Team-1

Rohan Reddy	23WU0102169
Rishik Reddy	23WU0102149
Amogh Reddy	23WU0102183
Sreepad Bhanu	23WU0102233
Hari Ramaneti	**23WU0102135**

Task 1

HYPOTHESIS SPACE AND INDUCTIVE BIAS, LEARNING TASKS- CLASSIFICATION, REGRESSION

Hypothesis space

The hypothesis space is the set of all possible models or functions a learning algorithm can consider to map inputs to outputs. The size and nature of the hypothesis space determine the algorithm's flexibility and complexity.

Real-Life Case: In email spam detection, the hypothesis space could consist of linear models, decision trees, or neural networks. Each model type represents a different hypothesis space with unique characteristics. For example, linear models might only capture simple relationships, whereas neural networks can represent highly complex mappings.

Inductive Bias

An inductive bias is a set of assumptions a machine learning model makes to learn patterns from data and make predictions. These assumptions help the model generalize to unseen data.

Real-Life Case: In image recognition, convolutional neural networks (CNNs) leverage the inductive bias of spatial hierarchies in images. It is this bias which allows the network to ignore many global patterns, such as textures and edges, to attain fast learning and good classification of images.

References:

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87. https://doi.org/10.1145/2347736.2347755

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Zhang, H., & Ling, C. X. (2018). Data preparation for data mining. *Data Mining and Knowledge Discovery Handbook*, 37-71. https://doi.org/10.1007/978-0-387-09823-4_2

Learning Tasks: Classification and Regression

Classification

Classification is the assignment of inputs to discrete categories. Commonly in classification tasks are algorithms like logistic regression, inside algorithms, and deep learning models.

Real-Life Case: As a classic classification problem, transactions are labeled as "fraudulent" or "non-fraudulent," and the task is to predict the next output label whether this transaction is fraudulent or not.

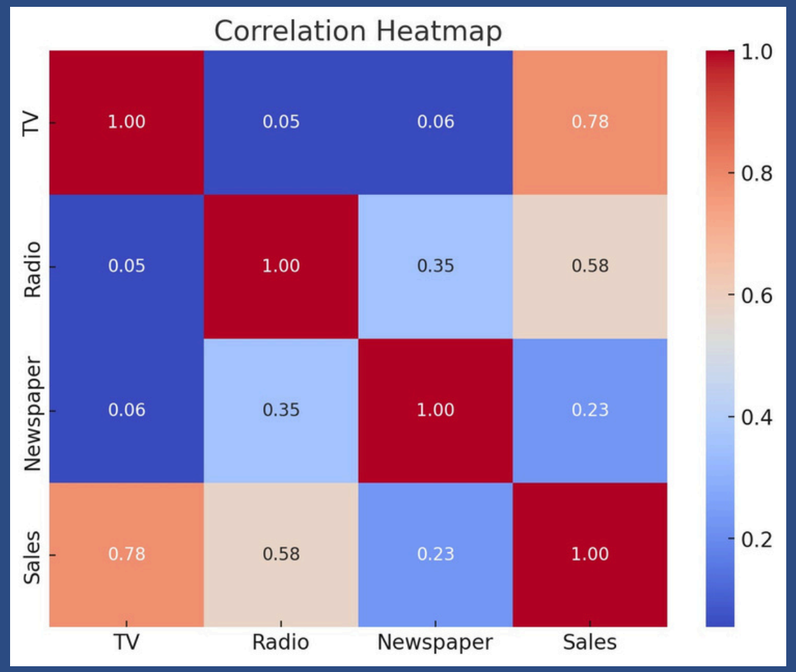
Regression

Regression means predicting continuous numerical values. Linear regression, polynomial regression, gradient boosting methods are used for regression tasks using algorithms.

Real-Life Case: A classic regression problem is predicting house prices. To predict the price they use features such as location, square footage and the number of rooms.

TASK 2

IS THERE A RELATIONSHIP BETWEEN TV COST OF TV ADVERTS AND SALES? WHAT ABOUT RADIO AND SALES? WHAT ABOUT NEWSPAPER AND APPARENT SALES? CAN YOU PREDICT THE APPARENT SALES GIVEN THE GIVEN SPENDS FOR ADVERTS ON TV?



