



SMART
CONSULTANCY

MACHINE LEARNING

ASSIGNMENT 2

Team:

R.Mallikarjun Reddy

Pyla Gagan

K.Himavanth Sai Sujana

Manikeshwar Reddy

Lalith Lavu

TASK 1:

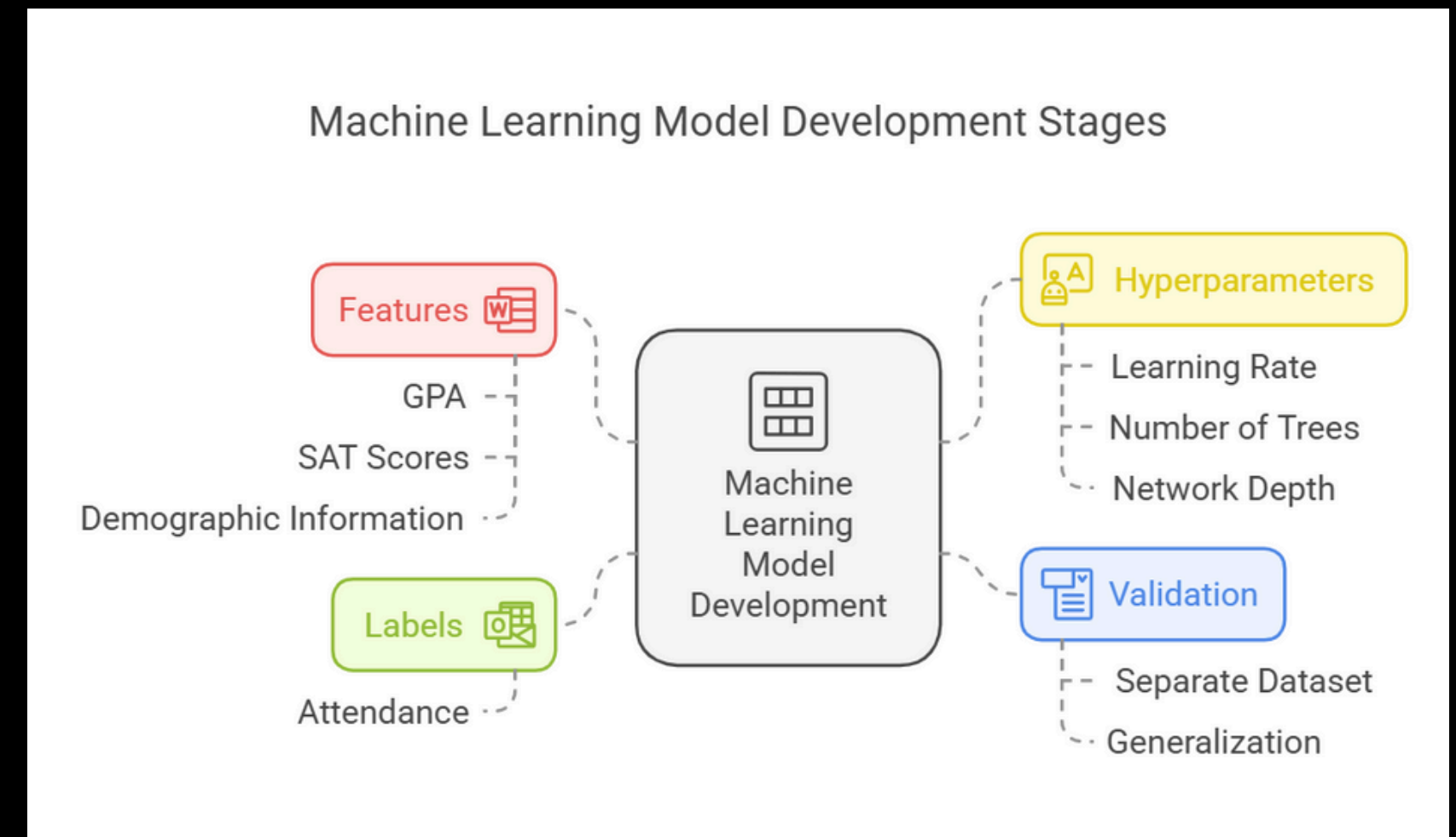
Learning Stages in ML

Features: Models use features as independent variables for production determination. Feature selection processes coupled with engineering techniques provide substantial benefits to model predictive abilities.

Labels :The direction my model aims to determine consists of Labels for prediction output. The model accuracy of supervised learning depends strongly on the quality of provided labels. Achieving both accurate and consistent labeling conditions fundamental for creating effective models.

