**CORE-PROJECT**

1. **Comparing with Machine learning algorithms With ANN and SVM.**
2. import numpy as np
3. import pandas as pd
4. from sklearn.model\_selection import train\_test\_split
5. from sklearn.preprocessing import StandardScaler
6. from sklearn.svm import SVC
7. from sklearn.neural\_network import MLPClassifier
8. import matplotlib.pyplot as plt
9. # Load the dataset
10. # Or load from a local CSV file:
11. # data = pd.read\_csv("your\_dataset.csv")
12. data = pd.read\_csv('Project1WeatherDataset.csv')
13. # Data preprocessing
14. X = data[['Temp\_C', 'Dew Point Temp\_C', 'Rel Hum\_%', 'Wind Speed\_km/h', 'Visibility\_km', 'Press\_kPa']]
15. y = data['Wind Speed\_km/h']  # Assuming 'Weather' contains the labels
16. # Split the data into training and testing sets
17. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
18. # Standardize the features
19. scaler = StandardScaler()
20. X\_train = scaler.fit\_transform(X\_train)
21. X\_test = scaler.transform(X\_test)
22. # Train an SVM model
23. svm\_model = SVC()
24. svm\_model.fit(X\_train, y\_train)
25. # Train an ANN model
26. ann\_model = MLPClassifier(hidden\_layer\_sizes=(100, 100), max\_iter=1000)
27. ann\_model.fit(X\_train, y\_train)
28. # Calculate accuracy for both models
29. svm\_accuracy = svm\_model.score(X\_test, y\_test) \* 100
30. ann\_accuracy = ann\_model.score(X\_test, y\_test) \* 100
31. print(f"SVM Model Accuracy: {svm\_accuracy:.2f}%")
32. print(f"ANN Model Accuracy: {ann\_accuracy:.2f}%")
33. # Create a bar chart to visualize the comparison
34. models = ['SVM', 'ANN']
35. accuracies = [svm\_accuracy, ann\_accuracy]
36. plt.bar(models, accuracies)
37. plt.ylabel('Accuracy (%)')
38. plt.title('SVM vs. ANN Weather Forecasting Accuracy')
39. plt.show()

<https://colab.research.google.com/drive/1bAd6A-wJK6Ae4kiWoe7pHZogMs8rcSXC#scrollTo=HQN0JCcJIm6r>

1. **Comparing with Machine learning algorithms With ANN and NWP.**
2. import numpy as np
3. import pandas as pd
4. from sklearn.model\_selection import train\_test\_split
5. from sklearn.neural\_network import MLPClassifier
6. from sklearn.metrics import accuracy\_score
7. import matplotlib.pyplot as plt
8. # Load the dataset
9. # Or load from a local CSV file:
10. # data = pd.read\_csv("your\_dataset.csv")
11. data = pd.read\_csv('/content/Project1WeatherDataset.csv')
12. # Data preprocessing
13. X = data[['Temp\_C', 'Dew Point Temp\_C', 'Rel Hum\_%', 'Wind Speed\_km/h', 'Visibility\_km', 'Press\_kPa']]
14. y = data['Wind Speed\_km/h']  # Assuming 'Weather' contains the labels
15. # Split the data into training and testing sets
16. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
17. # Train an ANN model
18. ann\_model = MLPClassifier(hidden\_layer\_sizes=(100, 100), max\_iter=1000)
19. ann\_model.fit(X\_train, y\_train)
20. # Make predictions using the ANN model
21. ann\_predictions = ann\_model.predict(X\_test)
22. # Simulate a Numerical Weather Prediction model (baseline model)
23. # This is a simplified representation of a baseline model.
24. # You can modify it to reflect your NWP model's behavior.
25. # For simplicity, we'll assume 70% accuracy.
26. baseline\_predictions = np.random.choice(y.unique(), size=len(y\_test))
27. # Calculate accuracy for both models
28. ann\_accuracy = accuracy\_score(y\_test, ann\_predictions) \* 100
29. baseline\_accuracy = accuracy\_score(y\_test, baseline\_predictions) \* 100
30. print(f"ANN Model Accuracy: {ann\_accuracy:.2f}%")
31. print(f"Numerical Weather Prediction Model Accuracy: {baseline\_accuracy:.2f}%")
32. # Create a bar chart to visualize the comparison
33. models = ['ANN', 'NWP Model']
34. accuracies = [ann\_accuracy, baseline\_accuracy]
35. plt.bar(models, accuracies)
36. plt.ylabel('Accuracy (%)')
37. plt.title('ANN vs. NWP Weather Forecasting Accuracy Comparison')
38. plt.show()