ANALYSIS RESULTS:

CORRELATION ANALYSIS:

THE CORRELATION BETWEEN TV ADS AND SALES IS 0.78, INDICATING A STRONG POSITIVE RELATIONSHIP. THE CORRELATION BETWEEN RADIO ADS AND SALES IS 0.58, SHOWING A MODERATE POSITIVE RELATIONSHIP. THE CORRELATION BETWEEN NEWSPAPER ADS AND SALES IS 0.23, SUGGESTING A WEAK RELATIONSHIP.

PREDICTIVE MODELING (TV ADS AND SALES):

MEAN SQUARED ERROR (MSE): 10.20

R^2 SCORE: 0.68 (68% OF THE VARIANCE IN SALES IS EXPLAINED BY TV AD COSTS).

THE REGRESSION EQUATION IS: SALES = 7.12 + 0.0465 × TV

SALES=7.12+0.0465×TV

THIS MEANS FOR EVERY ADDITIONAL UNIT SPENT ON TV ADS, SALES INCREASE BY APPROXIMATELY 0.0465 UNITS.

TASK 3: TENNIS DATA ANALYSIS

- ## PROBLEM OVERVIEW

  GOAL: WE WANT TO PREDICT WHETHER A PERSON WILL PLAY TENNIS BASED ON CERTAIN WEATHER CONDITIONS.

  DATA: WE HAVE A DATASET WITH WEATHER-RELATED FEATURES (LIKE OUTLOOK, TEMPERATURE, HUMIDITY, AND WINDY) AND A TARGET VARIABLE CLASS THAT INDICATES WHETHER THE PERSON WILL PLAY TENNIS OR NOT (+ FOR PLAY AND – FOR NOT).

- ## MAKING PREDICTIONS

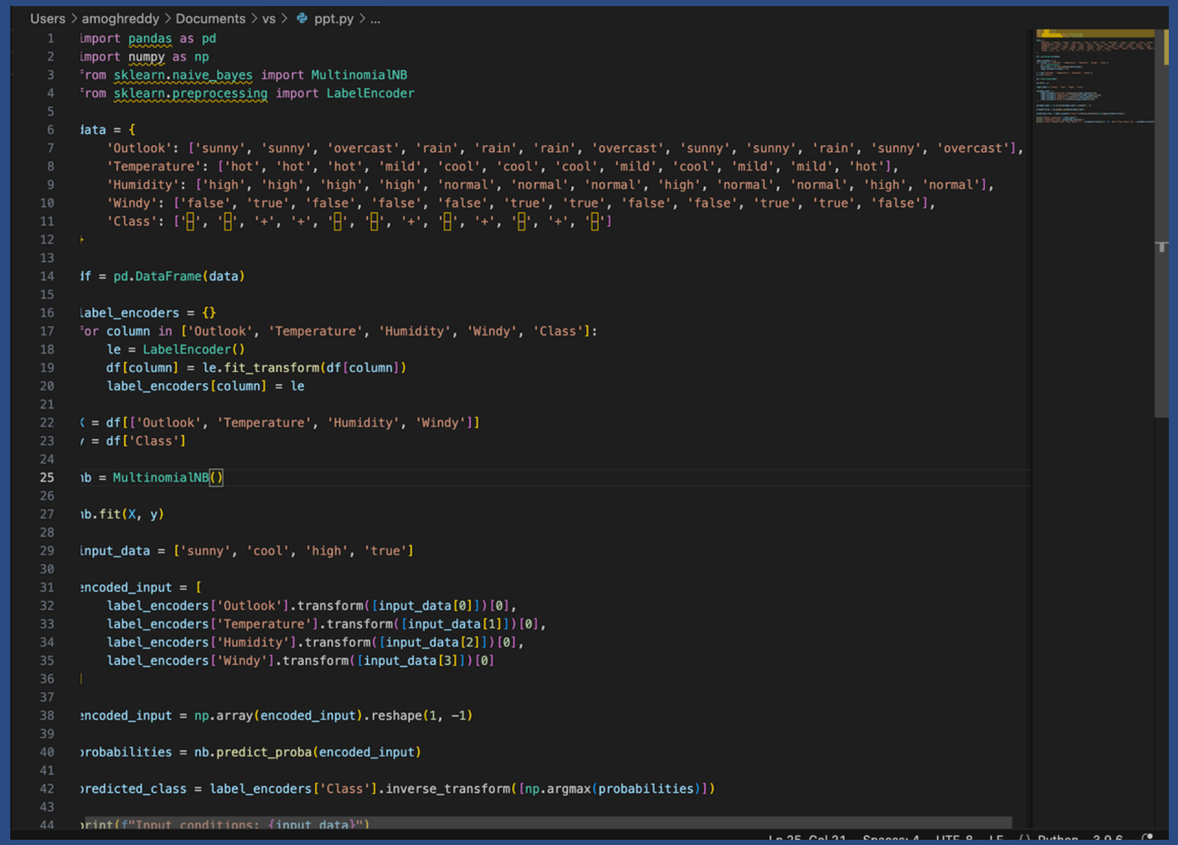
  INPUT DATA: WE TAKE NEW, UNSEEN WEATHER CONDITIONS, LIKE ['SUNNY', 'COOL', 'HIGH', 'TRUE'], WHICH REPRESENT THE CONDITIONS FOR A POTENTIAL TENNIS MATCH.
  DATA TRANSFORMATION: BEFORE PASSING THESE INPUTS TO THE MODEL, WE CONVERT THEM INTO THE SAME NUMERIC FORM AS THE TRAINING DATA.

