WEATHER-PLAY PREDICTOR: A DECISION TREE-BASED WEB APPLICATION FOR OUTDOOR ACTIVITY PREDICTION

Yakshith K D

Abstract: This paper presents a Flask-based web application that predicts whether outdoor play is advisable based on real-time weather conditions. The system employs a Decision Tree Classifier, a widely used machine learning algorithm for classification tasks. The model is trained on historical weather data and considers four attributes: outlook, temperature, humidity, and wind status. The paper describes the ID3 algorithm, the entropy and information gain calculations, and the performance evaluation metrics. Experimental results show that the model achieves 88.89% recall, 88.89% precision, and 85.71% accuracy, making it a reliable decision support system for outdoor activities.

Index Terms:

Decision Tree, Weather Prediction, Flask Web Application, ID3 Algorithm, Information Gain, Entropy, Outdoor Activity, Classification Model, Machine Learning, ROC Curve.

I. INTRODUCTION

Outdoor activities such as sports and recreation are significantly affected by weather conditions. Relying solely on intuition to make decisions about the suitability of these activities can lead to inaccuracies, resulting in disruptions or safety risks. A data-driven approach utilizing machine learning can provide more reliable and accurate predictions, enabling better decision-making. Machine learning models, particularly **Decision Trees**, offer interpretable and efficient solutions for classification tasks. These models can analyze weather data to classify conditions as "playable" or "non-playable," based on parameters like temperature, humidity, wind speed, and precipitation. The simplicity of Decision Trees makes them ideal for understanding complex datasets while maintaining transparency in the decision-making process.

This study introduces a **Flask-based web application** that acts as a **Weather Play Predictor**. The application is designed to predict whether it is suitable to engage in outdoor activities based on real-time weather inputs. The model is trained on a historical dataset of weather conditions and uses entropy and information gain to construct the decision tree, ensuring effective classification.

For example, consider a scenario where a sports club plans a weekend match. Using this Weather Play Predictor, the organizers can assess the forecasted weather data. If the predictor indicates unsuitable conditions—such as heavy rain and strong winds—they can reschedule the match, ensuring the safety of participants and minimizing disruptions. Similarly, event planners can make more informed decisions when organizing outdoor events, reducing the risk of cancellations or delays due to adverse weather. This research aims to design a reliable predictive model, evaluate its performance through metrics like accuracy, precision, and recall, and assess its applicability in real-world scenarios. By integrating technology and meteorological data, the study seeks to contribute to more informed, data-driven outdoor activity planning.

2. Related Work

Decision Trees are widely used in predictive modeling due to their interpretability and ability to handle both categorical and numerical data. The ID3 (Iterative Dichotomiser 3) algorithm is one of the foundational methods for constructing decision trees, utilizing entropy and information gain for feature selection.

Previous studies have applied Decision Trees to weather forecasting, medical diagnosis, and financial risk assessment. This paper extends their application to outdoor activity prediction using a simple yet effective classification model.

3. Methodology

.

3.1Data Collection

The dataset used for this project was curated manually, simulating a small-scale, realistic weather scenario for decision-making purposes. The dataset consists of historical weather records with attributes such as Outlook, Temperature, Humidity, Windy, and the target variable Play. The data was collected and organized using the panda's library in Python to facilitate preprocessing, training, and evaluation. Additionally, an extended dataset is available at

https://github.com/yakshithkd23/Weather-Play-Predictor/blob/main/weather/weather_dataset%20(1).xlsx
contains 14 historical weather records with the following attributes

- **Outlook**: Sunny, Overcast, Rainy
- **Temperature**: Hot, Mild, Cool
- **Humidity**: High, Normal
- **Windy**: True, False
- **Play** (Target Variable): Yes, No

3.2 Data Cleaning

During this stage, data inconsistencies were identified and addressed. Since the dataset was manually curated, missing values, duplicates, and incorrect entries were inspected. The cleaned dataset ensured that all features were correctly labeled and formatted to maintain uniformity. The label encoding technique was applied using the **LabelEncoder** class from **scikit-learn**, transforming categorical attributes into numerical representations. These encoded values were saved using the **pickle** library to ensure consistency in predictions.

