579

```
Accuracy: 0.8164529951006333
Precision: 0.8140974794264312
Recall: 0.8164529951006333
F1 Score: 0.8152506834277697
Cohen Kappa Score: 0.5257578001678098
```

- KNN algorithm without resampling

	precision	recall	f1-score	support
0	0.90	0.96	0.93	44880
1	0.56	0.47	0.51	7714
2	0.54	0.41	0.47	3859
3	0.50	0.32	0.39	1565
4	0.47	0.27	0.35	481
5	0.39	0.28	0.32	80
accuracy			0.83	58579
macro avg	0.56	0.45	0.49	58579
weighted avg	0.82	0.83	0.82	58579

```
Accuracy: 0.8347701394697759
Precision: 0.8161182137406356
Recall: 0.8347701394697759
F1 Score: 0.8227848915625144
Cohen Kappa Score: 0.5349195661912078
```

- KNN algorithm without resampling after Hyperparameter tuning

	precision	recall	f1-score	support
0	0.94	0.94	0.94	44880
1	0.56	0.56	0.56	7714
2	0.53	0.51	0.52	3859
3	0.44	0.42	0.43	1565
4	0.42	0.38	0.40	481
5	0.35	0.42	0.38	80
accuracy			0.84	58579
macro avg	0.54	0.54	0.54	58579
weighted avg	0.84	0.84	0.84	58579

```
Accuracy: 0.8425715700165588
Precision: 0.8408310725362154
Recall: 0.8425715700165588
F1 Score: 0.8416706778235012
Cohen Kappa Score: 0.5939604507252405
```

- KNN algorithm with SMOTE resampling after Hyperparameter tuning

	precision	recall	f1-score	support
0	0.95	0.92	0.93	44880
1	0.54	0.59	0.56	7714
2	0.50	0.53	0.52	3859
3	0.42	0.47	0.44	1565
4	0.35	0.41	0.38	481
5	0.32	0.45	0.37	80
accuracy			0.84	58579
macro avg	0.51	0.56	0.53	58579
weighted avg	0.84	0.84	0.84	58579

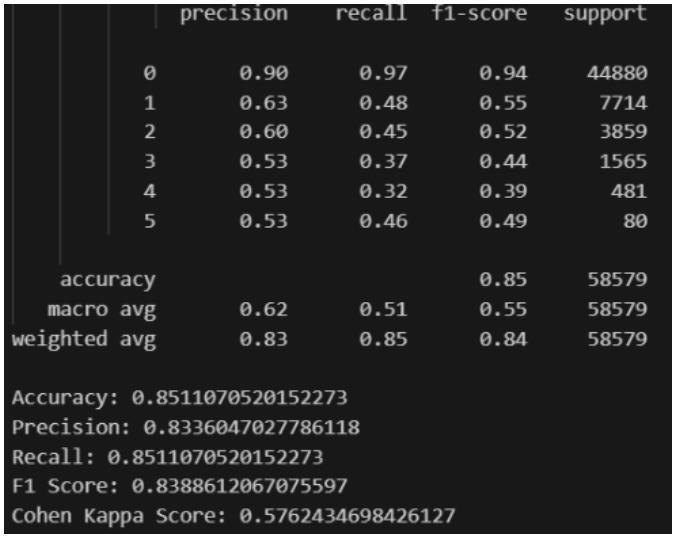
```
Accuracy: 0.8351115587497226
Precision: 0.8449433708554838
Recall: 0.8351115587497226
F1 Score: 0.8396162165525559
Cohen Kappa Score: 0.5931042765926839
```

- Random Forest without resampling

	precision	recall	f1-score	support
0	0.90	0.97	0.94	44880
1	0.63	0.48	0.55	7714
2	0.60	0.45	0.52	3859
3	0.53	0.37	0.44	1565
4	0.53	0.32	0.39	481
5	0.53	0.46	0.49	80
accuracy			0.85	58579
macro avg	0.62	0.51	0.55	58579
weighted avg	0.83	0.85	0.84	58579

```
Accuracy: 0.8511070520152273
Precision: 0.8336047027786118
Recall: 0.8511070520152273
F1 Score: 0.8388612067075597
Cohen Kappa Score: 0.5762434698426127
```

