KHAJA BANDANAWAZ UNIVERSITY

A

AAA ML ACTIVITY

ON

“MINIMUM LENGTH DESCRIPTION(MDL) PRINCIPLE”

An ML Activity submitted to Khaja Bandanawaz University, Kalaburagi,

In partial fulfillment of the requirements for the award of the degree of,

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Aliza Mahvash	UIN: 21KB02BS[006]
Afrah Ruheen	UIN: 21KB02BS[005]
Afifa Saher	UIN: 21KB02BS[004]

Under the guidance of

Dr. Shameem Akhter

Ph.D

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
FACULTY OF ENGINEERING & TECHNOLOGY,
KHAJA BANDANAWAZ UNIVERSITY
Kalaburagi-585104
2024-25

TABLE OF CONTENTS:

1. INTRODUCTION

In machine learning, selecting an appropriate model to fit data is crucial. Overfitting occurs when a model is overly complex, fitting the training data perfectly but generalizing poorly to new data. On the other hand, underfitting occurs when a model is too simple, failing to capture important patterns. The **Minimum Description Length (MDL)** principle provides a formal method for model selection by balancing model complexity and data fit.

This work explores the application of MDL to compare **linear regression** and **polynomial regression** models. The aim is to determine which model best fits a synthetic quadratic dataset by using MDL as a selection criterion.

Objective:

To implement a model selection approach based on the **Minimum Description Length (MDL) Principle** and apply it to linear and polynomial regression models for a given synthetic dataset.

2. OVERVIEW

The **Minimum Description Length (MDL) principle** is a formal method for model selection, grounded in information theory. It aims to find the model that compresses data the most by minimizing the combined length of (1) the model's description (complexity) and (2) the data fit (how well the model explains the data). MDL avoids overfitting by penalizing overly complex models, balancing simplicity with accuracy. It's often used as an objective criterion to select between models like linear or polynomial regression.

3. ALGORITHM

The Minimum Description Length principle is motivated by interpreting the definition of h_{MAP} in the light of basic concepts from information theory. Consider again the now familiar definition of h_{MAP}.

$$h_{MAP} = \text{argmax}_{h€H} P(D \mid h) P(h)$$

which can be equivalently expressed in terms of maximizing the $\log_2$

$$h_{MAP} = \text{argmax}_{h€H} \log_2 P(D \mid h) + \log_2 P(h)$$

or alternatively, minimizing the negative of this quantity

$$h_{MAP} = \text{argmin}_{h€H} - \log_2 P(D \mid h) – \log_2 P(h) \ldots\ldots\ldots\ldots(1)$$

Let us interpret the Equation(1) in light of the above result from coding theory.

- $-\log_2 P(h)$ is the description length of h under the optimal encoding for the hypothesis space H. In other words, this is the size of the description of hypothesis h using this optimal representation. In our notation, $L_{CH} (h) = - \log_2 P(h)$, where C_H is the optimal code for hypothesis space H.
- -log2 P(D | h) is the description length of the training data D given hypothesis h, under its optimal encoding. In our notation, $L_{C(D \mid h)} = - \log, P(D \mid h)$, where $C_{D|h}$ is the optimal code for describing data D assuming that both the sender and receiver know the hypothesis h.
- Therefore we can rewrite Equation (1) to show that h_{MAP} is the hypothesis h that minimizes the sum given by the description length of the hypothesis plus the description length of the data given the hypothesis.

$$h_{MAP} = \text{argmin}_{h€H} L_{CH} (h) + L_{C\ D|H} (D \mid h)$$

where C_H and $C_{D \mid h}$ are the optimal encodings for H and for D given h, respectively.

The Minimum Description Length (MDL) principle recommends choosing the hypothesis that minimizes the sum of these two description lengths. Of course to apply this principle in practice we must choose specific encodings or representations appropriate for the given learning task. Assuming we use the codes C_1 and C_2 to represent the hypothesis and the data given the hypothesis, we can state the MDL principle as

Minimum Description Length principle: Choose h_{MDL} where

$$h_{MAP} = \text{argmin}_{h€H} L_{C1} (h) + L_{C2} (D \mid h)$$

The above analysis shows that if we choose C_1 to be the optimal encoding of hypotheses C_H, and if we choose C_2 to be the optimal encoding $C_{D \mid h}$, then $h_{MDL} = h_{MAP}$.