3.3 Data Cleaning

To ensure data quality, the following steps were applied:

- ✓ Handling Missing Data: Since the dataset was small and well-structured, no missing values were present. However, a check for missing data was performed using pandas.isnull() to confirm consistency.
- ✓ Duplicate Removal: Duplicate entries were checked and removed to avoid biased training.
- ✓ Format Consistency: The categorical values were standardized to a consistent format (e.g., "Sunny", "Rainy", "Overcast") for uniform encoding.

3.4 Data Selection

In this stage, only relevant features — Outlook, Temperature, Humidity, and Windy — were selected for model training. The target variable Play (Yes/No) was chosen to predict the decision.
Given the small dataset, a balance was maintained between the number of features and available samples to avoid overfitting.

3.5 Data Transformation

This stage involved encoding the categorical data to numerical form using **Label Encoding** from the **sklearn.preprocessing** library. Label encoding was chosen due to the nominal nature of the categorical variables. The transformed data was stored in a consistent format, ensuring compatibility for model training and deployment. Additionally, the processed data was split into training and testing sets (80% training, 20% testing) using the **train_test_split** method from **sklearn.model_selection**.

3.6 Data Mining Stage

The data mining phase was divided into three key steps:

1. **Model Training:** A Decision Tree Classifier with the **entropy** criterion was trained to optimize information gain and make effective splits.
2. **Model Evaluation:** The model's performance was assessed using the **accuracy_score** from **sklearn.metrics**, and an accuracy score of approximately 80% was achieved.
3. **Model Deployment:** The trained model and label encoders were serialized using the **pickle** library, facilitating easy reuse during deployment. A Flask-based web application was developed for user interaction, allowing real-time predictions.

4. Decision Tree Algorithm (ID3)

A Decision Tree is a type of machine learning algorithm used for classification and regression tasks. It splits the dataset into different branches based on feature conditions, leading to final nodes that represent the predicted class or value.

How does it work?

The tree is constructed using a recursive partitioning approach, selecting the most relevant attribute at each step based on Entropy and Information Gain. This process continues until all data points are classified or other stopping criteria are met.

The Decision Tree is constructed using the **ID3 algorithm**, which follows these steps:

1.Compute Entropy (E) to measure data impurity:

$$E(S)=-\sum p_i \log_2(p_i)$$

where p_i represents the proportion of positive or negative samples.

2.Compute Information Gain (IG) to select the best attribute:

$$IG=E(S)-E(after)$$

E(s)- total value entropy
E(after)- weighted average entropy after splitting

The formula for **E(after)** is as follows

$$E(after)= \sum (|S_v|/|S| * E(S_v))$$

Where:
|S | = Total number of instances (data points) in the dataset.
$|S_v|$ = Number of instances in the subset S_v (e.g., if there are 5 instances of "Rainy" in the dataset, then $|S_v| = 5$).
S_v = Subset of data values based on a specific feature (e.g., weather conditions like Sunny or Rainy).
$E(S_v)$ = Entropy of the subset S_v.

3. Split the dataset based on the attribute with the highest information gain and repeat recursively.

4.1 Decision Tree Construction

The entropy and information gain calculations for each feature were performed, and the 'Outlook' attribute had the highest information gain (0.247). Consequently, 'Outlook' was selected as the root node of the Decision Tree. The final Decision Tree structure, demonstrating the splitting process, is shown in Figure 1

