Hands-on Data Science

WEATHER TYPE CLASSIFICATION

YUCHENG MA

School of Computer Science

12/12/2024

https://github.com/markfromcd/Weather-Type-Classification.git

1. RECAP

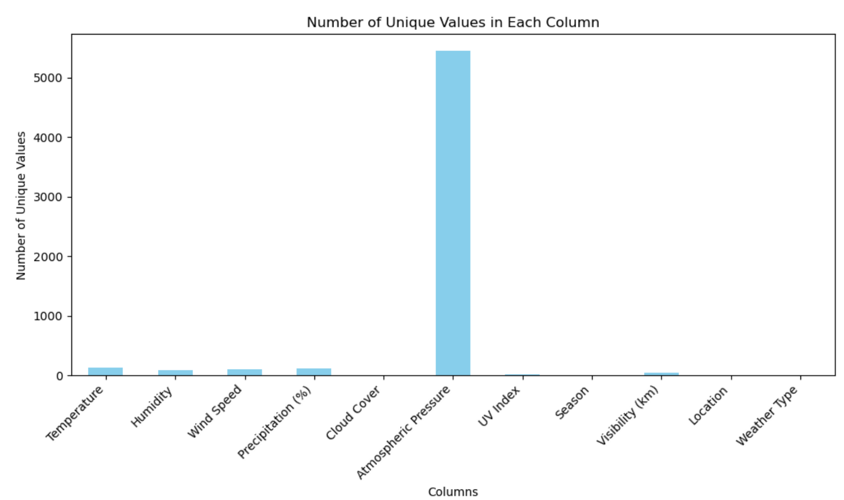
INTRO: A CLASSIFICATION PROBLEM

The problem: Given a set of weather-related features such as temperature, humidity, wind speed, precipitation percentage, cloud cover, atmospheric pressure, UV index, season, visibility, and location, the objective is to predict the weather type (e.g., Rainy, Cloudy, Sunny, Snowy).

Why important: Weather predictions **impact a wide range of industries and everyday life decisions**. Like: Real-world applicability, safety and disaster preparedness, environmental protection. In details: agriculture, transportation, retail, early warnings...

Source: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification/data

EDA. CHECK THE NUNIQUE VALUE OF EACH COLUMN



Only atmospheric pressure has a lot values, but it really plays a vital role in weather prediction, so I **will not drop it.**

PREPROCESSING

- StandardScaler: to scale each continuous features.
- OneHotEncoder: **Location**, **allowing models to interpret categorical features correctly.**
- OrdinalEncoder: **Season and Cloud Cover** since they may contain inherent orders.
- Since I used XGBoost, I translated target variables using label encoding.
- No missing value at all

2. CROSS VALIDATION

HOW I SPLIT THE DATA

Dataset → Stratified Split (80% Train(and Valid), 20% Test) → Kfold (4, each fold be valid set once) in Model Training & Evaluation