3.1 Applying Minimum Description Length (MDL) Principle to the existing program:

The **MDL principle** comes from information theory and balances two competing factors in model selection:

- **Model Complexity**: The number of parameters in the model. A more complex model can capture more detailed patterns but risks overfitting.
- **Data Fit**: How well the model explains or fits the observed data. The error in prediction (e.g., Mean Squared Error) is a measure of this fit.

Formally, MDL can be expressed as:

$$\text{MDL(M)} = \text{L (Model)} + \text{L(Data} \mid \text{Model)}$$

- L (Model) represents the cost (in terms of bits) of encoding the model, proportional to the number of parameters in the model.
- L (Data|Model) is the cost of encoding the data given the model, which is often approximated by the model's error on unseen data.

In this program, the **number of model parameters** serves as a proxy for model complexity, and the **Mean Squared Error (MSE)** serves as a proxy for the data fit term.

3.2 Linear vs Polynomial Regression:

- **Linear Regression** models the relationship between input XXX and output yyy as a straight line:

  $$y = w_1 X + w_0$$

  where w_1 is the slope and w_0 is the intercept.

- **Polynomial Regression** models a higher degree relationship:

  $$y=w_2 X^2 + w_1 X + w_0$$

  where the model includes a quadratic term to better capture non-linear relationships.

4. METHODOLOGY

4.1 Data Generation:

A synthetic quadratic dataset was generated, simulating a polynomial relationship between input X and output y:

$$y = 3X^2 + 2X + 1 + \epsilon$$

where ϵ is random noise to make the problem more realistic.

4.2 Model Training:

1. **Linear Regression Model**: A simple linear model was trained on the data.
2. **Polynomial Regression Model**: A polynomial model of degree 2 was trained on the same dataset.

4.3 MDL Calculation:

For each model, the **MDL** was calculated as:

MDL=Number of Parameters + Mean Squared Error (MSE)

- For **Linear Regression**, the number of parameters is 2 (slope and intercept).
- For **Polynomial Regression**, the number of parameters is 3 (quadratic term, linear term, and intercept).

5. MODEL SELECTION WITH MDL

The model with the smaller MDL value was selected as the better model, balancing complexity and data fit.

1) Linear Regression Model (Model A): A simple model with lower complexity but potentially poorer fit.

2) Polynomial Regression Model (Model B): A more complex model with a potentially better fit.

The following steps will be taken for model selection using MDL:

1. Fit Models: Fit both a linear regression model and a polynomial regression model to the data.
2. Calculate MDL for Each Model:

    - Model Complexity: We'll use the number of parameters as a proxy for model complexity.
    - Data Fit: We'll use the negative log-likelihood (or a related measure like mean squared error) to assess the fit.

3. Select the Best Model: The model with the lower MDL is preferred.

6. WORKING:

Step 1: Split the data into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=0)
```

Step 2: Train the Linear Regression Model

```
lin_reg = LinearRegression()

lin_reg.fit(X_train, y_train)

y_pred_lin = lin_reg.predict(X_test)

mse_lin = mean_squared_error(y_test, y_pred_lin)
```

MDL for Linear Regression: 2 parameters + MSE

```
mdl_lin = 2 + mse_lin
```

Step 3: Train the Polynomial Regression Model

```
poly_features = PolynomialFeatures(degree=degree, include_bias=False)

X_poly_train = poly_features.fit_transform(X_train)

X_poly_test = poly_features.transform(X_test)

poly_reg = LinearRegression()

poly_reg.fit(X_poly_train, y_train)

y_pred_poly = poly_reg.predict(X_poly_test)

mse_poly = mean_squared_error(y_test, y_pred_poly)
```

MDL for Polynomial Regression: (degree + 1) parameters + MSE

```
mdl_poly = degree + 1 + mse_poly
```

Step 4: Compare MDL values and choose the model with the smallest MDL

```
best_model = 'Linear' if mdl_lin < mdl_poly else f'Polynomial Degree {degree}'

return {

    'MDL_Linear': mdl_lin,

    'MDL_Polynomial': mdl_poly,

    'Best_Model': best_model
    }
```

Step 5: Generate synthetic data:

```
np.random.seed(0)