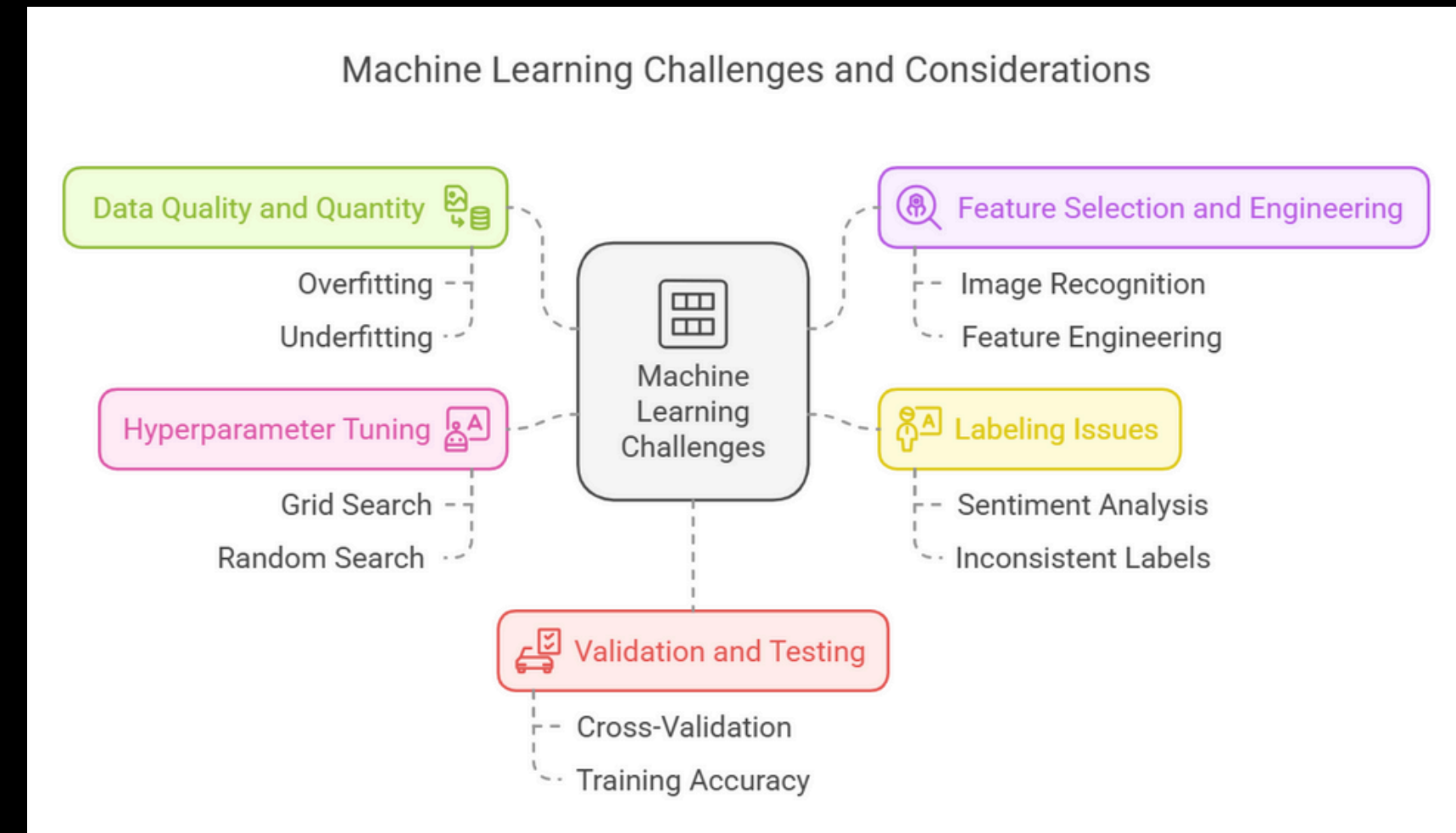
Hyperparameters: Hyperparameters represent adjustable configuration settings which programmers adjust before starting the learning process. Training becomes vulnerable to their influence through controlled parameters because these settings regulate the entire training timetable.

Validation: Model performance evaluation through validation occurs by using the process to assess models against unknown data. Avoiding overfitting requires appropriate validation approaches in decision-making processes.