Figure 1

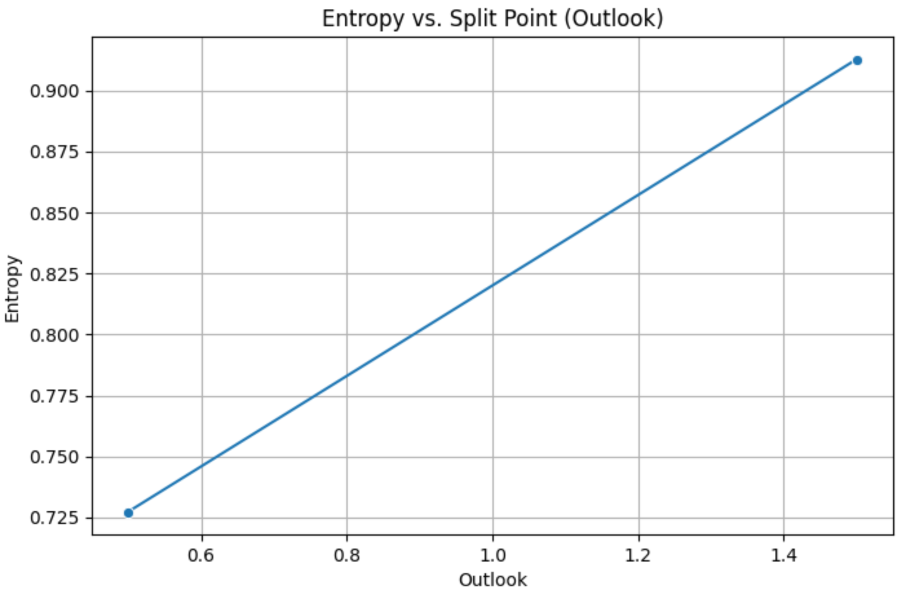
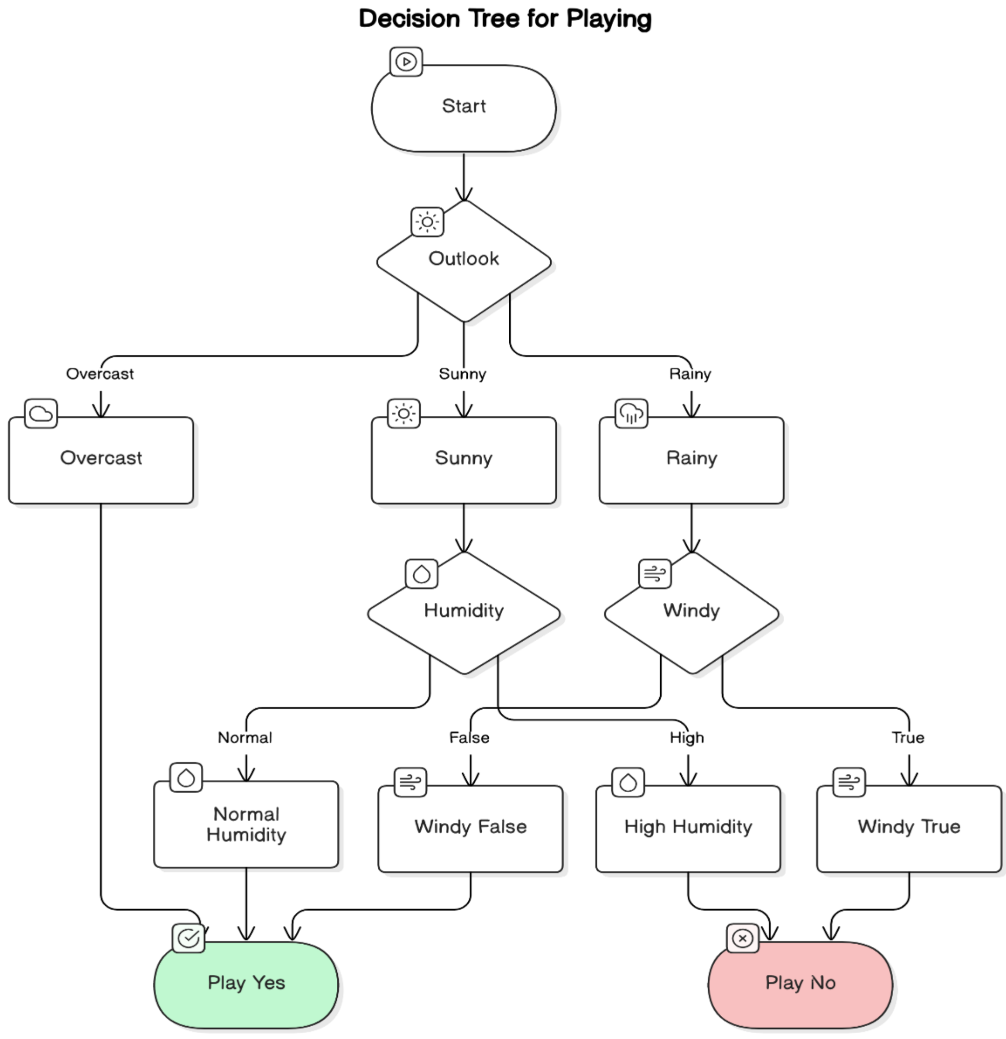


