The Libyan Academy
School of Applied Sciences and Engineering
Department of Electrical and Computer Engineering
Branch: Information Technology

Emerging Internetworking Technologies - ITE646

Predicting Forest Fires using Machine Learning: A Comprehensive Analysis and Model Development Journey

By:
Fooz Barakat

Agenda

- **Introduction to the Project and the Problem Definition** ✅
- Packages Used in The Project
- Data Overview and Features of the Used Data
- Data Exploration
- Machine Learning Models
- Evaluating Our Tuned Machine Learning
- Conclusion

Introduction to the Project and the Problem Definition

- ***Problem definition:*** Predicting forest fires using machine learning.
- ***Goal:*** Develop a model for accurate prediction based on environmental factors.
- ***Relevance:*** Critical for early detection and mitigation efforts, reducing impact on ecosystems and human communities.
- ***Importance:*** Demonstrates the application of machine learning in environmental research.
- ***Significance:*** Highlights the potential of machine learning to solve complex problems in various fields.

Agenda

- Introduction to the Project and the Problem Definition
- **Packages Used in The Project** ✅
- Data Overview and Features of the Used Data
- Data Exploration
- Machine Learning Models
- Evaluating Our Tuned Machine Learning
- Conclusion

Packages Used in The Project

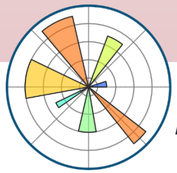
Numpy

- ***Role:*** Used for numerical computing.
- ***Contribution:*** Provides support for large, multi-dimensional arrays and matrices, essential for mathematical operations in machine learning algorithms.

Pandas

- ***Role:*** Used for data manipulation and analysis.
- ***Contribution:*** Helps in loading the dataset, handling missing values, and preparing the data for analysis and modeling.

Matplotlib



- ***Role:*** Used for data visualization.
- ***Contribution:*** Helps in creating visualizations such as line plots, scatter plots, and histograms to understand the data distribution and relationships

Seaborn

- ***Role:*** Built on top of matplotlib, used for statistical data visualization.
- ***Contribution***: Provides a high-level interface for drawing attractive and informative statistical graphics, enhancing the visualizations.

Scikit-learn

- ***Role:*** Used for machine learning tasks.
- ***Contribution:*** Provides a wide range of tools for building machine learning models, including classification, regression, clustering, and dimensionality reduction algorithms. It also offers tools for model selection and evaluation.

Packages Used in The Project

These tools are essential for this project as they perform exploratory data analysis (EDA), transform data, and build, evaluate, and tune machine learning models.

Predicting Forest Fire

```
# Import all the tools we need

# Regular EDA (exploring data analysis) and plotting libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Turn the categories into numbers
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

# Models from Scikit-Learn
from sklearn.svm import LinearSVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier

# Model Evaluation
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import plot_roc_curve
```

Agenda

- Introduction to the Project and the Problem Definition
- Packages Used in The Project
- **Data Overview and Features of the Used Data** ✅
- Data Exploration
- Machine Learning Models
- Evaluating Our Tuned Machine Learning
- Conclusion

Data Overview and Features of the Used Data

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	not fire
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire

- ***Date :*** (DD/MM/YYYY) Day, month ('june' to 'september'), year (2012)
- ***Temp*** : temperature noon (temperature max) in Celsius degrees: 22 to 42
- ***RH :*** Relative Humidity in %: 21 to 90
- ***Ws*** :Wind speed in km/h: 6 to 29
- ***Rain:*** total day in mm: 0 to 16.8
- ***FFMC:*** Fine Fuel Moisture Code (FFMC) index from the FWI system: 28.6 to 92.5
- ***DMC:*** Duff Moisture Code (DMC) index from the FWI system: 1.1 to 65.9
- ***DC***: Drought Code (DC) index from the FWI system: 7 to 220.4
- ***ISI:*** Initial Spread Index (ISI) index from the FWI system: 0 to 18.5
- ***BUI:*** Buildup Index (BUI) index from the FWI system: 1.1 to 68
- ***FWI:*** Fire Weather Index (FWI) Index: 0 to 31.1
- ***Classes:*** two classes, namely fire and no fire.

Agenda

- Introduction to the Project and the Problem Definition
- Packages Used in The Project
- Data Overview and Features of the Used Data
- **Data Exploration** ✅
- Machine Learning Models
- Evaluating Our Tuned Machine Learning
- Conclusion

Data Exploration

First we need to clean the data so we can start exploring

```
day             0
month           0
year            0
Temperature     0
RH              0
Ws              0
Rain            0
FFMC            0
DMC             0
DC              0
ISI             0
BUI             0
FWI             0
Classes         0
dtype: int64    1
```

```
fire           131
not fire       100
fire             4
not fire         2
fire             2
not fire         2
not fire         1
not fire         1
Name: Classes, dtype: int64
```