<https://colab.research.google.com/drive/1p97YTRe4OAQidIgW51mP6M_abHta8Z->i

1. **Comparing with Machine learning algorithms With ANN and Logistic Regression.**
2. import numpy as np
3. import pandas as pd
4. from sklearn.model\_selection import train\_test\_split
5. from sklearn.linear\_model import LogisticRegression
6. from sklearn.neural\_network import MLPClassifier
7. import matplotlib.pyplot as plt
8. # Load the dataset
9. # Or load from a local CSV file:
10. # data = pd.read\_csv("your\_dataset.csv")
11. data = pd.read\_csv('/content/Project1WeatherDataset.csv')
12. # Data preprocessing
13. X = data[['Temp\_C', 'Dew Point Temp\_C', 'Rel Hum\_%', 'Wind Speed\_km/h', 'Visibility\_km', 'Press\_kPa']]
14. y = data['Wind Speed\_km/h']  # Assuming 'Weather' contains the labels
15. # Split the data into training and testing sets
16. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
17. # Train a Logistic Regression model
18. lr\_model = LogisticRegression(max\_iter=1000)
19. lr\_model.fit(X\_train, y\_train)
20. # Train an ANN model
21. ann\_model = MLPClassifier(hidden\_layer\_sizes=(100, 100), max\_iter=1000)
22. ann\_model.fit(X\_train, y\_train)
23. # Calculate accuracy for both models
24. lr\_accuracy = lr\_model.score(X\_test, y\_test) \* 100
25. ann\_accuracy = ann\_model.score(X\_test, y\_test) \* 100
26. print(f"Logistic Regression Model Accuracy: {lr\_accuracy:.2f}%")
27. print(f"ANN Model Accuracy: {ann\_accuracy:.2f}%")
28. # Create a bar chart to visualize the comparison
29. models = ['Logistic Regression', 'ANN']
30. accuracies = [lr\_accuracy, ann\_accuracy]
31. plt.bar(models, accuracies)
32. plt.ylabel('Accuracy (%)')
33. plt.title('Logistic Regression vs. ANN Weather Forecasting Accuracy')
34. plt.show()

[**https://colab.research.google.com/drive/1Tb8swsYSYrDO0YMSWA5HFsq8R5UlBHv4**](https://colab.research.google.com/drive/1Tb8swsYSYrDO0YMSWA5HFsq8R5UlBHv4)

**4. Comparing with Machine learning algorithms With ANN and AdaBoost.**

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import AdaBoostClassifier

from sklearn.neural\_network import MLPClassifier

import matplotlib.pyplot as plt

# Load the dataset

# Or load from a local CSV file:

# data = pd.read\_csv("your\_dataset.csv")

data = pd.read\_csv('/content/Project1WeatherDataset.csv')

# Data preprocessing

X = data[['Temp\_C', 'Dew Point Temp\_C', 'Rel Hum\_%', 'Wind Speed\_km/h', 'Visibility\_km', 'Press\_kPa']]

y = data['Wind Speed\_km/h']  # Assuming 'Weather' contains the labels

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train an AdaBoost model

adaboost\_model = AdaBoostClassifier(n\_estimators=100, random\_state=42)

adaboost\_model.fit(X\_train, y\_train)

# Train an ANN model

ann\_model = MLPClassifier(hidden\_layer\_sizes=(100, 100), max\_iter=1000)

ann\_model.fit(X\_train, y\_train)

# Calculate accuracy for both models

adaboost\_accuracy = adaboost\_model.score(X\_test, y\_test) \* 100

ann\_accuracy = ann\_model.score(X\_test, y\_test) \* 100

print(f"AdaBoost Model Accuracy: {adaboost\_accuracy:.2f}%")

print(f"ANN Model Accuracy: {ann\_accuracy:.2f}%")

# Create a bar chart to visualize the comparison

models = ['AdaBoost', 'ANN']

accuracies = [adaboost\_accuracy, ann\_accuracy]

plt.bar(models, accuracies)

plt.ylabel('Accuracy (%)')

plt.title('AdaBoost vs. ANN Weather Forecasting Accuracy')

plt.show()

[**https://colab.research.google.com/drive/1A7sbjOlBp2I4FAwXaUGaXgpwM88\_ThKS**](https://colab.research.google.com/drive/1A7sbjOlBp2I4FAwXaUGaXgpwM88_ThKS)