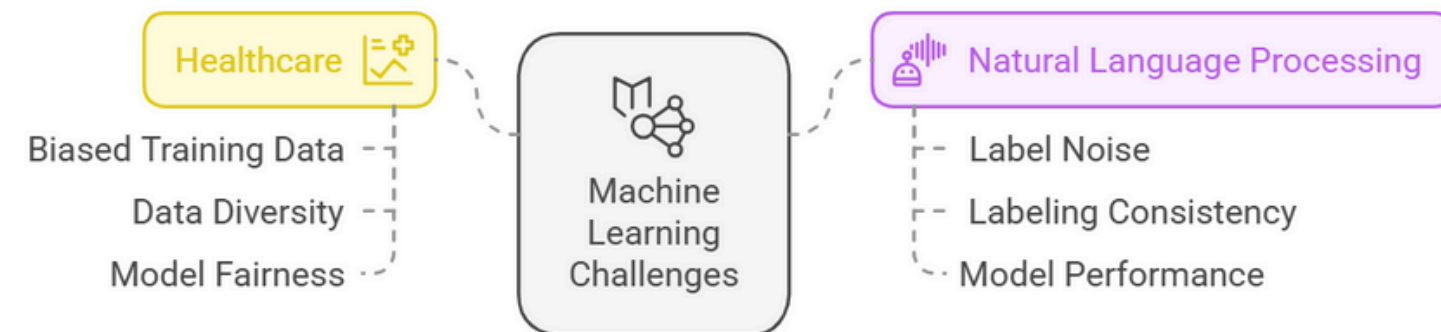


Issues in Machine Learning

- Data Quality and Quantity: Poor-quality or insufficient data leads to biased, overfit, or underfit models, limiting generalization (e.g., non-diverse healthcare datasets).
- Feature Selection: Irrelevant features harm model accuracy; automated systems must refine predictive elements (e.g., color, texture in images).
- Hyperparameter Tuning: Optimizing hyperparameters is resource-intensive and complex due to computational constraints.
- Overfitting and Underfitting: Overfitting occurs when a model captures noise and inaccuracies from a large dataset, adversely affecting its performance. Conversely, underfitting arises from a model being too simple for the data, resulting in incomplete and inaccurate predictions.



Addressing Challenges in Machine Learning Domains



Real-Life Cases

Healthcare: According to Obermeyer et al. (2019) machine learning encounters significant obstacles when used in healthcare because training data proves to be biased. New methods were developed to make data more diverse while achieving fair models.

Natural Language Processing: Gururangan et al. (2018) investigated how label noise affects sentiment analysis. The study examined methods that could boost model performance through better label consistency techniques.