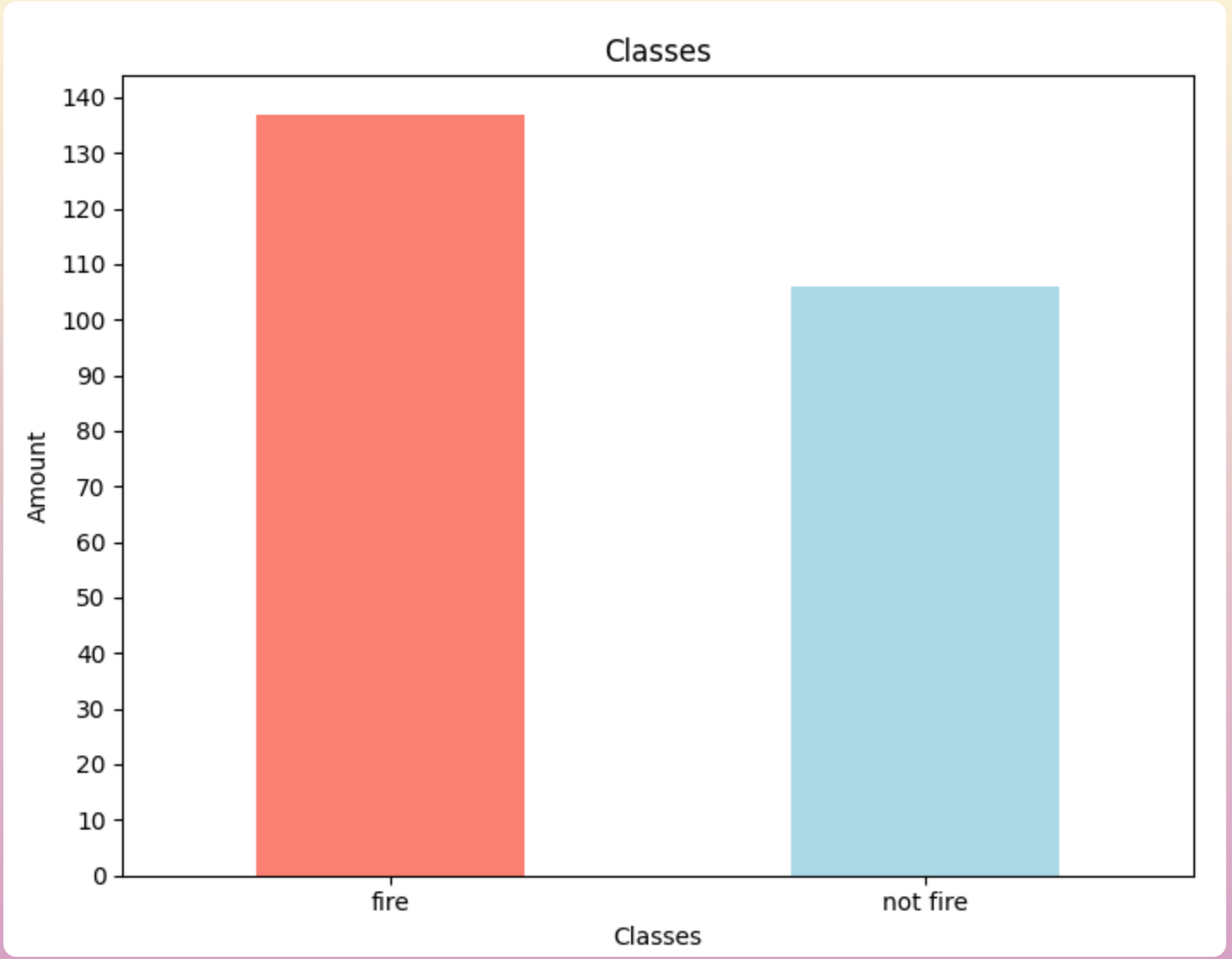
Predicting Forest Fire

```python
# ⇒ Load data
df_1 = pd.read_csv("./Bejaia region Dataset.csv")
df_2 = pd.read_csv("./Sidi-Bel Abbes Region Dataset.csv")

# merge the two datasets
df = df_1.append(df_2)

#Let's find out if we have an empty cells in the data
df.isna().sum()

# drop the missing target varibles
df.dropna(inplace=True)

# Let's find out the target classes
df["Classes"].value_counts()

# remove the extra spaces
# 1,2,3,5 ⇒ not fire
# 1, 3 ⇒ fire
df["Classes"].replace({"not fire ": "not fire", "not fire  ": "not fire",
"not fire   ": "not fire", "not fire    ": "not fire", "fire ": "fire",
"fire   ": "fire"}, inplace=True)
```

Data Exploration

Predicting Forest Fire

```
df["Classes"].value_counts().plot(kind="bar",
figsize=(8, 6),color=["salmon", "lightblue"])

plt.title("Classes")

plt.ylabel("Amount")
plt.yticks(np.arange(0, 145, 10))
plt.xticks(rotation=0);
```