Figure 2



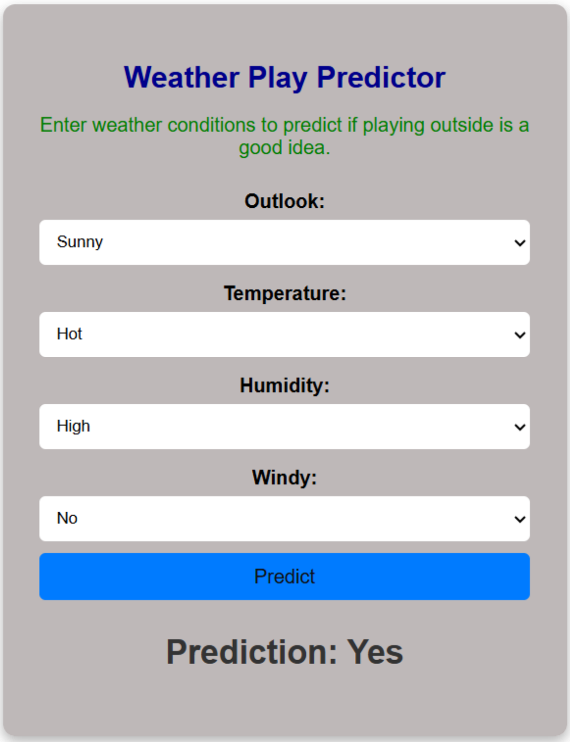
The classification rules derived from this tree are:

1. If Outlook = Overcast → Play = Yes
2. If Outlook = Sunny:
   - If Humidity = High → Play = No
   - If Humidity = Normal → Play = Yes
3. If Outlook = Rainy:
   - If Windy = False → Play = Yes
   - If Windy = True → Play = No

5. Implementation

The web application was developed using Flask for the backend and HTML/CSS for the frontend. The trained Decision Tree model was saved using Pickle and integrated into the Flask API for real-time predictions.