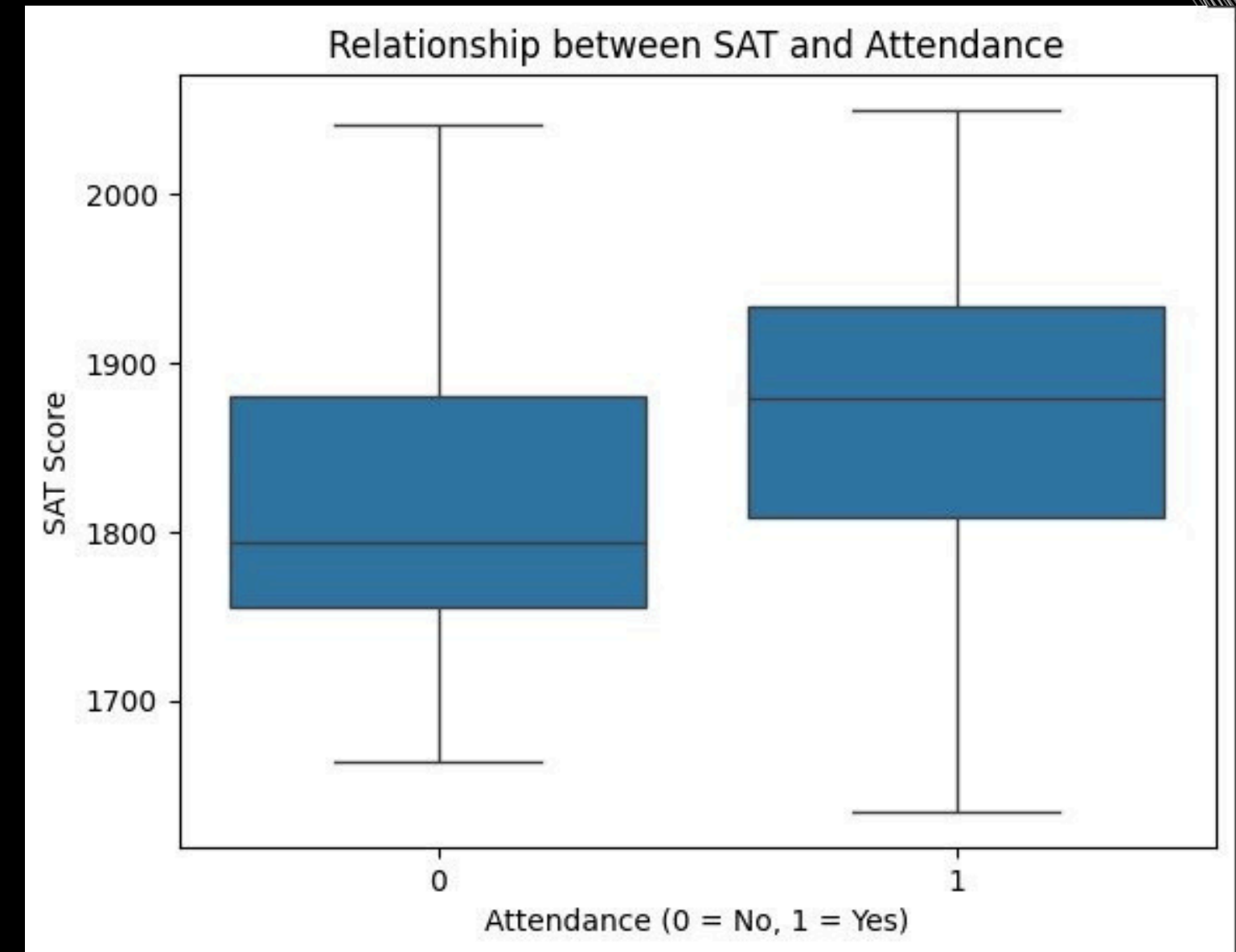
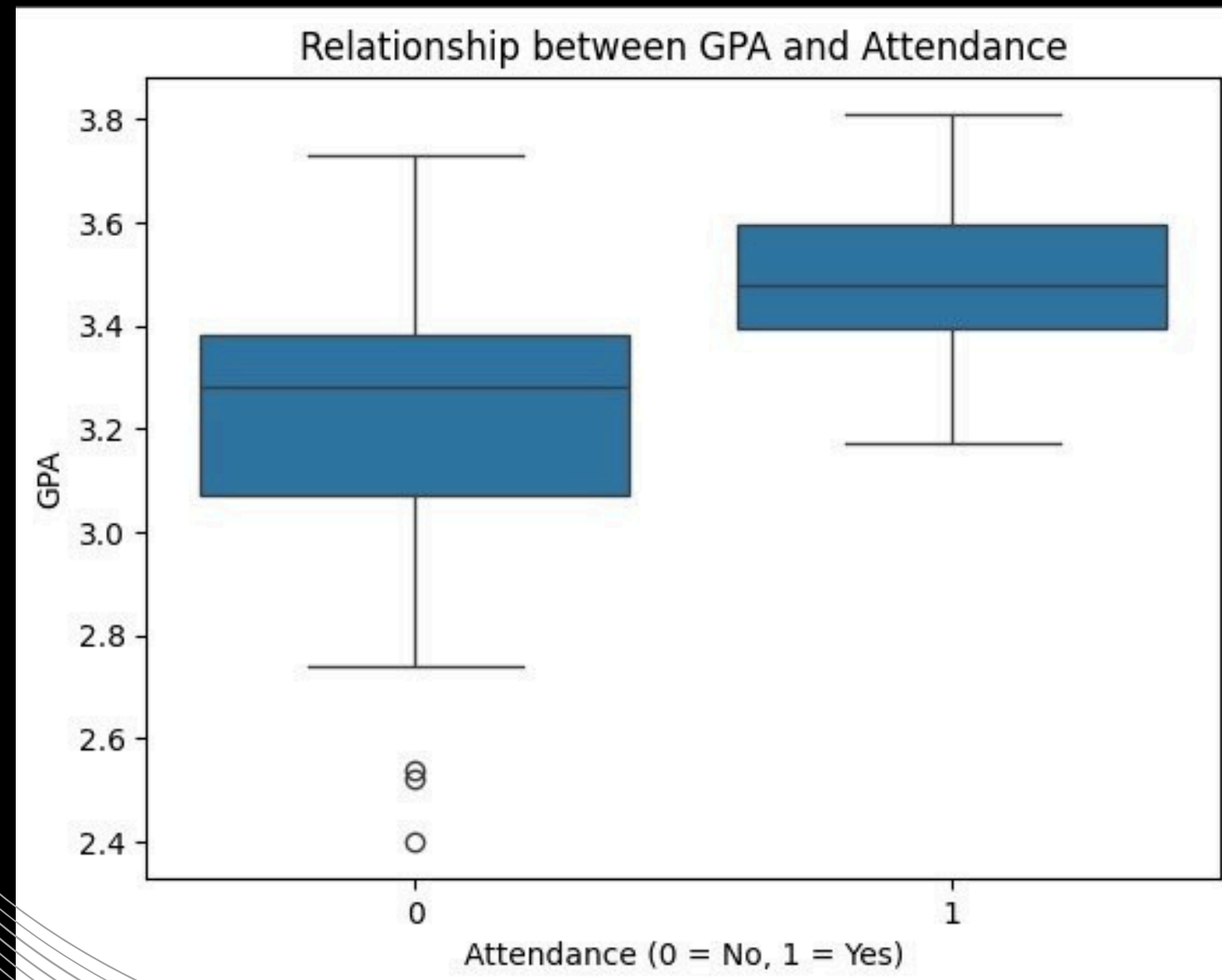
References

Gururangan, S., Marasović, A., Swayamdipta, S., et al. (2018). Annotation artifacts in natural language inference data. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, 1-6.