X = np.random.rand(100, 1) * 10  # Random data points

y = 3 * X**2 + 2 * X + 1 + np.random.randn(100, 1) * 10            ( Quadratic equation with noise )
```

Step 6: MDL Selection:

```
mdl_results = mdl_selection(X, y)

print(mdl_results)
```

7. ANALYSIS OF THE OUTPUT

Polynomial Regression Model (Lower MDL):

With an MDL of **99.033146** , the polynomial model has a significantly lower description length compared to the linear model. This suggests that, despite being more complex (having more parameters), it does a much better job of fitting the data.

The lower MDL value indicates that the polynomial model balances the trade-off between model complexity and data fit more effectively for this dataset.

As per the plot given below, although more complex, the polynomial model's better fit to the data results in a lower description length, compensating for its higher complexity.

Linear Regression Model (Higher MDL):

The MDL of **728.459033** for the linear model is much higher, indicating that while it is simpler (fewer parameters), it does not fit the data as well.

The higher MDL value suggests that the simplicity of the linear model does not compensate for its poorer fit to the data.

As per the plot given below, despite its simplicity (fewer parameters), the linear model's inability to closely fit the data leads to a higher description length (MDL), as indicated by its larger MSE.

The following is output by executing the code.

```
(728.459033273704 ,  99.033146907341)
```

The above output represents the Minimum Description Length (MDL) values for the two models we considered: the linear regression model (Model A) and the polynomial regression model (Model B). The values are:

- **Linear Regression Model (Model A):** MDL = 728.459033273704
- **Polynomial Regression Model (Model B):** MDL = 99.033146907341

The model prediction output can be plotted as:

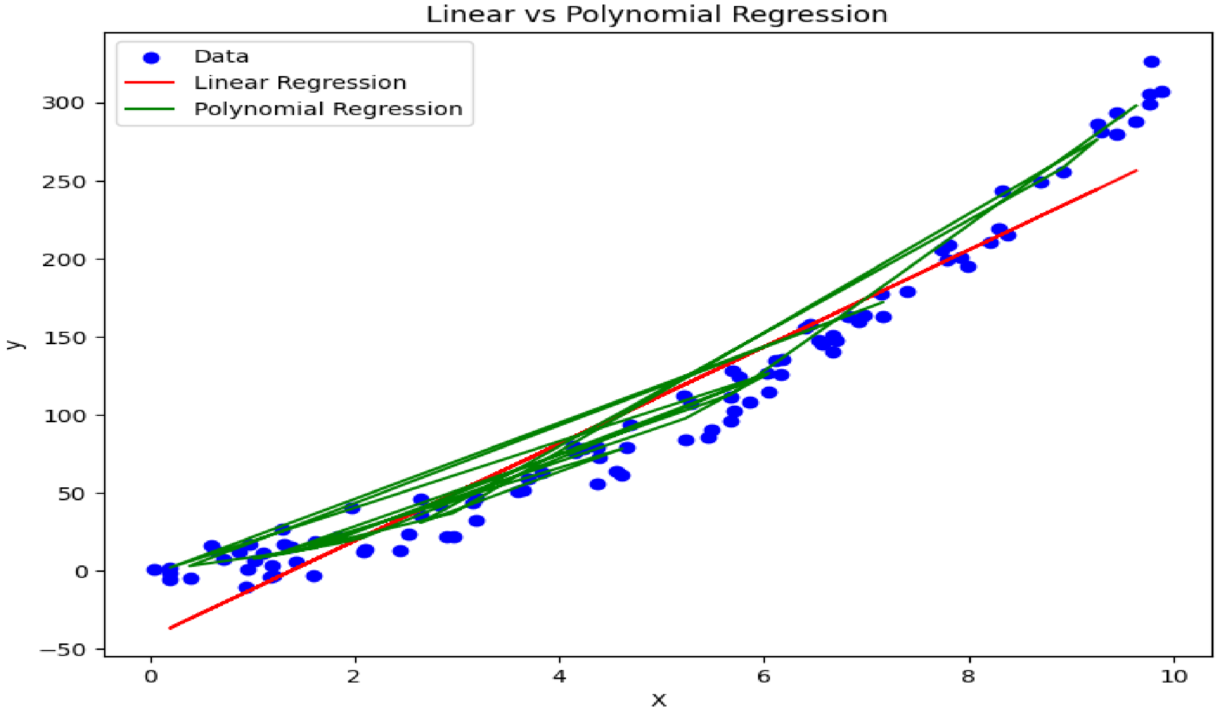

FIG. COMPARISON OF LINEAR VS POLYNOMIAL REGRESSION

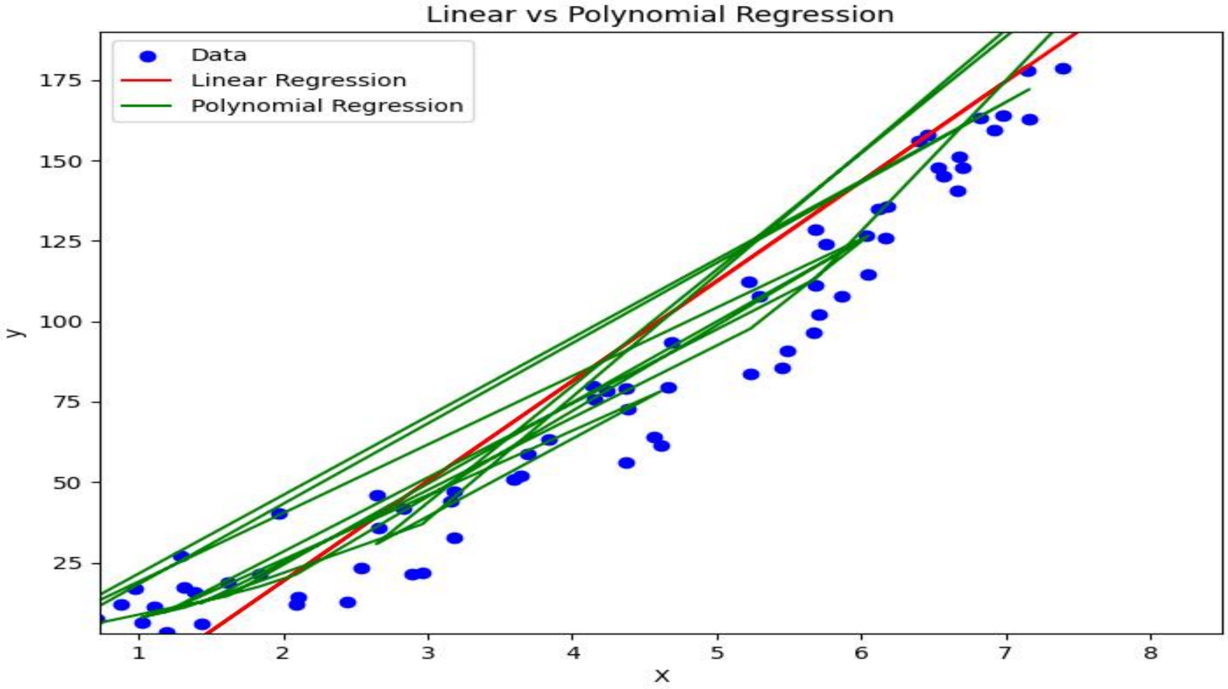

FIG. BROAD PERSPECTIVE ON REGRESSION MODELS

8. CONCLUSION:

The Minimum Description Length principle offers a robust framework for model selection and data analysis, emphasizing the importance of simplicity in achieving effective generalization from data. By providing a balance between model complexity and data fidelity, MDL serves as a valuable tool for researchers and practitioners alike. Despite its challenges in application, its theoretical foundations and empirical successes underscore its significance in the ongoing pursuit of understanding complex datasets and building predictive models. As the fields of machine learning and data science continue to evolve, the MDL principle remains relevant, guiding the development of models that are not only accurate but also interpretable and efficient.

9. REFERENCES:

- ✓ https://en.m.wikipedia.org/wiki/Minimum_description_length
- ✓ https://www.sciencedirect.com/topics/computer-science/minimum-description-length
- ✓ https://www.gabormelli.com/RKB/Minimum_Description_Length_Principle
- ✓ https://homepages.cwi.nl/~paulv/course-kc/mdlintro.pdf
- ✓ https://ojs.aaai.org/index.php/AAAI/article/view/28925/29759
- ✓ https://vitalflux.com/what-is-machine-learning-concepts-examples/