Data Exploration

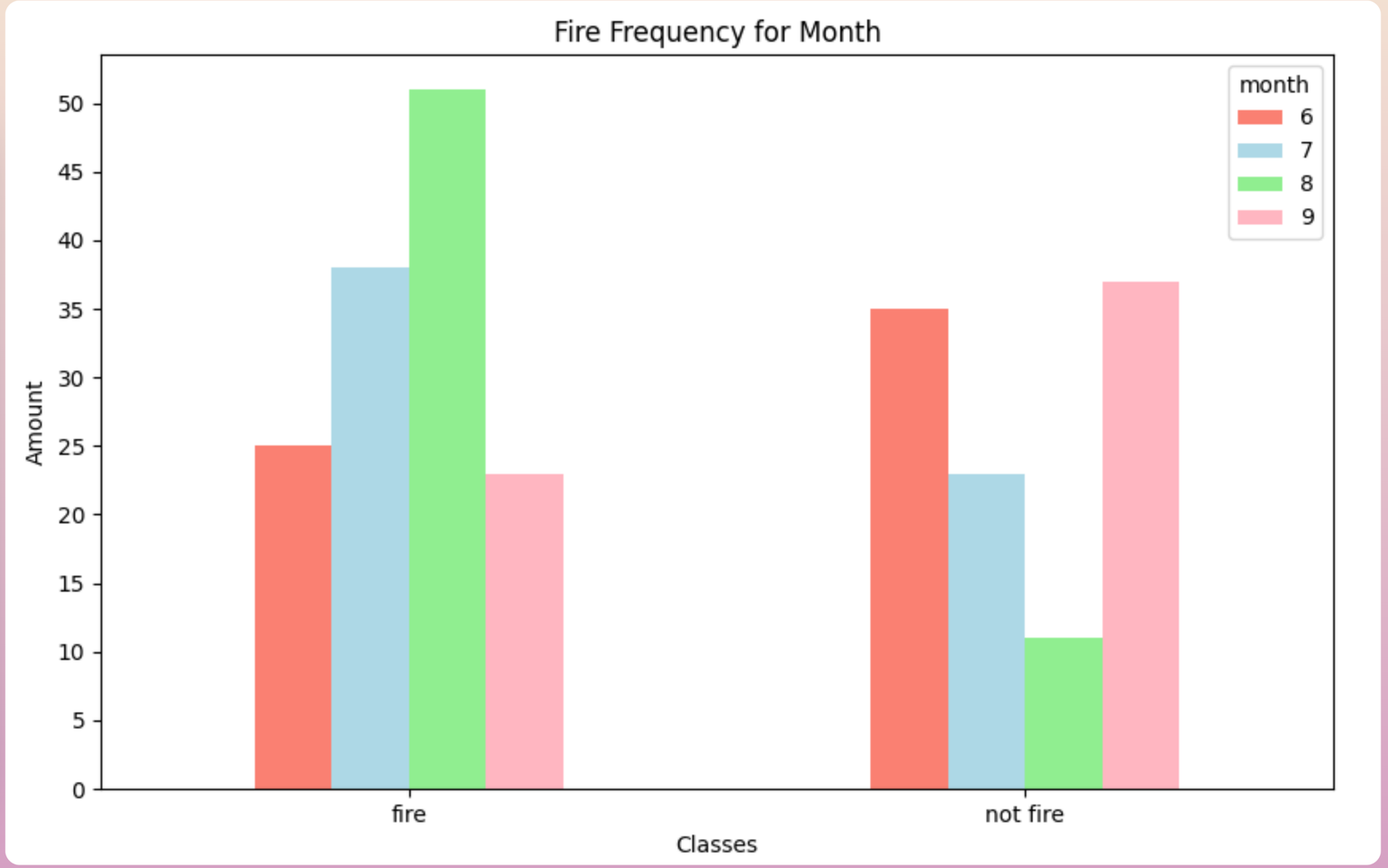
	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	ISI	BUI
count	243.000000	243.000000	243.0	243.000000	243.000000	243.000000	243.000000	243.000000	243.000000	243.000000	243.000000
mean	15.761317	7.502058	2012.0	32.152263	62.041152	15.493827	0.762963	77.842387	14.680658	4.742387	16.690535
std	8.842552	1.114793	0.0	3.628039	14.828160	2.811385	2.003207	14.349641	12.393040	4.154234	14.228421
min	1.000000	6.000000	2012.0	22.000000	21.000000	6.000000	0.000000	28.600000	0.700000	0.000000	1.100000
25%	8.000000	7.000000	2012.0	30.000000	52.500000	14.000000	0.000000	71.850000	5.800000	1.400000	6.000000
50%	16.000000	8.000000	2012.0	32.000000	63.000000	15.000000	0.000000	83.300000	11.300000	3.500000	12.400000
75%	23.000000	8.000000	2012.0	35.000000	73.500000	17.000000	0.500000	88.300000	20.800000	7.250000	22.650000
max	31.000000	9.000000	2012.0	42.000000	90.000000	29.000000	16.800000	96.000000	65.900000	19.000000	68.000000

Data Exploration

month	6	7	8	9
Classes				
fire	25	38	51	23
not fire	35	23	11	37

Predicting Forest Fire

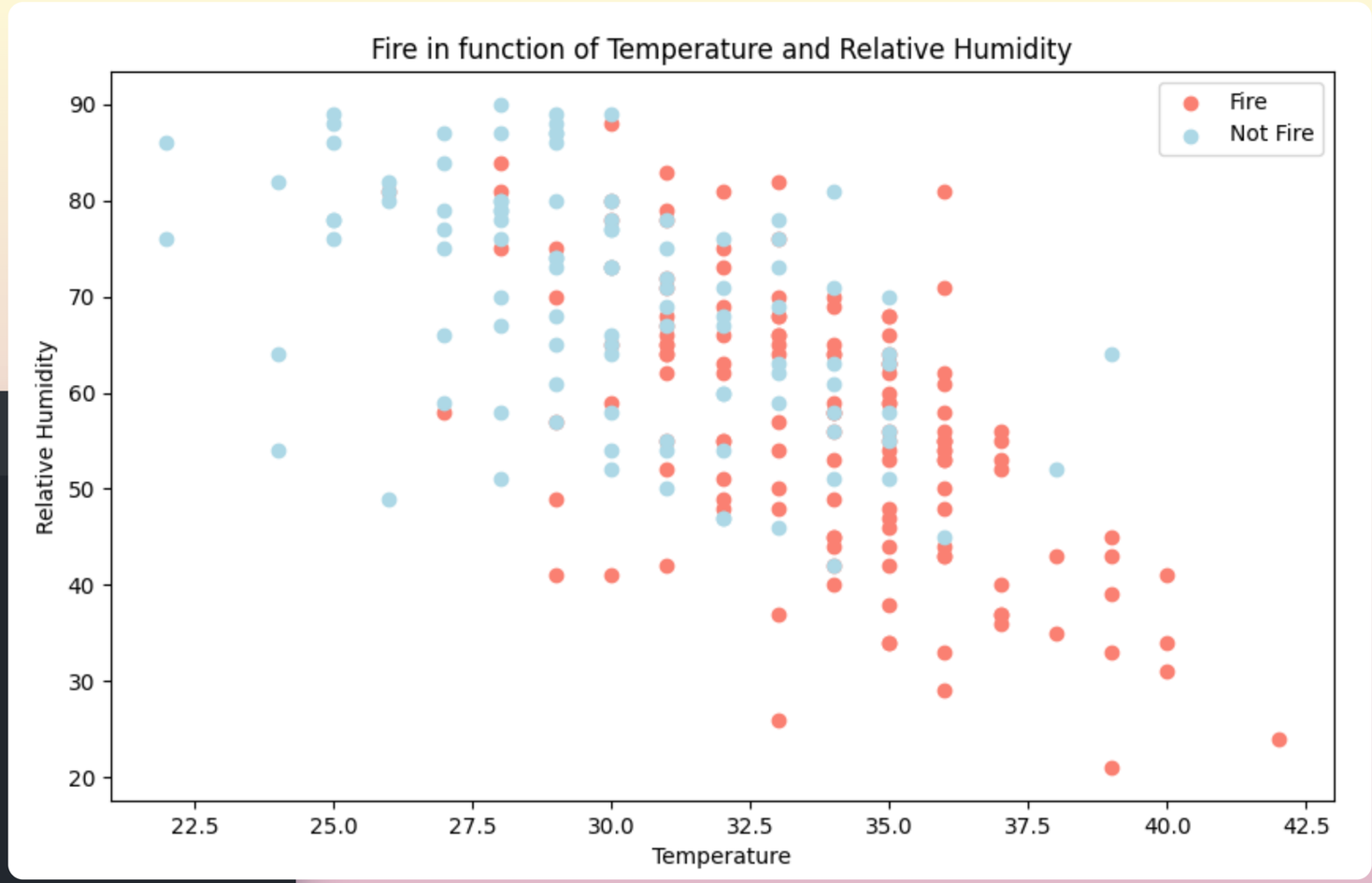
```
# Compare Classes column with month column
pd.crosstab(df.Classes, df.month)

# Create a plot of crosstab
pd.crosstab(df.Classes, df.month).plot(
kind="bar", figsize=(10, 6),
color=["salmon", "lightblue", "lightgreen",
"lightpink"])

plt.title("Fire Frequency for Month")

plt.ylabel("Amount")
plt.yticks(np.arange(0, 55, 5))
plt.xticks(rotation=0);
```