Stratification enusures Consistent Class Proportions Across Splits

CV PIPLINE

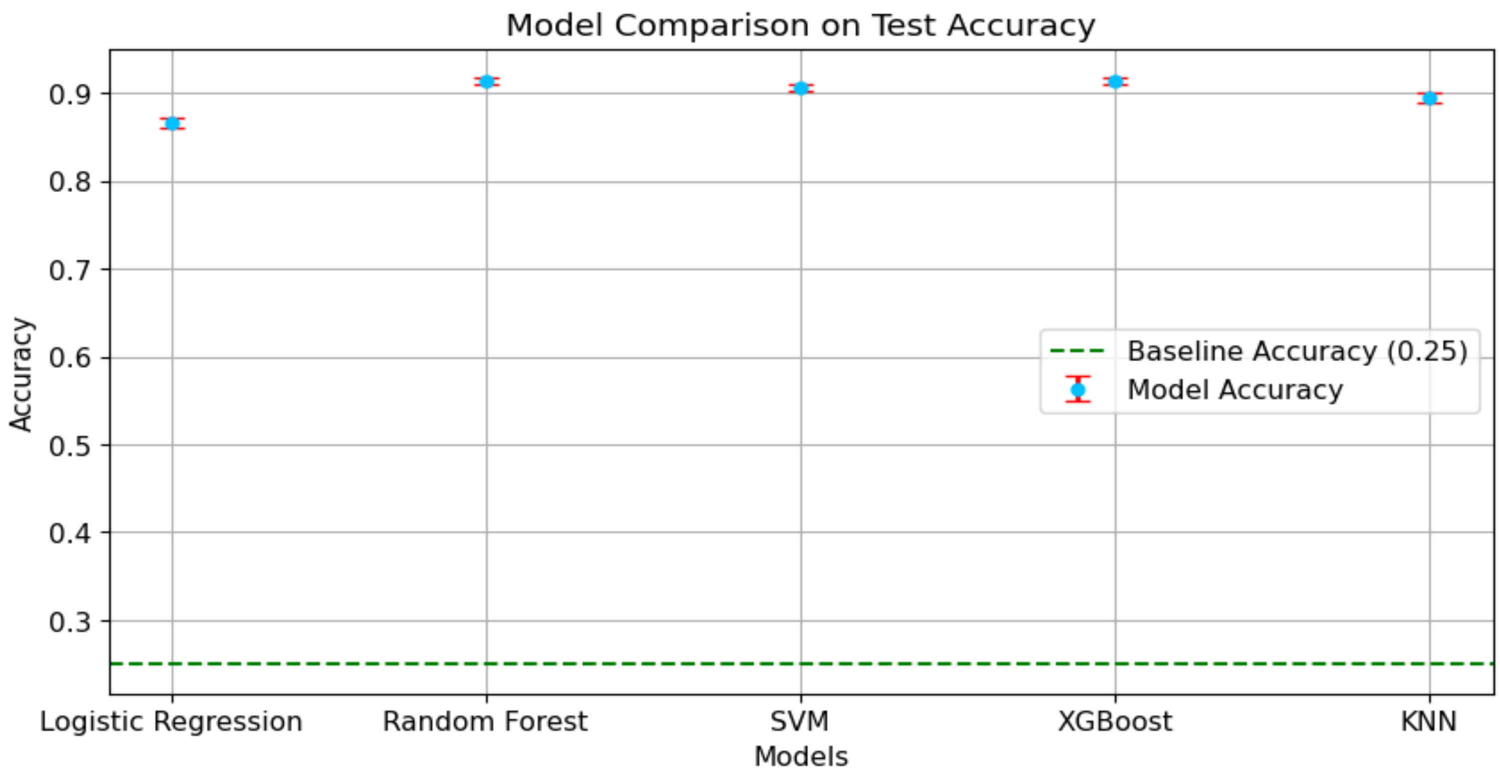
1. **Stratified K-Fold Cross-Validation**
2. **Preprocessing**
3. **Random State Variation:**
   - Used to repeat the cross-validation process for multiple random states (e.g., 10 runs).
   - Reduces bias caused by a single random split.
4. **Params Tuning (GridSearchCV):**
   - Explores combinations of hyperparameters for each model.
   - Evaluates each combination using stratified K-fold CV on the training set.
   - **Scoring Metric**: Accuracy for consistent evaluation across models.
5. **Store Scores**
   - Test Scores: Stores test accuracy for each random state.
   - Best Models: Retains the best model for each random state.
6. **Best Model Selection:**
   - Chooses the model and hyperparameters that yield the highest mean test scores and std.
7. **Final Model**

ALGORITHMS AND PARAMS

Algorithms	**Parameters**	**Best Combinations**
Logistic Regression	C = [0.01, 0.1, 1, 10, 100] penalty = ['l2'] solver = ['lbfgs', 'sag', 'newton-cg']	C = 0.1 Penalty = l2 Solver = lbfgs
Random Forest	max_depth = [5, 10, None] max_features = ['sqrt', 'log2', None] min_samples_split = [2, 5, 10]	max_depth = None Max_features = sqrt Min_sample_split = 10
SVM	C = [0.01, 0.1, 1, 10] gamma = [0.001, 0.01, 0.1, 1, 10, 100] kernel = ['rbf', 'sigmoid']	C = 1 gamma = 0.1 Kernel = rbf
XGBoost	learning_rate = [0.01, 0.1, 0.3] reg_lambda = [0e0, 1e-2, 1e-1, 1e0, 1e1, 1e2] max_depth = [3, 6, 9]	learning_rate = 0.3 reg_lambda = 0.01 Max_depth = 9
KNN	n_neighbors = [3, 5, 7, 10] weights = ['uniform', 'distance'] metric = ['euclidean', 'manhattan', 'minkowski']	n_neighbors = 7 weights = uniform metric = manhattan