- Random Forest with resampling and hyperparameter tuning



	precision	recall	f1-score	support
0	0.90	0.97	0.94	44880
1	0.63	0.48	0.55	7714
2	0.60	0.45	0.52	3859
3	0.53	0.37	0.44	1565
4	0.53	0.32	0.39	481
5	0.53	0.46	0.49	80
accuracy			0.85	58579
macro avg	0.62	0.51	0.55	58579
weighted avg	0.83	0.85	0.84	58579

Accuracy: 0.8511070520152273
Precision: 0.8336047027786118
Recall: 0.8511070520152273
F1 Score: 0.8388612067075597
Cohen Kappa Score: 0.5762434698426127

Key Observations

1. **Resampling Impact**:
    - SMOTE upsampling slightly reduced accuracy but significantly improved other metrics
    - Near Miss downsampling and Neighbourhood cleaning produced poor results compared to SMOTE
2. **Dimensionality Reduction Impact**:
    - Models with dimensionality reduction (PCA, LDA) underperformed by 5-8% in accuracy compared to models without dimensionality reduction
3. **Model Comparison**:
    - **Lowest Performer**: Decision Tree with Near Miss Downsampling and LDA
        - Accuracy: 12.7%
        - Precision: 0.51
        - Recall: 0.18
        - F1 Score: 0.22

        - Cohen's Kappa: 0.05
    - **Best Performer**: Random Forest (after hyperparameter tuning)
        - Accuracy: 85.1%
        - Precision: 0.83
        - Recall: 0.85
        - F1 Score: 0.83
        - Cohen's Kappa: 0.57
    - **Bagging Ensemble**:
        - Accuracy: 78.9%
        - Precision: 0.78
        - Recall: 0.78
        - F1 Score: 0.78
        - Cohen's Kappa: 0.63

Analysis of Bagging Ensemble

The bagging ensemble combines multiple models:

- Decision Tree classifiers with various preprocessing techniques
- KNN classifiers
- Random Forest classifier

While the Random Forest model achieved higher accuracy (85.1%), the bagging ensemble showed better performance in terms of Cohen's Kappa (0.63 vs 0.57), indicating better agreement between predicted and actual classes beyond random chance. This suggests the ensemble might be more reliable across all classes despite the slightly lower overall accuracy.

Implementation Details

The implementation involved:

1. Data preprocessing pipeline in `data.py`
2. Ensemble model implementation in `bagging_1.py`
3. Individual model training and evaluation
4. Model persistence using pickle

The `BaggingEnsemble` class provides functionality to:

- Load pre-trained models from pickle files
- Build a voting ensemble model
- Make predictions using the ensemble
- Evaluate performance using multiple metrics

Future Work

The project proposes several directions for future development:

1. **Deep Learning Models**:
    - Implement LSTM and Transformer-based models for spatio-temporal sequence prediction of drought severity
2. **Predictive Early Warning System**:
    - Build an automated pipeline to alert authorities based on forecasted drought severity levels
3. **Mobile and Web App Integration**:
    - Deploy an interactive app for localized drought prediction
    - Implement real-time farmer input and geotagged reporting capabilities

Conclusion

The project successfully developed machine learning models for drought prediction with a significant improvement over previous work (from 71% to 85.1% accuracy). The Random Forest model showed the best overall accuracy, while the bagging ensemble provided better balanced performance across all classes as indicated by the higher Cohen's Kappa score.

The analysis highlights the importance of proper handling of imbalanced data and careful feature selection. The project demonstrates that ensemble methods can provide more robust predictions for drought conditions, which could be valuable for agricultural planning and disaster management.