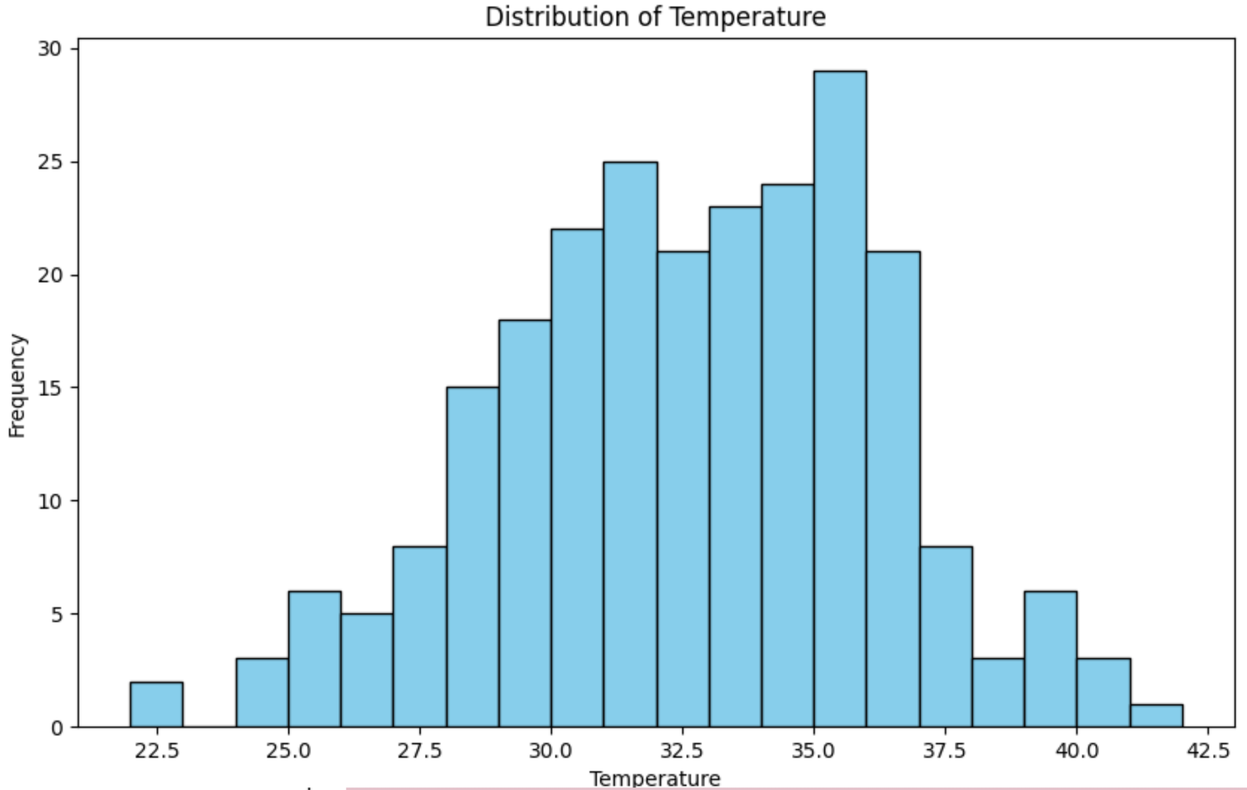
Data Exploration

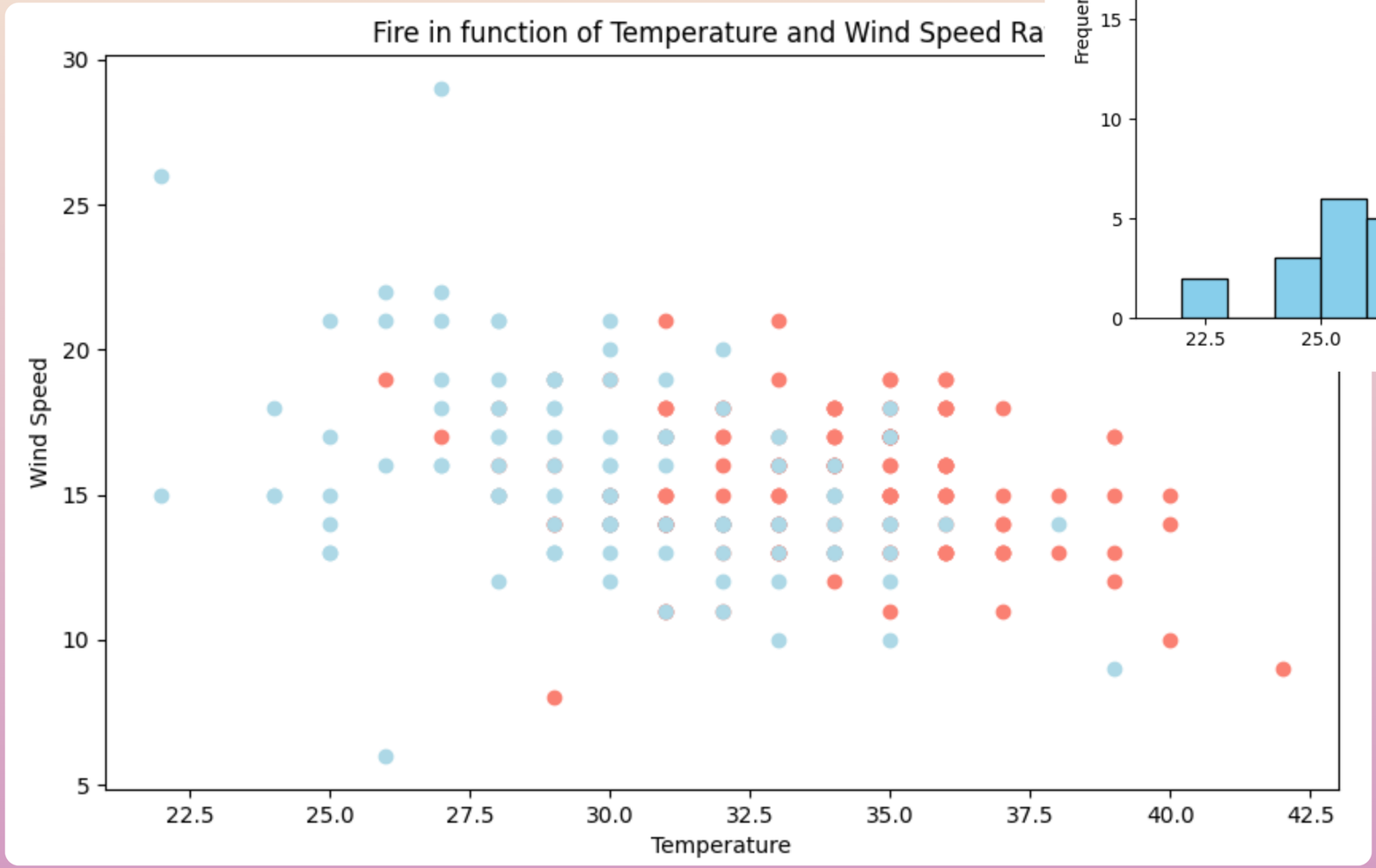
Predicting Forest Fire

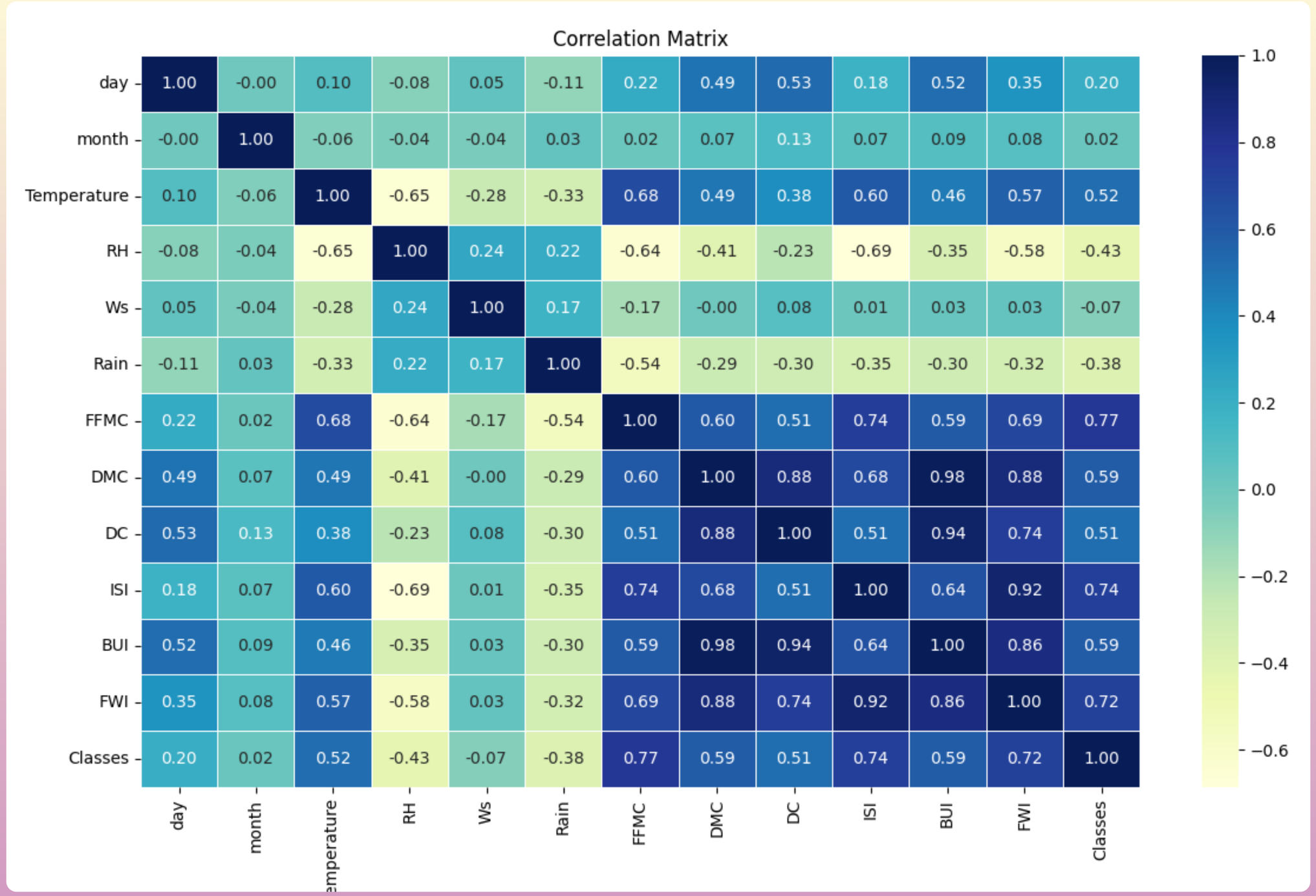
```
# Create another figure
plt.figure(figsize=(10, 6))

# Scatter with postivie examples
plt.scatter(df.Temperature[df.Classes=="fire"],
            df.RH[df.Classes=="fire"],
            c="salmon")

# Scatter with negative examples
plt.scatter(df.Temperature[df.Classes=="not fire"],
            df.RH[df.Classes=="not fire"],
            c="lightblue")

# Add some helpful info
plt.title("Fire in function of Temperature and Relative Humidity")
plt.xlabel("Temperature")
plt.ylabel("Relative Humidity")
plt.legend(["Fire", "Not Fire"]);
```