  PREDICTION: THE MODEL PREDICTS THE PROBABILITY OF TWO POSSIBLE OUTCOMES:
  PLAY TENNIS (+): THE LIKELIHOOD OF PLAYING TENNIS.
  DON’T PLAY TENNIS (–): THE LIKELIHOOD OF NOT PLAYING TENNIS.
  THE MODEL WILL OUTPUT THE PROBABILITIES, AND WE CHOOSE THE OUTCOME WITH THE HIGHEST PROBABILITY AS THE FINAL PREDICTION.

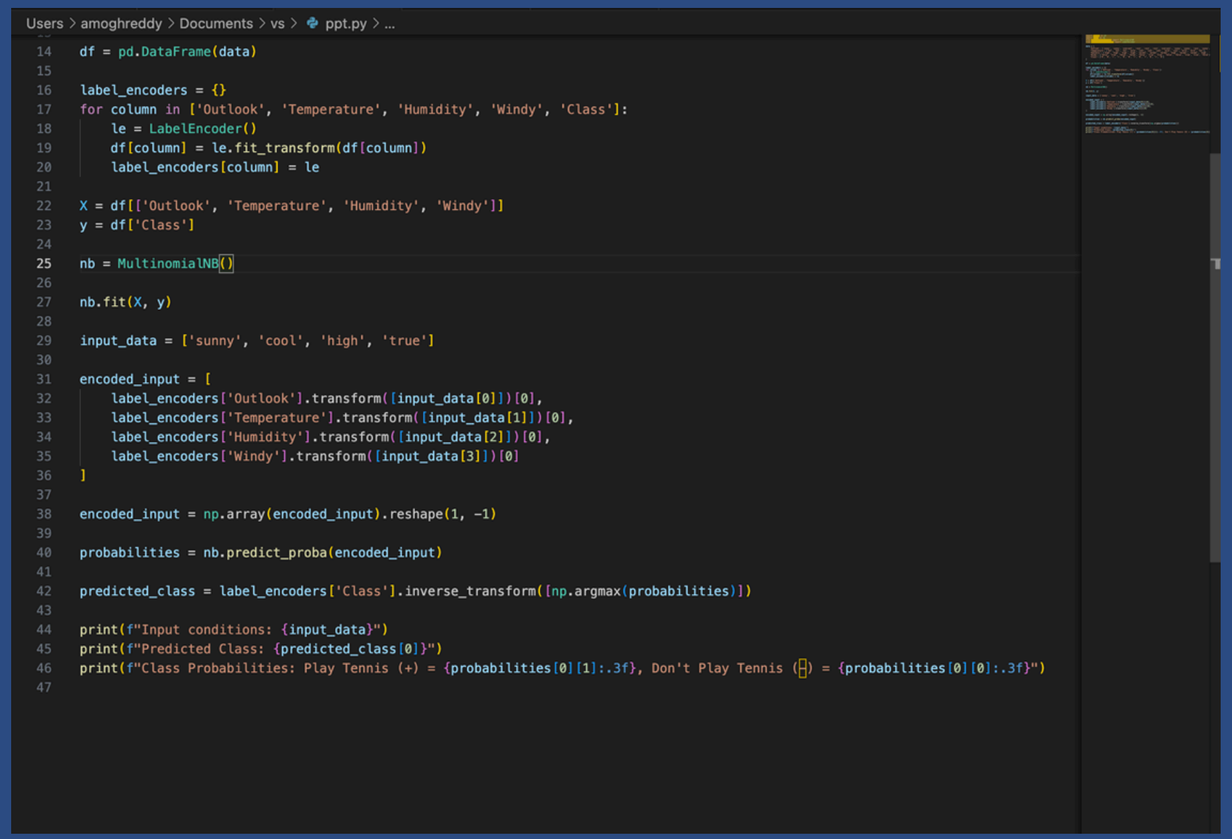
- OUTPUT

CLASS PREDICTION: THE MODEL GIVES US A PREDICTION OF WHETHER TENNIS WILL BE PLAYED (EITHER + OR −).
PROBABILITY VALUES: WE ALSO GET THE LIKELIHOOD OF EACH OUTCOME. FOR EXAMPLE:
"PLAY TENNIS (+)" = 0.606
"DON’T PLAY TENNIS (−)" = 0.394
THESE PROBABILITIES GIVE US A SENSE OF CONFIDENCE IN THE MODEL'S PREDICTION. IN THIS CASE, THE MODEL IS MORE CONFIDENT THAT THE PERSON WILL NOT PLAY TENNIS.

TASK 3: TENNIS DATA ANALYSIS



```
Users > amoghreddy > Documents > vs > ppt.py > ...
import pandas as pd
import numpy as np
from sklearn.naive_bayes import MultinomialNB
from sklearn.preprocessing import LabelEncoder

data = {
    'Outlook': ['sunny', 'sunny', 'overcast', 'rain', 'rain', 'rain', 'overcast', 'sunny', 'sunny', 'rain', 'sunny', 'overcast'],
    'Temperature': ['hot', 'hot', 'hot', 'mild', 'cool', 'cool', 'cool', 'mild', 'cool', 'mild', 'mild', 'hot'],
    'Humidity': ['high', 'high', 'high', 'high', 'normal', 'normal', 'normal', 'high', 'normal', 'normal', 'high', 'normal'],
    'Windy': ['false', 'true', 'false', 'false', 'false', 'true', 'true', 'false', 'false', 'true', 'true', 'false'],
    'Class': ['-', '-', '+', '+', '-', '-', '+', '-', '+', '-', '+', '-']
}

df = pd.DataFrame(data)

label_encoders = {}
for column in ['Outlook', 'Temperature', 'Humidity', 'Windy', 'Class']:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