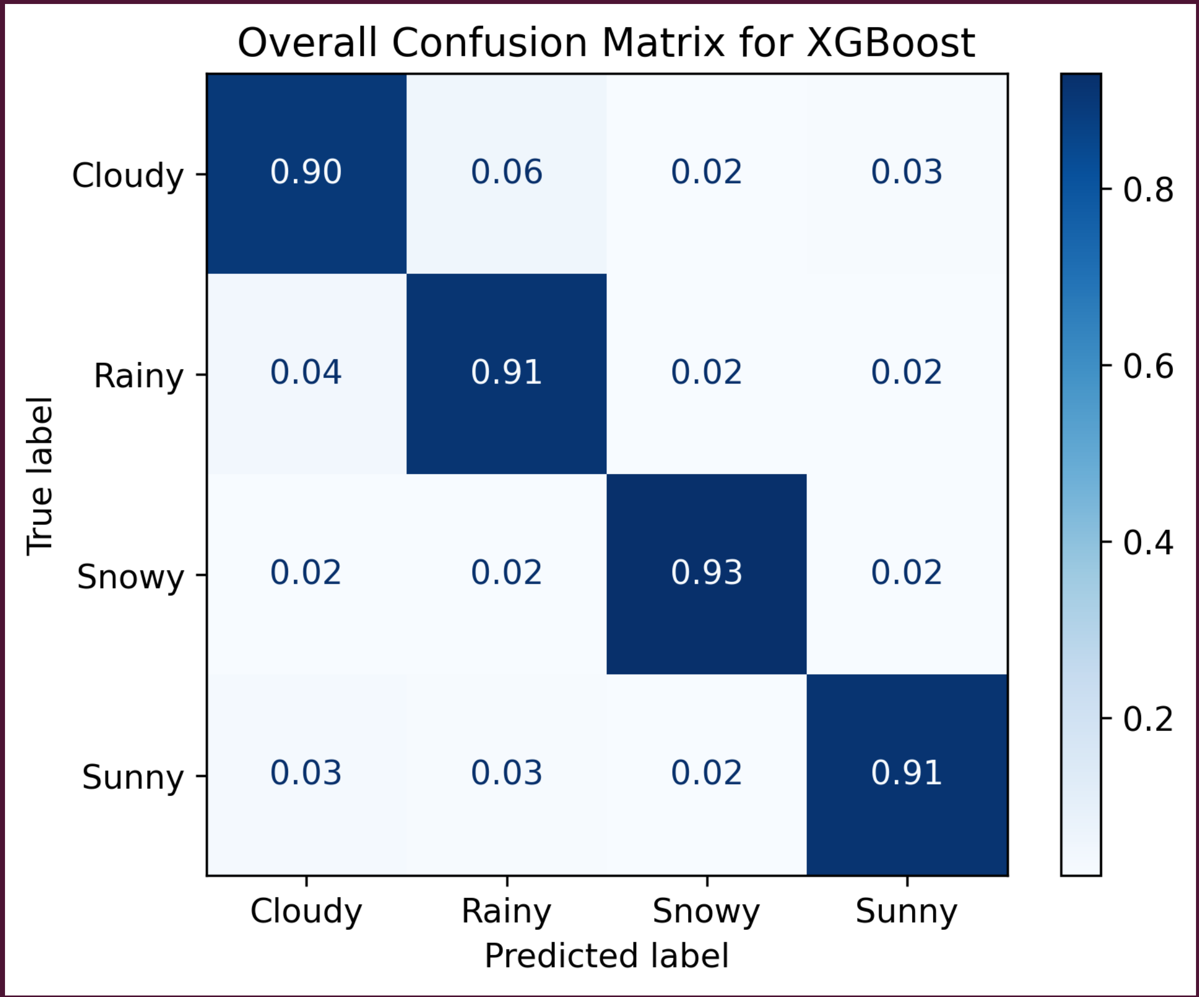
3. RESULTS

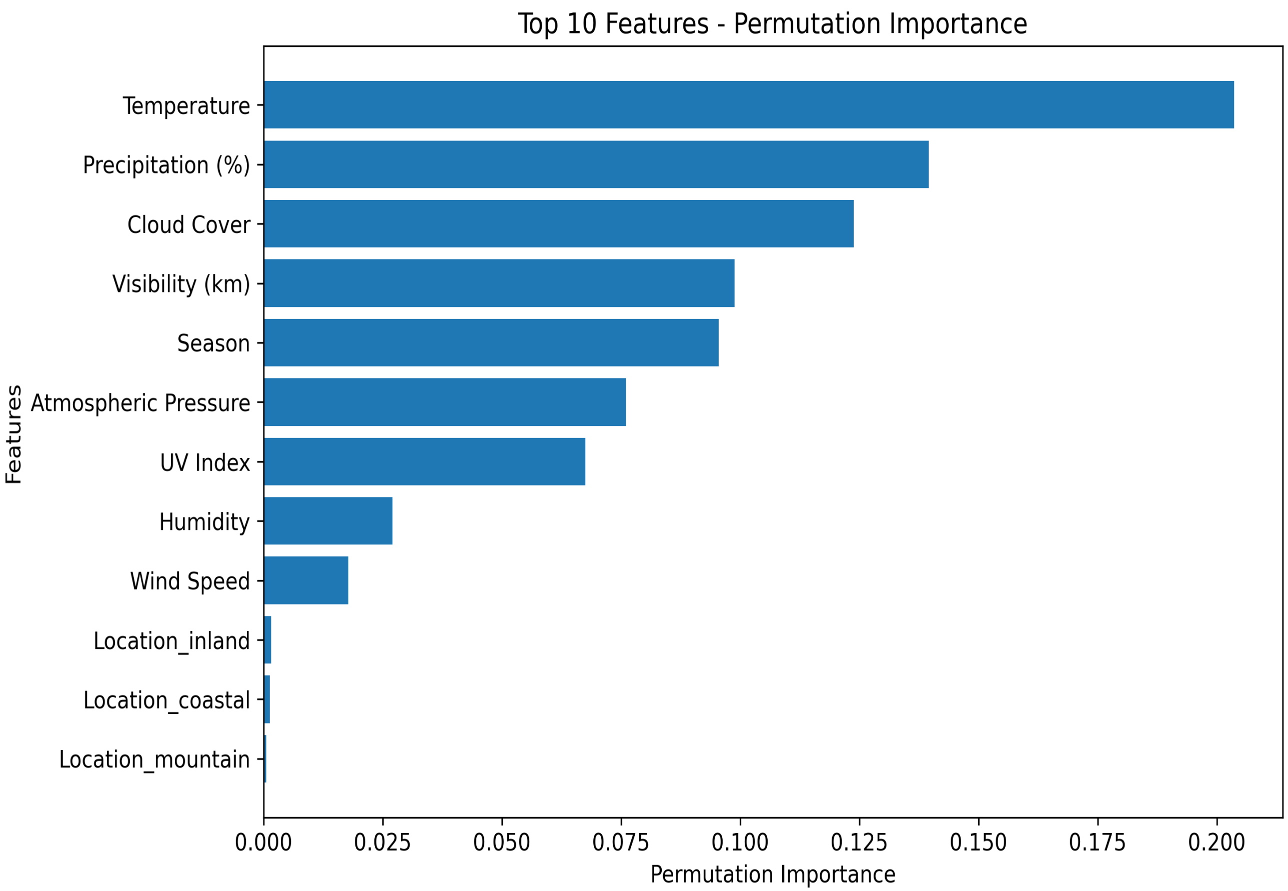
SCORES



Model Comparison on Test Accuracy
Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Baseline Accuracy (0.25)
Model Accuracy
Logistic Regression
Random Forest
SVM
XGBoost
KNN
Models