Data Exploration






Correlation Matrix
day 1.00 -0.00 0.10 -0.08 0.05 -0.11 0.22 0.49 0.53 0.18 0.52 0.35 0.20
month -0.00 1.00 -0.06 -0.04 -0.04 0.03 0.02 0.07 0.13 0.07 0.09 0.08 0.02
Temperature 0.10 -0.06 1.00 -0.65 -0.28 -0.33 0.68 0.49 0.38 0.60 0.46 0.57 0.52
RH -0.08 -0.04 -0.65 1.00 0.24 0.22 -0.64 -0.41 -0.23 -0.69 -0.35 -0.58 -0.43
Ws 0.05 -0.04 -0.28 0.24 1.00 0.17 -0.17 -0.00 0.08 0.01 0.03 0.03 -0.07
Rain -0.11 0.03 -0.33 0.22 0.17 1.00 -0.54 -0.29 -0.30 -0.35 -0.30 -0.32 -0.38
FFMC 0.22 0.02 0.68 -0.64 -0.17 -0.54 1.00 0.60 0.51 0.74 0.59 0.69 0.77
DMC 0.49 0.07 0.49 -0.41 -0.00 -0.29 0.60 1.00 0.88 0.68 0.98 0.88 0.59
DC 0.53 0.13 0.38 -0.23 0.08 -0.30 0.51 0.88 1.00 0.51 0.94 0.74 0.51
ISI 0.18 0.07 0.60 -0.69 0.01 -0.35 0.74 0.68 0.51 1.00 0.64 0.92 0.74
BUI 0.52 0.09 0.46 -0.35 0.03 -0.30 0.59 0.98 0.94 0.64 1.00 0.86 0.59
FWI 0.35 0.08 0.57 -0.58 0.03 -0.32 0.69 0.88 0.74 0.92 0.86 1.00 0.72
Classes 0.20 0.02 0.52 -0.43 -0.07 -0.38 0.77 0.59 0.51 0.74 0.59 0.72 1.00
day
month
emperature
RH
Ws
Rain
FFMC
DMC
DC
ISI
BUI
FWI
Classes
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Agenda

- Introduction to the Project and the Problem Definition
- Packages Used in The Project
- Data Overview and Features of the Used Data
- Data Exploration
- **Machine Learning Models** ✔️
- Evaluating Our Tuned Machine Learning
- Conclusion

Machine Learning Models

- ***Definition:*** Modeling refers to the process of creating and training machine learning models to make predictions or decisions based on data.

- ***Purpose:*** Models are used to find patterns in data and make predictions or decisions without being explicitly programmed.

- ***Types of Models in Machine Learning***
  - **Supervised Learning:**
    - Regression: Predicts a continuous value based on input features.
    - Classification: Predicts a categorical label based on input features.
  - **Unsupervised Learning:**
    - Clustering: Groups similar data points together without any prior labels.
    - Dimensionality Reduction: Reduces the number of input variables in the data.
  - **Semi-Supervised Learning:** Combines both labeled and unlabeled data for training.
  - **Reinforcement Learning:** Focuses on making a sequence of decisions to maximize a reward.
  - **Neural Networks and Deep Learning:** Models inspired by the structure and function of the brain, capable of learning complex patterns in data.

Machine Learning Models

The models that are used in the project are:

- ***Linear SVC (Support Vector Classifier):***
  - **Type**: Supervised Learning, Classification
  - **Description:** Linear SVC is a type of Support Vector Machine (SVM) that uses a linear kernel to separate classes in a dataset.

- ***K-Nearest Neighbors Classifier:***
  - **Type:** Supervised Learning, Classification
  - **Description:** K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm that classifies new data points based on the majority class among their K nearest neighbors.

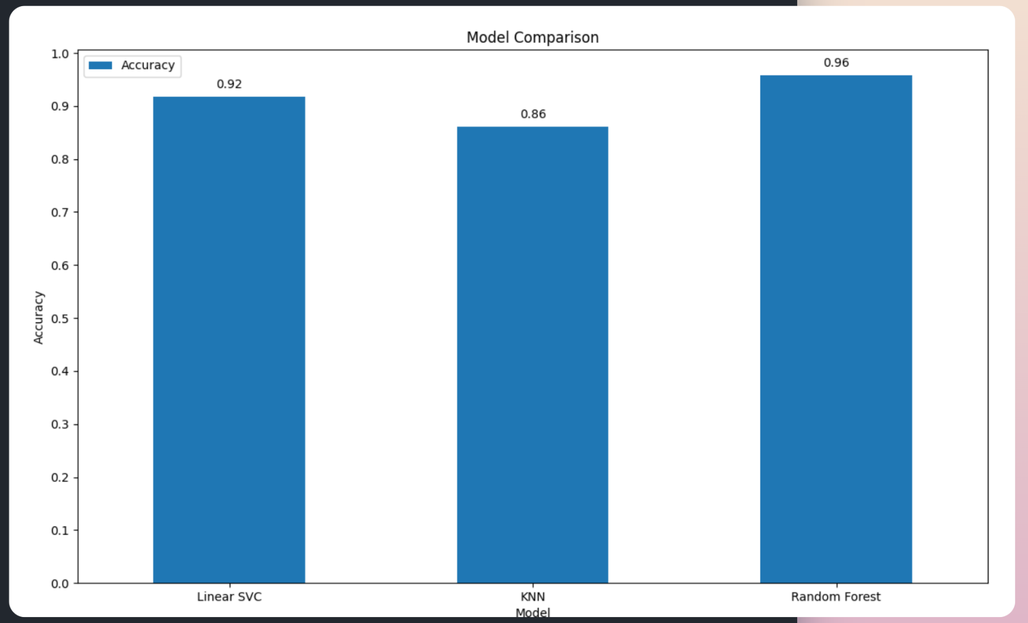
- ***Random Forest Classifier:***
  - **Type:** Supervised Learning, Classification
  - **Description:** Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes of the individual trees.