X = df[['Outlook', 'Temperature', 'Humidity', 'Windy']]
y = df['Class']

nb = MultinomialNB()

nb.fit(X, y)

input_data = ['sunny', 'cool', 'high', 'true']

encoded_input = [
    label_encoders['Outlook'].transform([input_data[0]])[0],
    label_encoders['Temperature'].transform([input_data[1]])[0],
    label_encoders['Humidity'].transform([input_data[2]])[0],
    label_encoders['Windy'].transform([input_data[3]])[0]
]

encoded_input = np.array(encoded_input).reshape(1, -1)

probabilities = nb.predict_proba(encoded_input)

predicted_class = label_encoders['Class'].inverse_transform([np.argmax(probabilities)])

print(f"Input conditions: {input_data}")
```



```
Users > amoghreddy > Documents > vs > ppt.py > ...
df = pd.DataFrame(data)

label_encoders = {}
for column in ['Outlook', 'Temperature', 'Humidity', 'Windy', 'Class']:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

X = df[['Outlook', 'Temperature', 'Humidity', 'Windy']]
y = df['Class']

nb = MultinomialNB()

nb.fit(X, y)

input_data = ['sunny', 'cool', 'high', 'true']

encoded_input = [
    label_encoders['Outlook'].transform([input_data[0]])[0],
    label_encoders['Temperature'].transform([input_data[1]])[0],
    label_encoders['Humidity'].transform([input_data[2]])[0],
    label_encoders['Windy'].transform([input_data[3]])[0]
]

encoded_input = np.array(encoded_input).reshape(1, -1)

probabilities = nb.predict_proba(encoded_input)

predicted_class = label_encoders['Class'].inverse_transform([np.argmax(probabilities)])

print(f"Input conditions: {input_data}")
print(f"Predicted Class: {predicted_class[0]}")
print(f"Class Probabilities: Play Tennis (+) = {probabilities[0][1]:.3f}, Don't Play Tennis (-) = {probabilities[0][0]:.3f}")
```

