Feature Engineering in Machine Learning

Prepared by: Muhammad Raheeq ur Rehman

Date: 07 August 2025

# Table of Contents

1. Introduction

2. Tools and Libraries Used

3. Module 1: Handling Missing Values

4. Module 2: Encoding Categorical Variables

5. Module 3: Feature Scaling

6. Module 4: Polynomial and Interaction Features

7. Module 5: Binning and Feature Extraction

8. Module 6: Log and Power Transformation

9. Module 7: Target Encoding and Geospatial Features

10. Module 8: Dimensionality Reduction (PCA)

11. Conclusion

# 1. Introduction

This project focuses on the fundamental and advanced techniques of feature engineering, a critical step in the machine learning pipeline. The quality and relevance of features significantly impact model performance. This report covers practical implementations of feature engineering methods using Python and libraries such as Pandas, NumPy, and Scikit-learn.

# 2. Tools and Libraries Used

- Python 3.x

- NumPy

- Pandas

- Scikit-learn

# 3. Module 1: Handling Missing Values

Missing data is handled using techniques like deletion and imputation.  
Example:  
 df['age'] = df['age'].fillna(df['age'].mean())  
 df = df.dropna()

# 4. Module 2: Encoding Categorical Variables

Convert categorical variables into numerical format.  
Example:  
 df = pd.get\_dummies(df, columns=['gender'])  
 le = LabelEncoder()  
 df['gender\_encoded'] = le.fit\_transform(df['gender'])

# 5. Module 3: Feature Scaling

Normalize features to bring them on the same scale.  
Example:  
 scaler = StandardScaler()  
 df\_scaled = scaler.fit\_transform(df)

# 6. Module 4: Polynomial and Interaction Features

Generate new features from existing ones by interactions or powers.  
Example:  
 poly = PolynomialFeatures(degree=2)  
 df\_poly = poly.fit\_transform(df)

# 7. Module 5: Binning and Feature Extraction

Convert continuous features into bins and extract useful features from text/dates.  
Example:  
 df['age\_bin'] = pd.cut(df['age'], bins=[0,30,60,90])  
 df['text\_len'] = df['text'].apply(len)

# 8. Module 6: Log and Power Transformation

Used to normalize skewed data.  
Example:  
 df['log\_income'] = np.log(df['income'])  
 pt = PowerTransformer()  
 df\_transformed = pt.fit\_transform(df[['income']])

# 9. Module 7: Target Encoding and Geospatial Features

Encoding using the target variable and calculating distances from coordinates.  
Example:  
 df['city\_encoded'] = df['city'].map(df.groupby('city')['target'].mean())  
 from geopy.distance import geodesic  
 distance = geodesic(point1, point2).km

# 10. Module 8: Dimensionality Reduction (PCA)

Reduce the number of features while retaining most variance.  
Example:  
 pca = PCA(n\_components=2)  
 df\_pca = pca.fit\_transform(scaler.fit\_transform(df))

# 11. Conclusion

Feature engineering is one of the most critical tasks in a machine learning workflow. Properly engineered features can drastically improve the model's performance, accuracy, and generalization. This report has covered several practical feature engineering techniques that form the foundation of effective data preprocessing.