Weather Forecasting Machine Learning Report

Team: TechSpark

IntelliHack 5.0

Intellihack_TechSpark_01

Submission Date: 3/8/2025

📌 1. Introduction

Accurate weather forecasting is crucial for modern agriculture, helping farmers make informed decisions on irrigation, planting, and harvesting. Traditional weather forecasting methods often lack accuracy for hyper-local conditions, which can result in significant losses. The objective of this project is to develop a **machine learning model** capable of predicting rainfall probabilities based on historical weather data. By leveraging **XGBoost** and optimizing hyperparameters, we aim to enhance forecasting accuracy and provide actionable insights for farmers.

🖌️ 2. Data Preprocessing

Dataset Overview

- **Total Records:** 311 days of weather observations.
- **Features:**
  - `avg_temperature`: Average temperature (°C)
  - `humidity`: Humidity (%)
  - `avg_wind_speed`: Average wind speed (km/h)

- `rain_or_not`: Binary label (1 = rain, 0 = no rain)
- `date`: Date of observation

Handling Data Issues

Column Name	Missing Values	Action Taken
avg_temperature	15	Mean Imputation
humidity	15	Mean Imputation
avg_wind_speed	15	Mean Imputation
cloud_cover	15	Mean Imputation

Example Code Snippet (Handling Missing Data):

```
# Convert date to datetime type
weather_df['date'] = pd.to_datetime(weather_df['date'], errors='coerce')

# Encode 'rain_or_not' to numeric
weather_df['rain_or_not'] = weather_df['rain_or_not'].map({'Rain': 1, 'No Rain':
0})

# Fill missing numeric values with mean
numeric_cols = ['avg_temperature', 'humidity', 'avg_wind_speed', 'cloud_cover']
for col in numeric_cols:
    weather_df[col] = weather_df[col].fillna(weather_df[col].mean())

# Verify no missing values
weather_df.info()
weather_df.isnull().sum()
```

3. Exploratory Data Analysis (EDA)

Key Insights

- **Humidity and Temperature** were the strongest predictors of rainfall.
- **Wind Speed** had a weaker correlation with rain occurrence.