Dataset

Outlook	Temperature	Humidity	Windy	Class
2	1	0	0	–
2	1	0	1	–
0	1	0	0	+
1	2	0	0	+
1	0	1	0	–
1	0	1	1	–
0	0	1	1	+
2	2	0	0	–
2	0	1	0	+
1	2	1	1	–
2	2	0	1	+
0	1	1	0	–

```
: X does not have valid feature names, but MultinomialNB was fitted with feature names
  warnings.warn(
Input conditions: ['sunny', 'cool', 'high', 'true']
Predicted Class: -
Class Probabilities: Play Tennis (+) = 0.606, Don't Play Tennis (-) = 0.394
```

Task 4

Problem Statement:

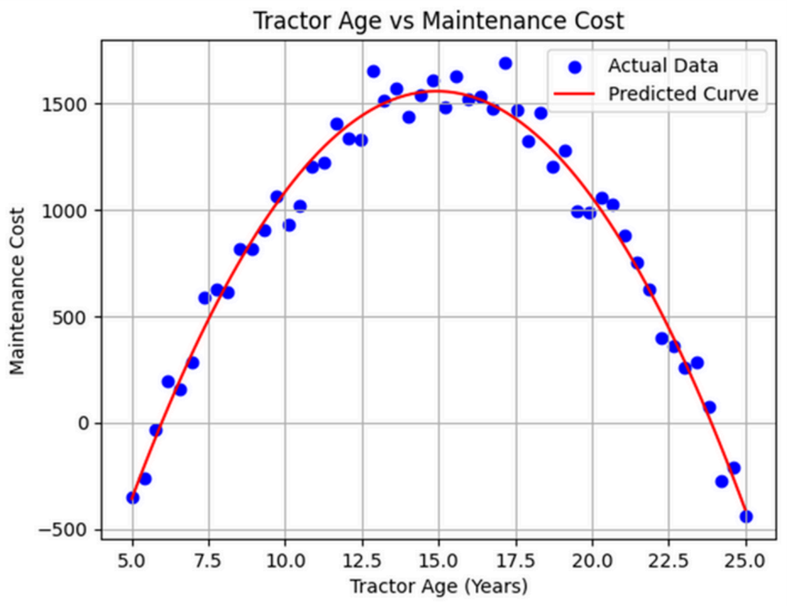
A farm operations student studies the relationship between the maintenance cost and the age of farm tractors in potato farms

---

Observations

- Maintenace cost increases with tractor age upto 15 years
- After peaking , the cost declines as the tractor ages

Approach



Data Collection

↓

Data Analysis

↓

Regression model

↓

Optimize Strategies

This will determine the optimal replacement period for a tractor to maximize its cost efficiency.

Thank you