SUMMER INTERNSHIP PROJECT REPORT

ON

Crime Prediction Analysis Using Stacked Ensemble Learning Model

SUBMITTED BY -

1. Dipayan Kar
2. Debnath Patra
3. Rupam Ghosh
4. Abhay Sarkar

INTERNS AT NIT SIKKIM

AND

STUDENTS OF JALPAIGURI GOVERNMENT ENGINEERING COLLEGE

UNDER THE GUIDANCE OF

DR. BAM BAHADUR SINHA

(ASST PROFESSOR, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT)

AND

DR. PRATYAY KUILA

(ASST PROFESSOR, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT)

ACKNOWLEDGEMENT

First and foremost, we are deeply thankful to, Dr Bam Bahadur Sinha, Assistant Professor, CSE Dept, NIT SIKKIM and DR. Pratyay Kuila, Assistant Professor, CSE Dept, NIT SIKKIM, our supervisor, for their invaluable guidance and support throughout the internship. Their insightful feedback, encouragement, and expert advice played a crucial role in shaping this project and enhancing our learning experience.

Lastly, we are grateful to our academic advisors and peers for their support and encouragement during this period. Their motivation and constructive criticism have been invaluable in developing my skills and understanding of machine learning and data analysis.
This internship has been an enriching journey, and we are sincerely appreciative of all the support and guidance we received along the way.

Thank you all for being an integral part of this learning experience.

1. Introduction

Crime prediction is an emerging field that leverages machine learning and data analysis to forecast criminal activities in different regions based on historical data. The goal of this project is to develop a model that can predict crime rates across 28 Indian states and union territories over a period of 12 years using a stacked ensemble learning approach.

The project aims to accurately categorize crime severity into three categories: low, medium, and high, by analysing various crime attributes.

2. Objective

The primary objective of this project was to develop a predictive model capable of categorizing crime severity (low, medium, high) for Indian states and UTs based on historical crime data.

The model leverages a stacked ensemble learning approach, combining the strengths of multiple base models to improve prediction accuracy.

3. Project Overview

This project involves building a stacked ensemble learning model that combines predictions from multiple base models, including Support Vector Machines (SVM), Decision Trees, and Random Forests.

The data used in this project includes synthetic crime data generated for 28 states and union territories of India over 12 years. The dataset includes attributes such as the number of murders, rapes, kidnappings, and other crimes, which are used to predict the overall crime severity for each year.

4. Team Structure and Responsibilities in the 8-week Internship Project

1. **Rupam Ghosh:**
    - Overall design of the stacked-based ensemble learning model.
    - Developed the Random Forest Classifier model (as standalone as well as the base model in Meta-model) and contributed to model validation.
2. **Abhay Sarkar:**
    - Focused on data collection and cleaning, ensuring the quality of input data for modelling.
    - Contributed to the implementation of the Support Vector Machine (as standalone as well as the base model in Meta-model) model.
3. **Debnath Patra:**
    - Developed the Decision Tree Classifier (as standalone as well as the base model in Meta-model) model and conducted hyperparameter tuning.
    - Assisted in the preparation of the final report.
4. **Dipayan Kar:**
    - Conducted data preprocessing, feature engineering, and model evaluation.
    - Participated in the integration of models for the stacked ensemble.
    - Developed Naïve Bayes Classifier as standalone model from scratch and compare it's result with the proposed Stacked-based Ensemble Learning Model

5. Data Collection and Preprocessing

5.1 Data Collection

The dataset used in this project was synthetically generated to mimic real-world crime data. The data includes 12 years (from 2001 to 2012) of crime records for 28 states/UTs of India. Each record includes 16 features representing different types of crimes (e.g., murder, rape, theft), and a label representing the total crime severity.

