

**Project Report**

**Poverty Estimation Using**

**Predictive Analytics**

**Batch-7**

**Names: Submitting to:**

**Parul Sheel (500109597) Mam Achala Shakya**

**Shrijan Uniyal (500107378)**

**Introduction:**

This project aims to develop a machine learning framework for estimating poverty status based on socio-economic indicators. Using data from 2001 and 2011, we applied feature engineering, preprocessing techniques, and multiple classification models to predict whether poverty rates in 2011 exceed a defined threshold. The project highlights the challenges of working with socio-economic data, such as missing values and encoding, and demonstrates the effectiveness of machine learning in addressing these challenges.

**Objectives:**

The primary goal of this project is to classify regions into "poor" or "non-poor" categories based on their socio-economic characteristics and poverty rates. This binary classification problem leverages historical data and feature engineering to train predictive models.

**Motivation:**

Poverty estimation plays a crucial role in policy-making and resource allocation. Accurate predictions can help identify areas requiring intervention, improving the effectiveness of poverty alleviation programs.

**Data Source:**

The dataset contains socio-economic indicators and poverty rates for 2001 and 2011 across various regions. Features include education levels, employment rates, and other socio-economic metrics.

**Key Characteristics**

* **Size**: <Insert number> rows, <Insert number> columns.
* **Target Variable**: poverty\_status (binary: 1 if 2011 poverty rate > 20%, else 0).
* **Challenges**:
  + Missing values in numeric columns.
  + Presence of categorical variables requiring encoding.
  + Variability in feature scales.

**Methodology**

* **Preprocessing**

1. **Missing Value Treatment**:
   * Numeric columns were imputed using the median.
   * Missing values in categorical columns (if any) were handled through mode imputation or one-hot encoding.
2. **Feature Engineering**:
   * Created poverty\_status, a binary target variable based on the 20% threshold for 2011 poverty rates.
   * Dropped original poverty rate columns after deriving the target.
3. **Encoding**:
   * Categorical variables were converted into numeric representations using one-hot encoding.
4. **Feature Scaling**:
   * StandardScaler was applied to normalize features for certain models (e.g., Random Forest).

**Model Selection**

Two machine learning algorithms were used:

1. **Logistic Regression**:
   * A linear model for binary classification.
   * Does not require scaled features but benefits from clean and encoded data.
2. **Random Forest Classifier**:
   * An ensemble learning method leveraging decision trees.
   * Handles feature importance effectively and performs well with scaled data.

**Evaluation Metrics**

Models were evaluated using:

* **Accuracy**: Overall correctness of predictions.
* **Confusion Matrix**: Breakdown of true positives, true negatives, false positives, and false negatives.
* **Precision, Recall, F1-score**: To balance performance assessment on imbalanced datasets.

**Discussion**

* **Insights**
* Logistic Regression provided interpretable results, allowing insight into the relationship between features and poverty status.
* Random Forest achieved higher accuracy and robustness due to its ability to capture complex feature interactions.
* **Challenges**
* **Threshold Sensitivity**: The choice of a 20% poverty threshold significantly impacts the classification.
* **Imbalanced Data**: The distribution of "poor" vs. "non-poor" regions may have affected model performance.
* **Missing Data**: Median imputation is effective but might oversimplify the underlying data trends.

**Conclusion**

* The project demonstrates that machine learning can effectively classify regions based on poverty status. While Logistic Regression provided interpretability, Random Forest achieved superior performance through ensemble learning. However, the accuracy of predictions is influenced by the data's completeness and quality.

### References

* **Machine Learning Techniques**

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.

* + URL: https://scikit-learn.org/stable/
* **Preprocessing Techniques**

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Elsevier.

* + Discusses data preprocessing methods, including handling missing data, encoding, and scaling.
* **Evaluation Metrics**

Sokolova, M., & Lapalme, G. (2009). *A systematic analysis of performance measures for classification tasks*. Information Processing & Management, 45(4), 427-437.

* + Explains metrics like accuracy, precision, recall, and F1-score.
* **Feature Engineering**

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media.

* + Focused on creating robust features for predictive models.
* **Random Forest Algorithm**

Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5-32.

* + A seminal paper introducing Random Forests as an ensemble method.
* **Logistic Regression**

Cox, D. R. (1958). *The Regression Analysis of Binary Sequences (with discussion)*. Journal of the Royal Statistical Society. Series B (Methodological), 20(2), 215-242.

* + Foundational paper on logistic regression.
* **Python Libraries**

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, 56-61.

* + URL: https://pandas.pydata.org

Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90-95.

* + URL: <https://matplotlib.org>