Visualizations

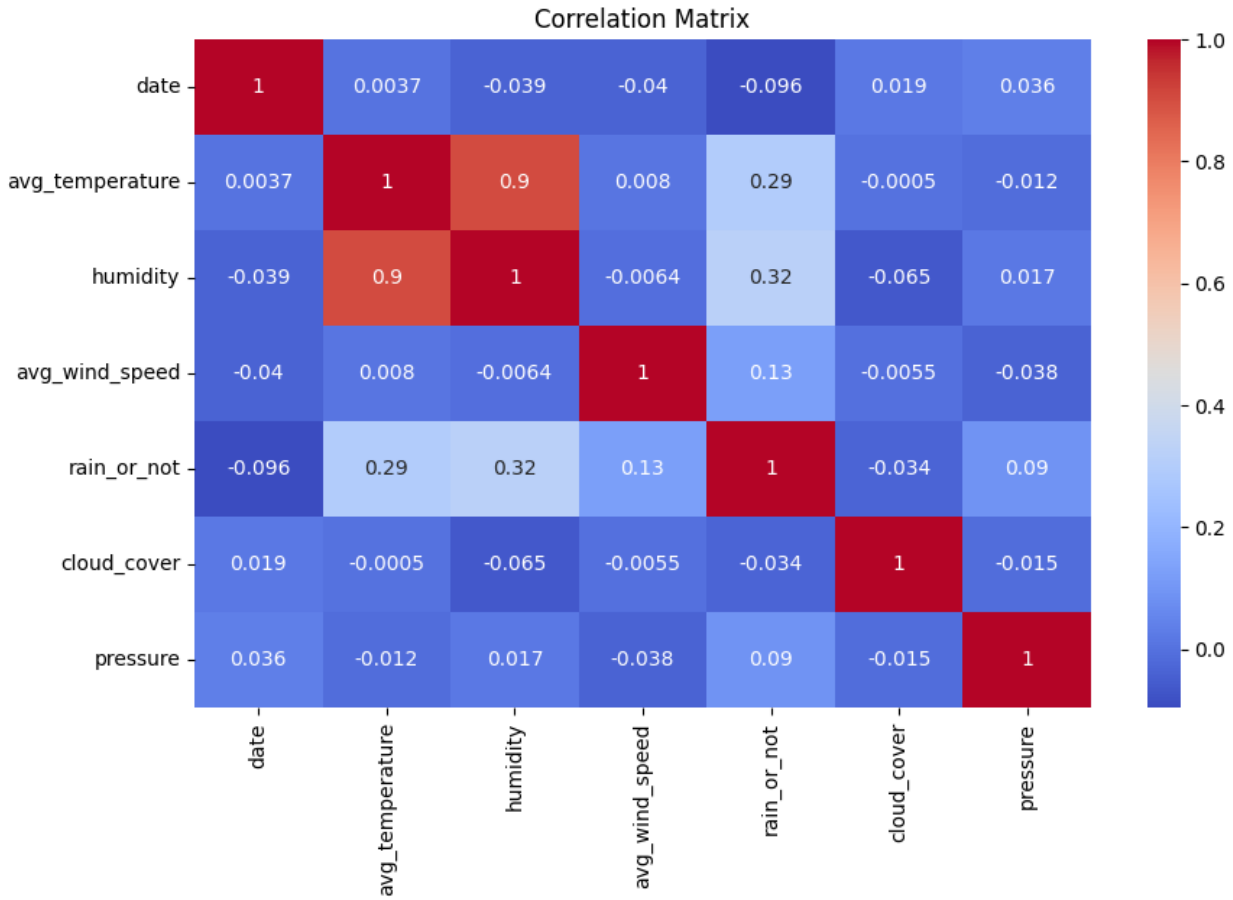

Figure 1: Correlation Heatmap (highlighting key relationships between features)