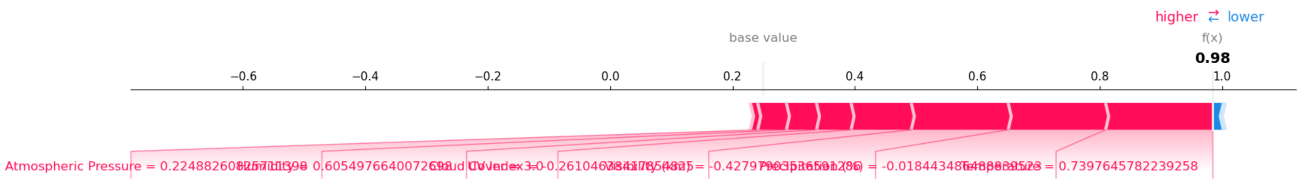
COMFUSION MATRIX OF XGBOOST

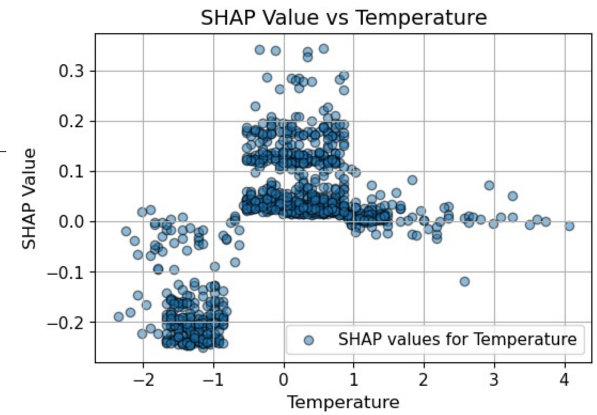


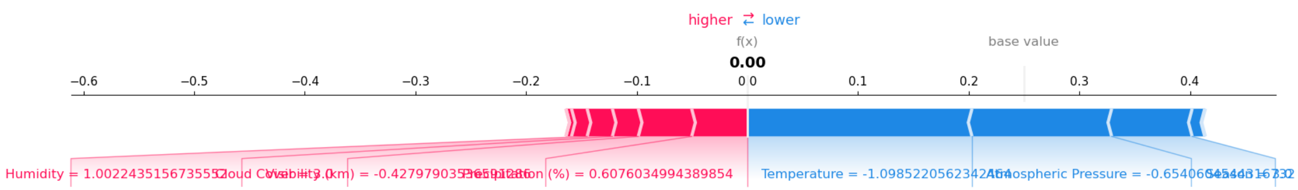

Top 10 Features - Permutation Importance
Temperature
Precipitation (%)
Cloud Cover
Visibility (km)
Season
Atmospheric Pressure
UV Index
Humidity
Wind Speed
Location_inland
Location_coastal
Location_mountain
Features
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Permutation Importance

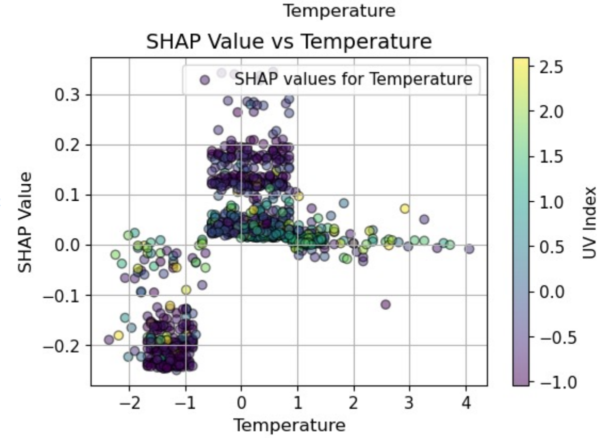
GLOBAL

LOCAL


higher ⇄ lower
base value
f(x)
0.98
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Atmospheric Pressure =
0.7397645782239258


SHAP Value vs Temperature
SHAP Value
Temperature
SHAP values for Temperature


higher ⇄ lower
f(x)
0.00
base value
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
Humidity =
(%) = 0.6076034994389854
Temperature =
Atmospheric Pressure =


SHAP Value vs Temperature
SHAP Value
Temperature
SHAP values for Temperature
UV Index

STEPS TAKEN TO MAKE THE MODEL INTERPRETABLE

Feature Importance Analysis:

Feature_importances_ :

- Extracted feature importances for models like Random Forest and XGBoost using the feature_importances_ attribute.
- Visualized global feature importances with horizontal bar plots to identify the most influential features for model predictions.

Permutation Importance:

Applied permutation importance to assess the impact of individual features on model performance.

Shuffled the values of each feature in the test set and observed how it affected the accuracy of the model.

Visualized results with boxplots to show the variability of model performance when features were perturbed.

SHAP (SHapley Additive exPlanations):

Local Interpretability:

- Used SHAP to explain individual predictions.
- Visualized SHAP force plots for specific test instances to show how each feature contributed to the prediction, relative to the base value.
- Highlighted features that increased or decreased the likelihood of a specific class.

Confusion Matrix Analysis:

Generated confusion matrices to inspect how well the model classified instances of each class.

Normalized confusion matrices for better visualization of errors across classes.

Used decoded class labels for better interpretability of the confusion matrix.

I LEARNED FROM MODELS

01

Model Performance:

- **XGBoost** and **Random Forest** achieved the best accuracy and stability. (Both above 91.2%)
- **Temperature, Precipitation, Cloud Cover** were the most impactful features across models.

02

Interpretability:

- **SHAP values** provided insights into global and local feature contributions.
- Permutation importance confirmed feature significance, with notable drops in accuracy when key features were shuffled.

I LEARNED FROM PROJECT

After encoding, careful attention to the order of feature name concatenation to prevent feature-value mismatches

.**XGBoost Specifics**: XGBoost is highly effective in handling missing values and performs exceptionally well in classification tasks. However, it requires target variables to be numeric and cannot directly process string-based target variables.

4. OUTLOOK

MODELS

- **Deep Learning**:
  - Explore neural networks, particularly for large datasets or when patterns might be nonlinear and complex.

INTERPRETABILITY

- **Check all the importance in the 5 metrics**:
  - Weight, Gain, Cover, Total_gain, Total_cover.
- **Partial Dependence Plots (PDP)**:
  - Examined the marginal relationship between a feature and the predicted outcome while controlling for other features.
  - Visualized how specific features (e.g., Temperature) impacted predictions across the feature range.

THANKS FOR LISTENING