5.2 Data Attributes

- **STATE/UT**: The name of the state or union territory.
- **YEAR**: The year in which the crime data was recorded.
- **MURDER/ATTEMPT TO MURDER**: Number of murders or attempted murders.
- **RAPE**: Number of rapes reported.
- **KIDNAPPING & ABDUCTION**: Number of kidnappings and abductions.
- **DACOITY**: Number of dacoities (armed robberies) reported.
- **PREPARATION AND ASSEMBLY FOR DACOITY**: Instances of preparation for dacoity.
- **ROBBERY**: Number of robberies reported.
- **BURGLARY**: Number of burglaries reported.
- **THEFT**: Number of thefts reported.
- **RIOTS**: Number of riots reported.
- **CHEATING**: Number of cheating cases reported.
- **COUNTERFEITING**: Instances of counterfeiting reported.
- **ARSON**: Number of arson cases reported.
- **HURT/GRIEVOUS HURT**: Number of hurt or grievous hurt cases.
- **DOWRY DEATHS**: Number of dowry deaths reported.

- **CRUELTY BY HUSBAND OR HIS RELATIVES**: Instances of cruelty by husband or his relatives reported.

5.3 Data Preprocessing

- **Feature Engineering**: Polynomial features were generated to capture interactions between different crime types.
- **Normalization**: The features were standardized using StandardScaler to ensure that each feature contributes equally to the model.
- **Label Encoding**: The target variable, which categorizes crime severity into low, medium, and high, was encoded using LabelEncoder.
- **Categorization**: The total crime counts were categorized into three classes (low, medium, high) using quantile-based binning.
- **Polynomial Features**: Additional polynomial features were generated to capture non-linear relationships between the features.
- **Data Splitting:** Stratified manual split was applied to ensure a balanced representation of crime categories in the training and testing datasets.

6. Model Development

6.1 Base Models

The following base models were developed and tuned using GridSearchCV for hyperparameter optimization:

- **Support Vector Machine (SVM)**: A robust classifier used for its ability to handle high-dimensional data.
- **Decision Tree**: A simple yet effective model that uses a tree structure to make decisions based on feature values.
- **Random Forest**: An ensemble of decision trees that improves accuracy and reduces overfitting by averaging multiple decision trees' predictions.

6.2 Hyperparameter Tuning

GridSearchCV was employed to perform hyperparameter tuning for each of the base models. The parameters tuned included:

- **SVM**: C (regularization parameter) and kernel (linear or RBF).
- **Decision Tree**: max_depth and min_samples_split.
- **Random Forest**: n_estimators and max_depth.

6.3 Stacking Strategy

The stacking strategy involved training each base model using Stratified K-Fold cross-validation. The predictions from the base models were then used as input features for the meta-model, an SVM, to make the final predictions.

6.4 Final Model Training

The predictions from the base models were combined to train a meta-model, which in this case was an SVM. This stacked approach leverages the strengths of each base model to improve overall predictive accuracy. Thus, it does-

- Re-trained each base model on the full training dataset.
- Collected predictions for the test dataset using the trained base models.
- Trained the meta-model using the validation predictions from the base models.
- Final predictions were made using the meta-model on the test dataset.

7. Model Evaluation

The final stacked ensemble model achieved the following results:

- **Accuracy:** [0.992]
- **Classification Report:**

  Precision, Recall, and F1-Score for each crime category (low (0), medium (1), high (2))—

	precision	recall	f1-score	support
0	1.00	1.00	1.00	723
1	1.00	0.99	0.99	763
2	0.99	0.99	0.99	769

The model demonstrated improved accuracy and stability compared to individual base models, confirming the effectiveness of the stacking approach.

8. Extra-Models implemented for Comparison with Stack-Based Crime Prediction Model

(Link- https://github.com/R3d-Dr4g0n/ML-Internship-Project)

1. Standalone Decision Tree

- Description: A simple decision tree classifier that makes decisions based on splitting the dataset into smaller subsets based on feature values.

```
Accuracy: 0.89
Precision: 0.90
Recall: 0.89
F1 Score: 0.89
```

2. Naive Bayes Classifier

- Description: A probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between features.

```
Predictions: ['Medium' 'Low' 'Low' ... 'Low' 'Low' 'Low']
Accuracy Score: 0.9571322985957132
```

3. Support Vector Machine (SVM)

- Description: A linear classifier that attempts to find the hyperplane that best separates the data into different classes.

```
Training Accuracy: 0.588243449327603
Testing Accuracy: 0.5676274944567627
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.64      0.60       904
           1       0.58      0.49      0.53       900
```