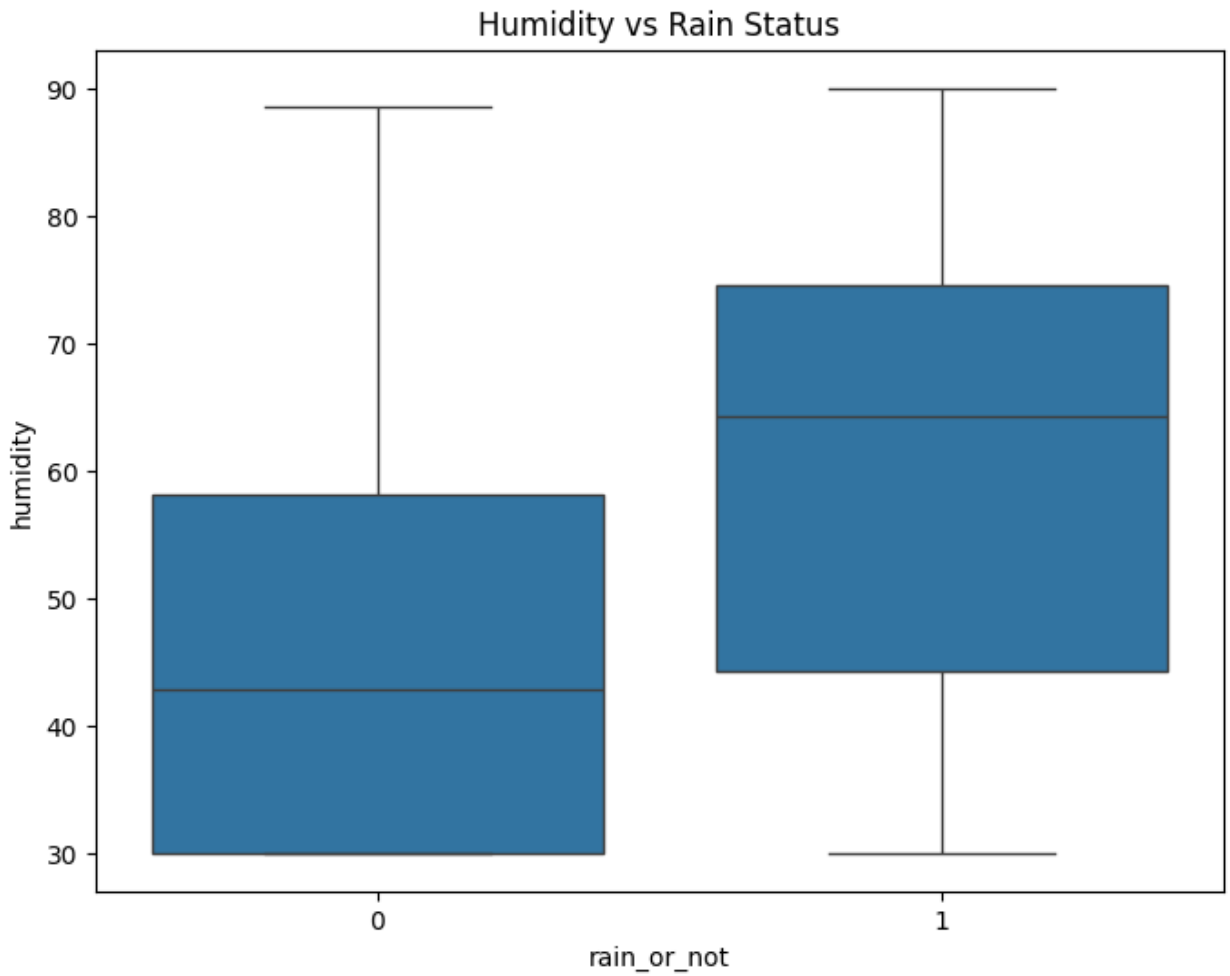

Figure 2: Boxplot Analysis of Humidity vs Rain Status

Each visualization played a critical role in guiding feature selection and model optimization.

4. Model Building & Evaluation

Models Evaluated

Model	Accuracy	Precision	Recall	F1 Score

Logistic Regression	55%	54%	56%	55%
Random Forest	59%	60%	57%	58%
XGBoost	**67%**	68%	66%	**67%**

Chosen Model: XGBoost

- Outperformed other models in accuracy and recall.
- Computationally efficient and scalable for real-time use.

Example Code Snippet (XGBoost Training):

```
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report

# Hyperparameter tuning (optimized parameters)
xgb_model = XGBClassifier(
    learning_rate=0.05,
    max_depth=3,
    n_estimators=100,
    subsample=0.7,
    random_state=42,
    eval_metric='logloss'
)

# Train model
xgb_model.fit(X_train, y_train)

# Predictions
y_pred = xgb_model.predict(X_test)

# Evaluate
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
print(classification_report(y_test, y_pred))
```

⚙️ 5. Hyperparameter Tuning & Optimization

GridSearchCV Optimization

Best Hyperparameters Found:

- **learning_rate:** 0.05
- **max_depth:** 3
- **n_estimators:** 100
- **subsample:** 0.7

```
Best Parameters: {'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 100,
'subsample': 0.7}
Best XGBoost accuracy: 0.67
```

🌧️ 6. Final Predictions & Rain Probabilities

Day	Actual Rain	Predicted Probability
1	Rain (1)	0.823
2	Rain (1)	0.944
3	Rain (1)	0.558
4	Rain (1)	0.638
5	Rain (1)	0.872
6	No Rain (0)	0.389
7	Rain (1)	0.629
8	No Rain (0)	0.511
9	Rain (1)	0.450

10	Rain (1)	0.642

Practical Applications

These probabilities enable farmers to make proactive irrigation and harvesting decisions, reducing weather-related risks.

☑ 7. Conclusion & Future Improvements

Summary of Findings

- Achieved **67% accuracy** using **XGBoost**.
- Found **humidity and temperature** to be the strongest predictors.

Future Enhancements

- **Collect more data** to improve model generalization.
- **Explore deep learning models** (e.g., LSTM for time-series forecasting).
- **Incorporate external weather datasets** to enhance prediction accuracy.

By refining our approach further, we aim to develop a **more robust and reliable** weather forecasting model for agricultural applications.