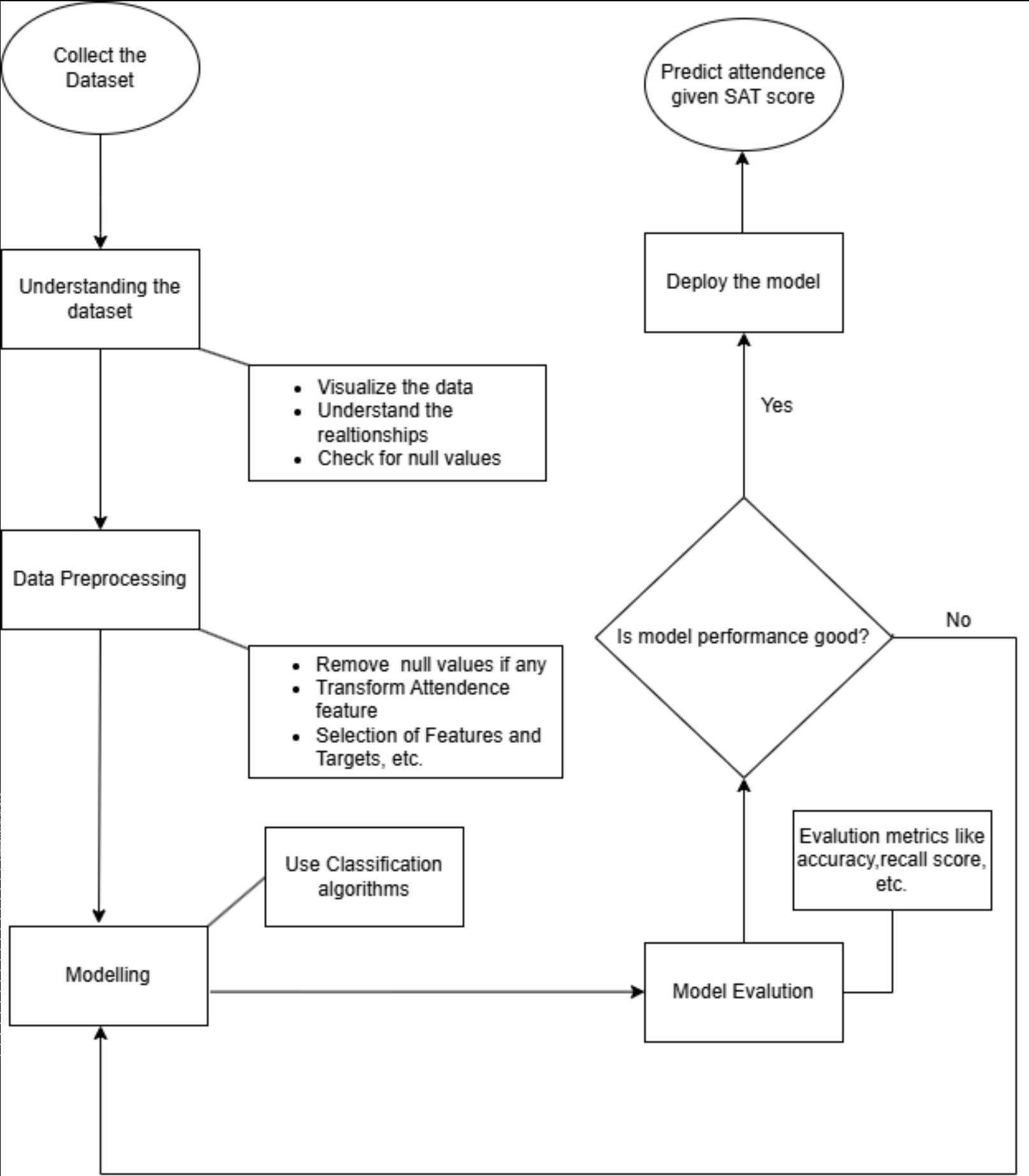
Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464), 447-453. <https://doi.org/10.1126/science.aax2342>

TASK 2:

- Is there a relationship between GPA and attendance?
- What about between SAT and attendance?
- Can you predict the attendance given the SAT score?