Machine Learning Models

Fits and evaluates given machine learning models
models : a dict of different Scikit-Learn machine learning models
x_train : training data (no labels)
x_test : testing data (no labels)
y_train : training labels
y_test : test labels

Predicting Forest Fire

```python
# Put models in a dictionary
models = {"Linear SVC": LinearSVC(),
          "KNN": KNeighborsClassifier(),
          "Random Forest": RandomForestClassifier()}

# Create a function to fit and score models
def fit_and_score(models, x_train, x_test, y_train, y_test):
    # Set random seed
    np.random.seed(5)
    # Make a dictionary to keep model scores
    model_scores = {}
    # Loop through models
    for name, model in models.items():
        # Fit the model to the data
        model.fit(x_train, y_train)
        # Evaluate the model and append its score to model_scores
        model_scores[name] = model.score(x_test, y_test)
    return model_scores

model_scores = fit_and_score(models=models, x_train=x_train, x_test=x_test, y_train=y_train,y_test=y_test)
```




```
{'Linear SVC': 0.8767123287671232,
 'KNN': 0.863013698630137,
 'Random Forest': 0.958904109589041}
```

Agenda

- Introduction to the Project and the Problem Definition
- Packages Used in The Project
- Data Overview and Features of the Used Data
- Data Exploration
- Machine Learning Models
- **Evaluating Our Tuned Machine Learning** ✅
- Conclusion

Evaluating Our Tuned Machine Learning

Next Steps for Model Improvement:

- ***Hyperparameter Tuning:*** Adjusting the settings (hyperparameters) of the model to improve performance. This is done manually or through automated techniques like RandomizedSearchCV or GridSearchCV.

- ***Feature Importance:*** Determining which features (attributes) of the dataset are most influential in predicting forest fires. This helps in understanding the underlying factors driving the predictions.

- ***Confusion Matrix***: A table that describes the performance of a classification model. It shows the number of correct and incorrect predictions made by the model compared to the actual outcomes.

- ***Cross-Validation:*** A technique used to assess how the results of a model will generalize to an independent dataset. It helps prevent overfitting.

Evaluating Our Tuned Machine Learning

- ***Precision, Recall, F1 Score:*** Metrics used to evaluate the performance of a classification model, especially when the classes are imbalanced.

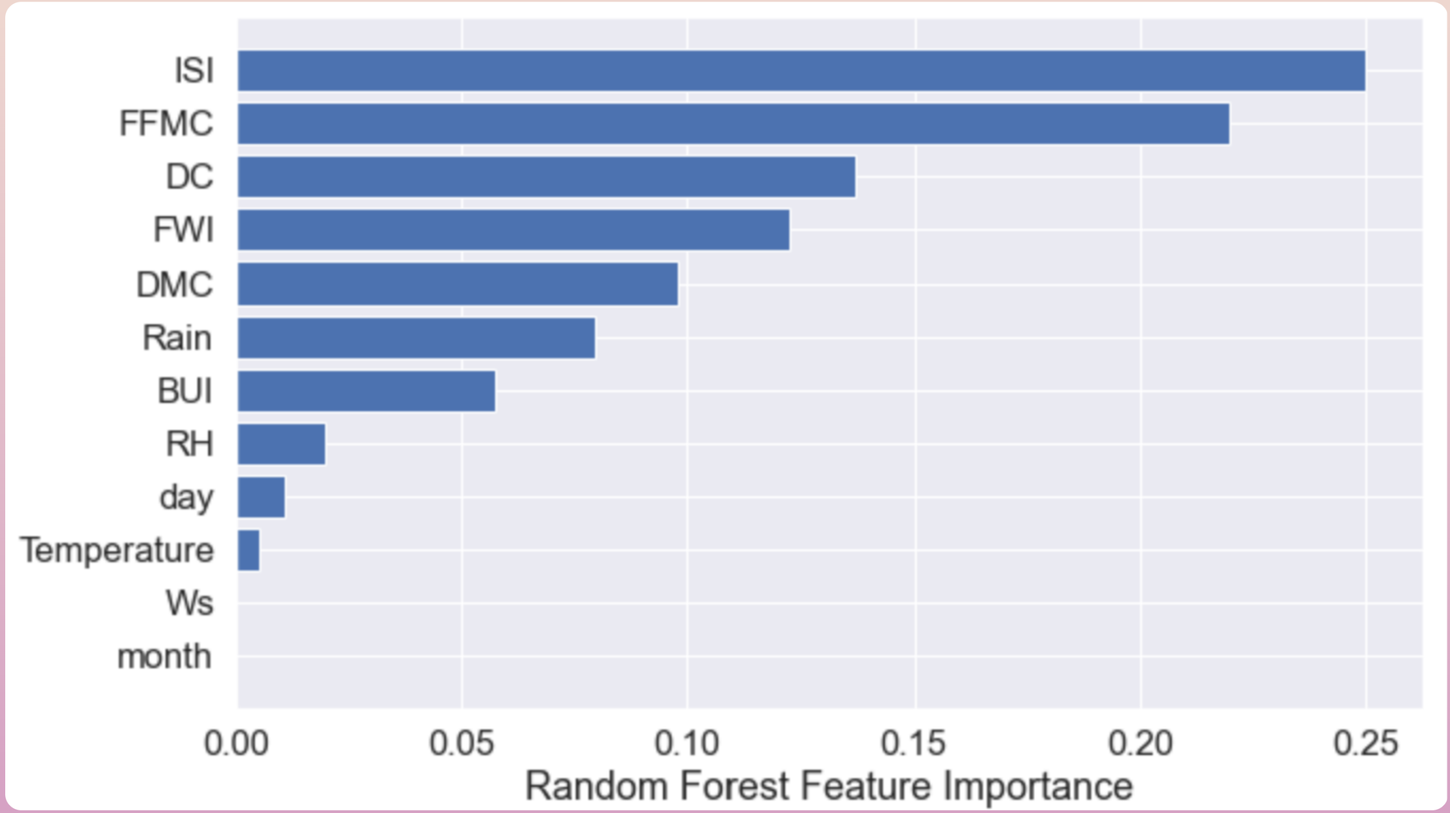
- ***Classification Report:*** A summary of the key classification metrics (precision, recall, F1-score, and support) for each class in the dataset.