4. Random Forest Classifier

- Description: An ensemble learning method that constructs multiple decision trees and merges their results for more accurate and stable predictions.

```
Training Accuracy: 1.0
Testing Accuracy: 0.9634146341463414
Classification Report:
              precision    recall  f1-score   support

        High       0.98      0.98      0.98       620
         Low       0.96      0.96      0.96       593
      Medium       0.95      0.94      0.94       591
```

9. Why Stacked-based Ensemble Model is better than the other models?

i. **Combining Strengths of Multiple Models:**

The Stacked Ensemble model leverages the strengths of various models to mitigate their individual weaknesses. For instance, while the SVM might struggle with non-linear relationships, the Random Forest can capture them effectively. The meta-model (in this case, another SVM) learns how to combine these predictions optimally, resulting in improved performance.

ii. **Reduced Overfitting:**

Standalone models like the Decision Tree can overfit to training data, especially when the dataset is large and complex. The Stacked Ensemble model, by combining predictions from multiple models, reduces the risk of overfitting, leading to more generalized and accurate predictions on unseen data.

iii. **Increased Stability and Robustness:**

The ensemble method adds robustness by averaging out errors from individual models. For example, if one model performs poorly on certain types of data, the other models in the ensemble can compensate for it, leading to more consistent performance across diverse datasets.

iv. **Higher Accuracy:**

As evidenced by the accuracy metrics, the Stacked Ensemble model achieved the highest accuracy (0.992) compared to the standalone models. This increase in accuracy is a direct result of the ensemble approach, which typically yields better results than any single model used in isolation.

v. **Versatility:**

The ensemble model's versatility allows it to handle different types of input data and adapt to various patterns within the data. This flexibility is crucial in complex tasks like crime prediction, where different models might capture different aspects of the data.

vi. **Meta-Learning Capability:**

The meta-model in the Stacked Ensemble learns how to weight the predictions from different base models. This meta-learning capability enables the model to make more informed decisions, taking into account the strengths of each base model.

10. Challenges and Solutions to given Stack-Based Crime Prediction Model:-

1. **Data Imbalance:**
    - **Challenge:** Crime categories were not uniformly distributed.
    - **Solution:** Stratified splitting and oversampling techniques were employed.
2. **Hyperparameter Tuning:**
    - **Challenge:** Finding the optimal parameters for each model.
    - **Solution:** GridSearchCV was used to automate the search for the best hyperparameters.
3. **Model Integration:**
    - **Challenge:** Combining predictions from diverse models into a cohesive meta-model.
    - **Solution:** Careful selection and training of the meta-model using validation data ensured a smooth integration.
4. **Model Complexity**:
    - **Challenge:** The stacked ensemble model introduced additional complexity.
    - **Solution:** Careful tuning of hyperparameters and rigorous cross-validation were necessary to ensure the model's robustness.

11. Conclusion

This project successfully demonstrated the effectiveness of stacked ensemble learning for crime prediction analysis. By combining the strengths of different models, the stacked approach provided a more accurate prediction of crime severity than any single model alone.

The use of synthetic data allowed for controlled experimentation, and the methodologies developed in this project can be applied to real-world crime data for further analysis.

The Stacked Ensemble model outperforms the standalone models in predicting crime rates due to its ability to combine the strengths of multiple models, reduce overfitting, and improve accuracy and robustness.
This makes it a superior choice for complex prediction tasks, especially in the domain of crime analysis where accurate forecasting is essential for effective decision-making and policy formulation.

12. Future Work

Future enhancements could include:

- **Incorporating Temporal Data**: Using time-series analysis to capture trends and seasonality in crime data.
- **Geospatial Analysis**: Integrating geospatial features to account for regional crime patterns.
- **Real-Time Prediction**: Adapting the model for real-time crime prediction to assist law enforcement agencies in proactive crime prevention.

13. Acknowledgements

We would like to thank our mentors and peers for their guidance and support throughout the project. Their insights and feedback were invaluable in the successful completion of this internship.

Submitted by:

Team Members:

- Dipayan Kar (Roll- 21101106010)
- Debnath Patra (Roll- 21101106061)
- Rupam Ghosh (Roll- 21101106033)
- Abhay Sarkar (Roll- 21101106042)

Internship Duration: 8 weeks (19 June- 15August)

Submission Date: 17-08-2024