TASK 2:



Decision Tree Model Accuracy: 0.88

Sample Predictions with Decision Tree:

	SAT	Attendance	DT Predicted Attendance
0	1714	0	0
1	1664	0	0
2	1760	0	0
3	1685	0	0
4	1693	0	0
..
79	1936	1	1
80	1810	1	1
81	1987	0	1
82	1962	1	1
83	2050	1	1

[84 rows x 3 columns]

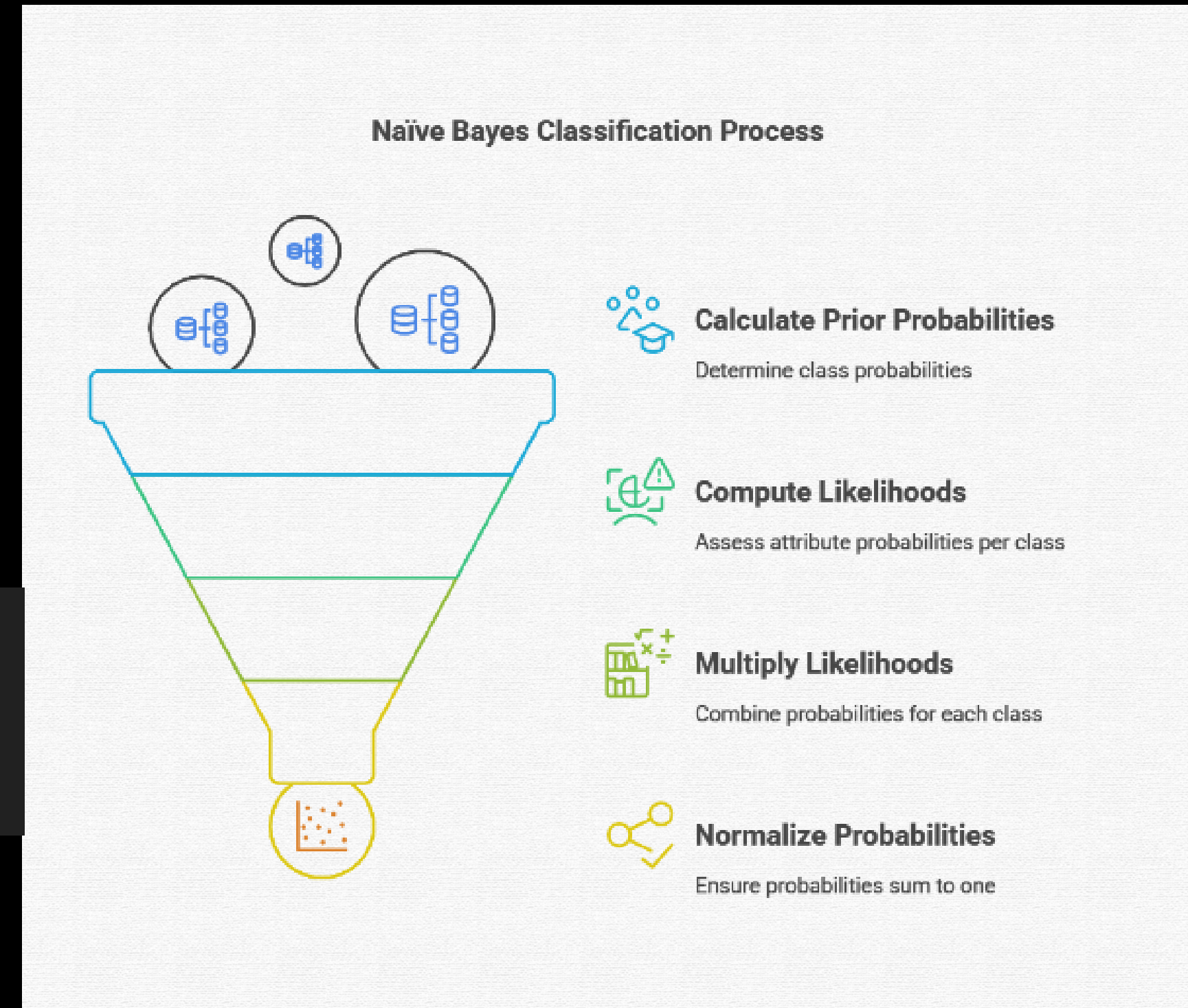
TASK 3:

What probability should be used to make the prediction?

The probability used to make a prediction is the posterior probability for each class. The class with the highest posterior probability is chosen as the prediction. Naive Bayes uses the Bayes theorem to compute these probabilities.