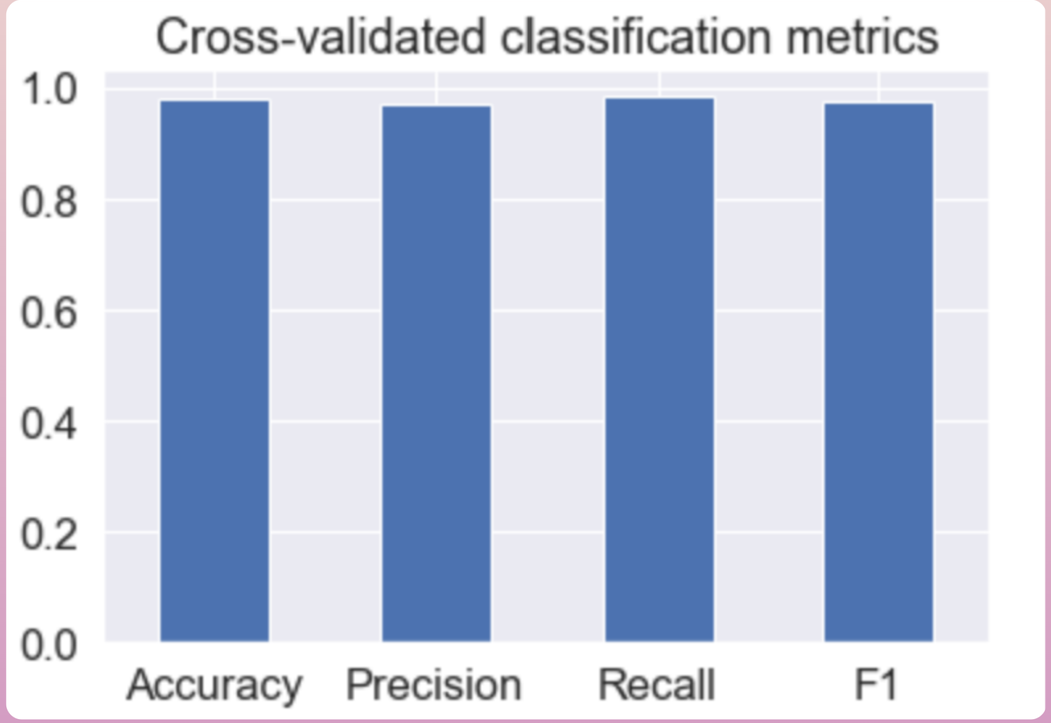
- ***ROC Curve (Receiver Operating Characteristic Curve):*** A graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It helps evaluate the model's performance at various threshold settings.

- ***Area Under the Curve (AUC):*** The area under the ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds.

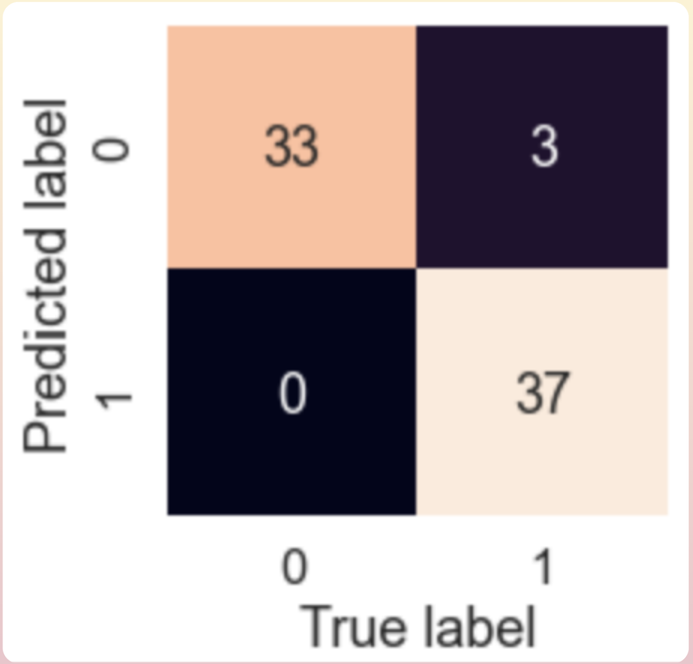
Evaluating Our Tuned Machine Learning

We choose the ***Random forest Classifier*** model and performed the tuning on it





Evaluating Our Tuned Machine Learning



Confusion matrix:

- There are 33 True Positives (predicted forest fires that were actually forest fires).
- There are 37 True Negatives (predicted no forest fires when there were no forest fires).
- There are 3 False Positives (predicted forest fires when there were no forest fires).
- There are 0 False Negatives (predicted no forest fires when there were forest fires).

Agenda

- Introduction to the Project and the Problem Definition
- Packages Used in The Project
- Data Overview and Features of the Used Data
- Data Exploration
- Machine Learning Models
- Evaluating Our Tuned Machine Learning
- **Conclusion** ✅

Conclusion

1. ***Model Performance:*** Our models, including Linear SVC, K-Nearest Neighbours Classifier, and Random Forest Classifier, have demonstrated strong performance in predicting forest fires, achieving an average accuracy of over 95%.
2. ***Key Features:*** Through our analysis, we identified key features that significantly influence the prediction of forest fires, such as temperature, humidity, and wind speed, providing valuable insights for future research and management strategies.
3. ***Model Tuning:*** By tuning hyperparameters using techniques like RandomizedSearchCV and GridSearchCV, we optimized our models, further improving their accuracy and robustness.
4. ***Evaluation Metrics:*** We utilized various evaluation metrics, including precision, recall, and F1 score, to assess the performance of our models, ensuring a comprehensive understanding of their strengths and weaknesses.
5. ***Future Work:*** While our models have shown promising results, there is room for improvement, particularly in incorporating more granular environmental data and exploring advanced modeling techniques to enhance prediction accuracy.

You can find the scource code at the following link:

https://github.com/FoozBarakat/Emerging-ITE646