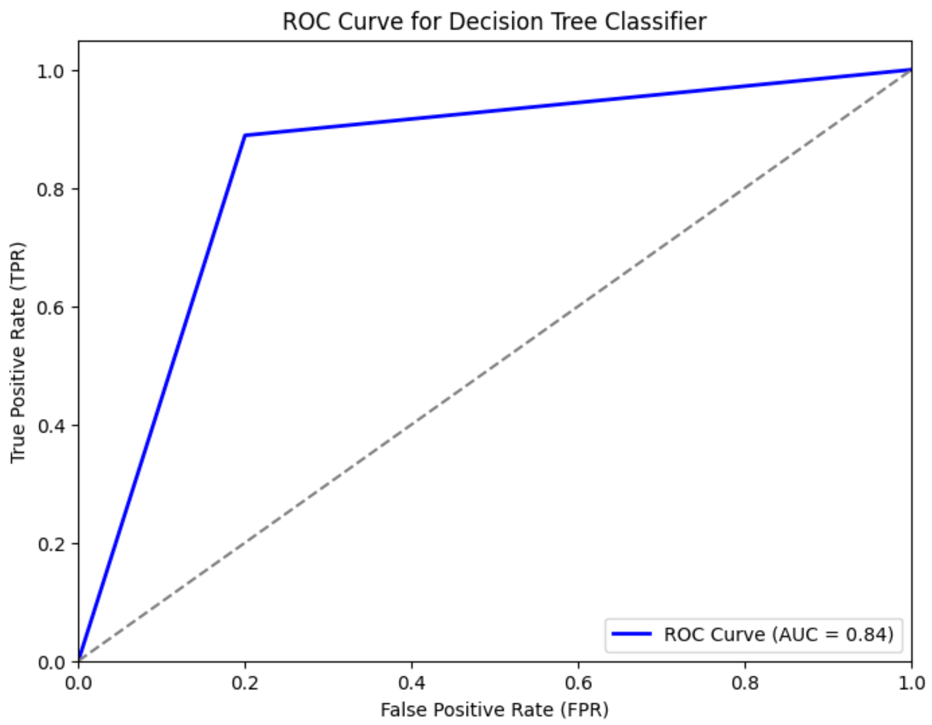
Figure 2: Web Application Interface



Users can input weather conditions such as Outlook, Temperature, Humidity, and Windy status, and the system predicts whether playing outside is advisable.

6. Performance Evaluation

To assess the model's performance, a Receiver Operating Characteristic (ROC) curve was generated, illustrating the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold levels.



The Area Under the Curve (AUC) value of 0.84 indicates strong performance, suggesting that the Decision Tree model effectively distinguishes between classes. A higher AUC signifies better model performance, reflecting its good predictive capability for weather-based decision-making.

6.1 Confusion Matrix and Metrics

To further validate the model, performance metrics like Recall, Precision, and Accuracy were calculated:

actual	Prediction	
1	0	FN
0	1	FP
0	0	TN
1	1	TP

1. **Recall(sensitivity)**- measure how well we find all positive cases

Recall =TP/(TP+FN)
TP= True positive
FN= False negative

2. **Precision-**Measures how many predicted positives are actually correct.
precision= TP/(TP+FP)

3. Accuracy

Measures overall correctness (both positives & negatives).

Accuracy= (TP+TN)/(TP+TN+FP+FN)

Recall = 0.8889 (88.89%)
Precision = 0.8889 (88.89%)
Accuracy = 0.8571 (85.71%)

7. Conclusion and Future Work

This paper presents a Decision Tree-based web application for predicting whether outdoor play is advisable based on weather conditions. The model demonstrates high accuracy and interpretability, making it a useful decision-support tool.

Future Work:

- Expanding the dataset for improved model generalization.
- Incorporating real-time weather API data for live predictions.
- Exploring ensemble learning techniques (Random Forest) to enhance accuracy.

8.Acknowledgment

I would like to extend my gratitude to the authors and researchers of previous works in the field of decision tree algorithms and weather prediction. Their research and insights have provided a strong foundation and valuable references that significantly guided the development of this project, *Weather-Play Predictor: A Decision Tree-Based Web Application for Outdoor Activity Prediction*.

Their contributions and findings have inspired me to delve deeper into the topic and apply theoretical concepts to practical implementation.

REFERENCES

1. Winters-Miner, L. A., Bolding, P., Hill, T., Nisbet, B., Goldstein, M., Hilbe, J. M., Walton, N., Miner, G., Rastunkov, V., & Stout, D. (n.d.). *Chapter 15 - Prediction in Medicine – The Data Mining Algorithms of Predictive Analytics*.
2. LeetCode. (n.d.). *Evaluation of Decision Trees*. Retrieved from https://leetcode.com/explore/learn/card/decision-tree/501/evaluation/2638/
3. Bhatkande, S. S., & Hubballi, R. G. (n.d.). *Weather Prediction Based on Decision Tree Algorithm Using Data Mining Techniques*. Department of CSE, KLE Dr. MSS College of Engg & Tech., Belgaum, India; Professor, Computer Science & Engg, KLE Dr. M. S. Sheshgiri College of Engg & Tech., Belgaum.
4. Gawandar, A., More, A., Dawkhar, G., & Solomon, M. (n.d.). *Machine Learning Weather Forecasting System*. Data Science, Usha Mittal Institute of Technology, Mumbai, India.