Bayes Theorem:

$$P(Class|Data) = \frac{P(Class) \cdot P(Data|Class)}{P(Data)}$$



```

from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import CategoricalNB
import numpy as np

# Dataset
data = [
    ['sunny', 'hot', 'high', False, '-'],
    ['sunny', 'hot', 'high', True, '-'],
    ['overcast', 'hot', 'high', False, '+'],
    ['rain', 'mild', 'high', False, '+'],
    ['rain', 'cool', 'normal', False, '+'],
    ['rain', 'cool', 'normal', True, '-'],
    ['overcast', 'cool', 'normal', True, '+'],
    ['sunny', 'mild', 'high', False, '-'],
    ['sunny', 'cool', 'normal', False, '+'],
    ['rain', 'mild', 'normal', False, '+'],
    ['sunny', 'mild', 'normal', True, '+'],
    ['overcast', 'mild', 'high', True, '+'],
    ['overcast', 'hot', 'normal', False, '+'],
    ['rain', 'mild', 'high', True, '-']
]

le = LabelEncoder()
data = np.array(data)
for col in range(data.shape[1] - 1):
    data[:, col] = le.fit_transform(data[:, col])

data[:, -1] = le.fit_transform(data[:, -1]) # '+' -> 1, '-' -> 0

X = data[:, :-1].astype(int)
y = data[:, -1].astype(int)

model = CategoricalNB()
model.fit(X, y)

# (sunny, cool, high, strong -> [2, 1, 0, 1])
new_data = np.array([[2, 1, 0, 1]])

probabilities = model.predict_proba(new_data)
predicted_class = model.predict(new_data)[0]
print("Class Probabilities:")
for i, prob in enumerate(probabilities[0]):
    print(f"Class {i} ({'+' if i == 1 else '-'}) : {prob:.4f}")

print(f"\nPredicted Class: ({'+' if predicted_class == 1 else '-'})")

```

OUTPUT:

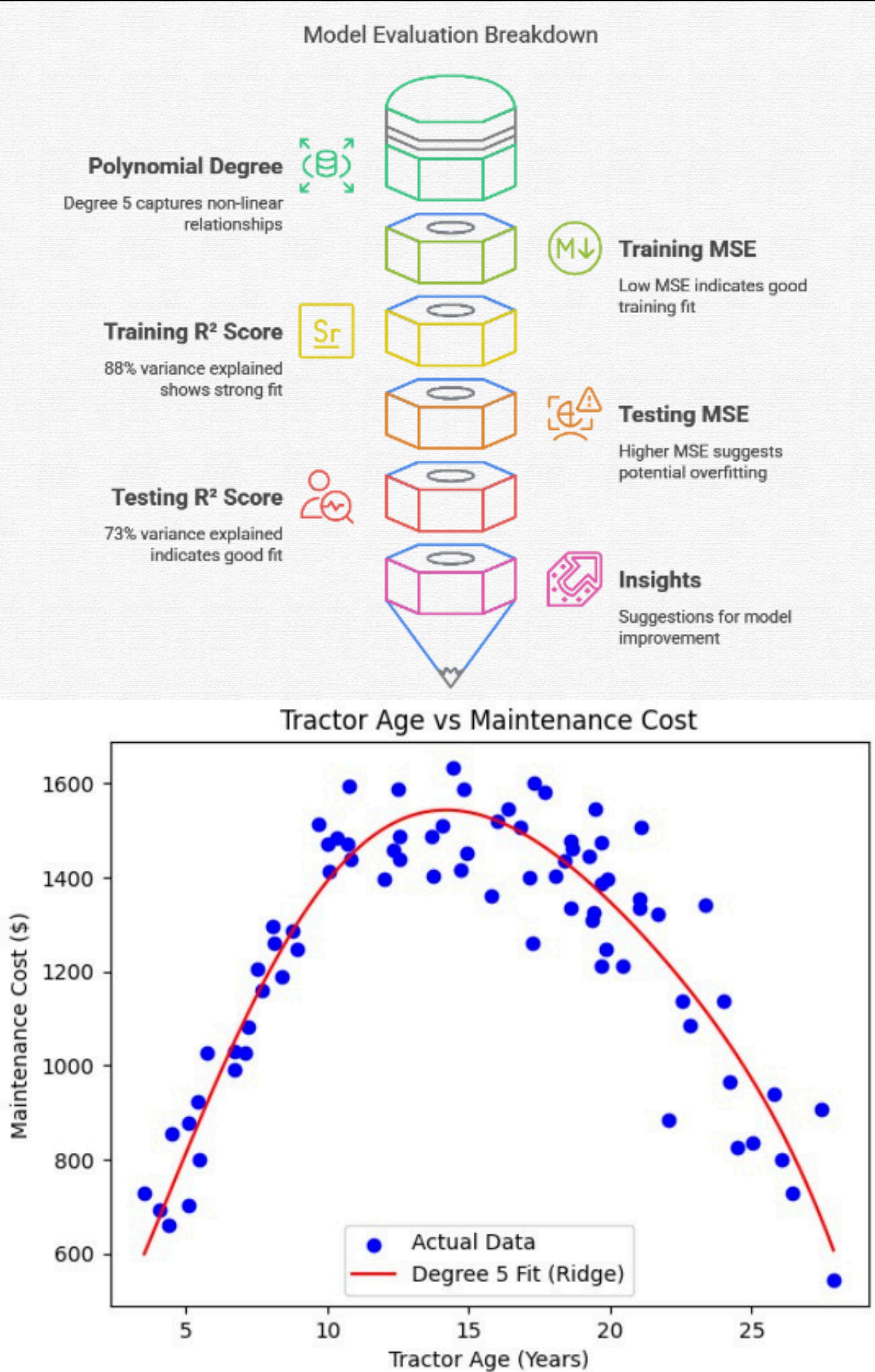
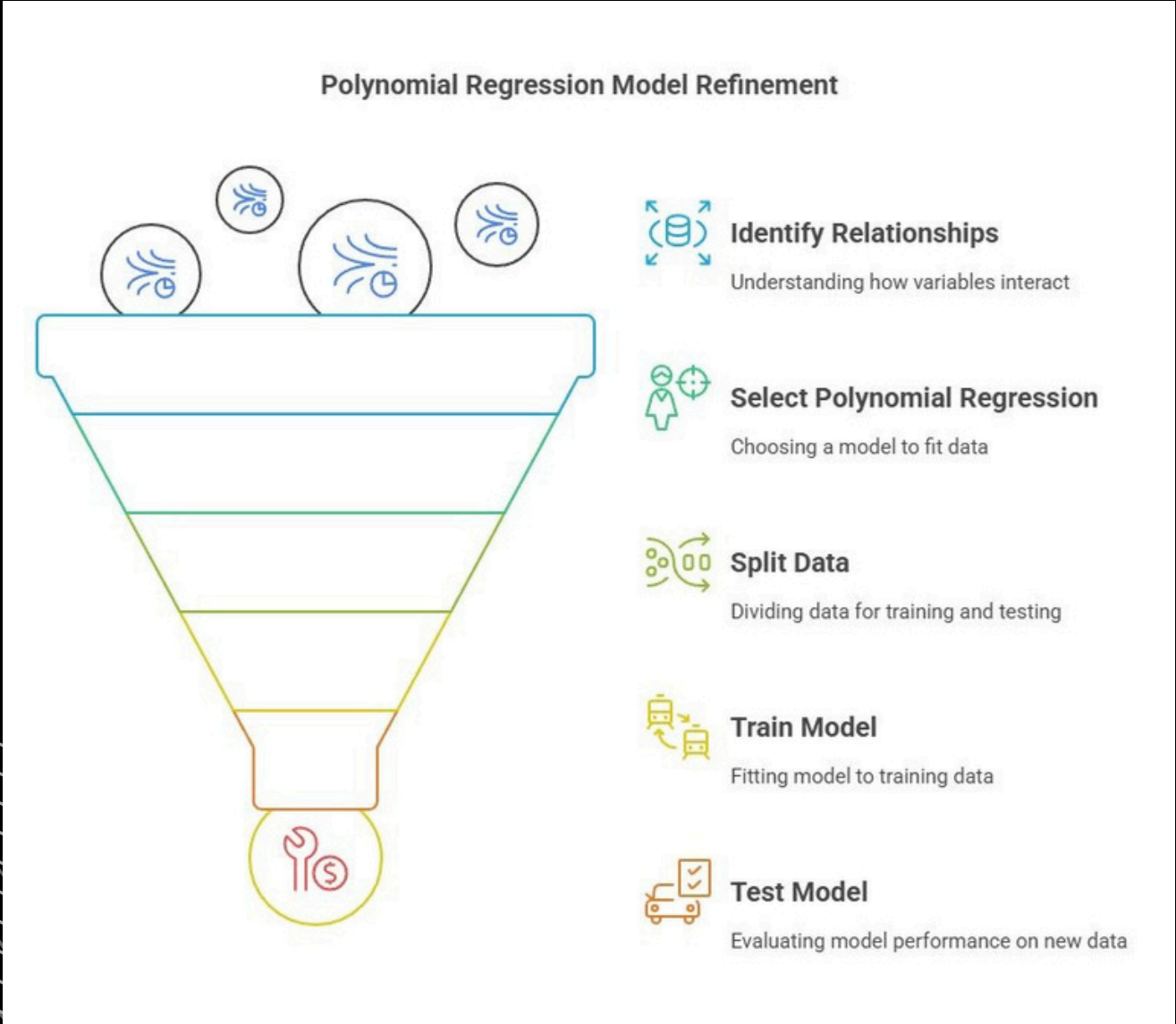
Class Probabilities:

Class 0 (-): 0.1627

Class 1 (+): 0.8373

Predicted Class: (+)

TASK 4:



Thank You