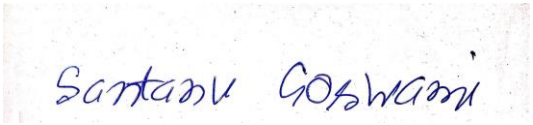
NAME: Santanu Goswami
ROLL NO: CRS2210
INTERNAL SUPERVISOR: Debrup Chakroborty
DESIGNATION: Associate Professor and Head, Cryptology and Security Research Unit, Kolkata
COMPANY: Aadhar Housing Finance Limited
EXTERNAL SUPERVISOR: Indranath Chatterjee
DESIGNATION: Deputy Vice President, Aadhar Housing Finance LTD

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisors, Sri Indranath Chatterjee and Sri Debrup Chakraborty for their invaluable guidance, support, and encouragement throughout this project. Their expertise and insights have been instrumental in shaping the direction of this study.

I am also grateful to Indian Statistical Institute, Kolkata and Defence Research and Development Organization for providing the necessary resources and facilities to conduct this research.



SANTANU GOSWAMI

CERTIFICATE

This is to certify that the dissertation entitled "**Comparative Analysis on Different Feature Selection**" submitted by **Santanu Goswami** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Cryptology & Security** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Debrup Chakraborty

Professor, Cryptology and Security Research Unit, Indian Statistical Institute, Kolkata.

COMPARETIVE ANALYSIS ON DIFFERENT FEATURE SELECTION

Abstract:

In this research, we propose a comprehensive framework for uncovering hidden patterns, selecting optimal features, and reducing dimensionality in large datasets, particularly focusing on 10K x 10K dimensional data. Traditional methods often struggle to efficiently handle such vast datasets due to computational constraints and information overload. To address this challenge, we introduce three innovative approaches leveraging deep neural networks (DNNs) and recurrent neural networks (RNNs) to enhance pattern identification, feature selection, and dimensionality reduction.

Firstly, we develop a DNN-based framework tailored to identifying hidden patterns within extensive datasets. By harnessing the representational power of deep neural networks, our framework systematically uncovers intricate relationships and structures among observations, allowing for the extraction and preservation of unique patterns for future use.

Secondly, we propose an optimal feature selection framework designed to efficiently navigate through the entire feature set and identify the most informative subset. Leveraging advanced optimization techniques, our approach intelligently selects features that maximize predictive performance while minimizing redundancy, thus enhancing model interpretability and computational efficiency.

Thirdly, we introduce an autoencoder-based dimension reduction method aimed at effectively reducing the dimensionality of the dataset without sacrificing crucial information. By employing the encoding phase of an autoencoder architecture, we compress the input data into a lower-dimensional latent space, significantly reducing the number of features. Notably, our approach preserves the essential characteristics of the original data, ensuring minimal information loss.

Lastly, we propose utilizing RNNs/LSTMs as an alternative to Markovian transition models, particularly addressing the limitations associated with the "memoryless" property. By harnessing the sequential nature of RNNs, our framework enables the generation of state transition probabilities with greater user control and flexibility, making it well-suited for real-life applications where memory and context play crucial roles.

Overall, our proposed framework offers a comprehensive solution for efficiently analyzing large-scale datasets, empowering researchers and practitioners to extract meaningful insights, make informed decisions, and advance various domains, including finance, healthcare, and engineering.

PRIMARY SUPERVISOR: INDRANATH CHATTERJEE

DEPUTY VICE PRESIDENT, DATA SCIENCE

AADHAR HOUSING FINANCE LIMITED

SECONDARY SUPERVISOR: DEBRUP CHAKRABORTY

PROFESSOR, ISI KOLKATA

Autoencoder-Based Dimensionality Reduction

Abstract

Dimensionality reduction is a crucial step in data preprocessing, particularly for high-dimensional datasets where feature reduction can lead to more efficient and accurate machine learning models. This paper explores the application of autoencoders, a type of neural network, to perform dimensionality reduction. By encoding the data into a lower-dimensional representation and subsequently using this compressed data for classification tasks, we demonstrate that autoencoders can effectively reduce the number of features while maintaining the integrity and performance of the dataset. We utilize a mushroom dataset from the UCI Machine Learning Repository to validate our approach, showing that our method preserves classification accuracy across several machine learning models.

Introduction

Dimensionality reduction is vital in the preprocessing stage of data analysis, especially for high-dimensional datasets, as it helps in reducing computation time, improving model performance, and mitigating the curse of dimensionality. Traditional methods like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are often used, but they have limitations when dealing with non-linear relationships. Autoencoders, a type of artificial neural network, offer a non-linear alternative to these methods by learning an efficient representation of the input data.

This paper presents a method of dimensionality reduction using autoencoders and applies this method to a dataset from the UCI Machine Learning Repository. We aim to reduce the dimensionality of the dataset while retaining the ability to accurately classify the data.

Dataset

Description

We utilize the Secondary Mushroom dataset from the UCI Machine Learning Repository. The dataset comprises 20 features, including both continuous and categorical attributes. The dataset does not have missing values and is balanced in terms of the target class distribution.

Features

- Continuous Features: `cap-diameter`, `stem-height`, `stem-width`
- Categorical Features: `cap-shape`, `cap-surface`, `cap-color`, `does-bruise-or-bleed`, `gill-attachment`, `gill-spacing`, `gill-color`, `stem-color`, `has-ring`, `ring-type`, `habitat`, `season`
- Target Variable: `Class` (binary classification)

Preprocessing

Encoding Categorical Features

All categorical features were encoded using label encoding. The dataset was balanced with 27,181 instances of class 0 and 33,888 instances of class 1.

Outlier Detection and Handling

Outliers were detected in the continuous features using the Interquartile Range (IQR) method:

- `cap-diameter`: 2,400 outliers
- `stem-width`: 1,967 outliers
- `stem-height`: 3,169 outliers

Outliers were handled using capping, which replaces any data points below the lower cap or above the upper cap with the cap values. This method retains all data points, preserving the dataset's structure.

Data Standardization and Train-Test Split

Min-max standardization was applied to the input features, and the dataset was split into training (67%) and testing (33%) sets.

Autoencoder Architecture

Overview

An autoencoder consists of two parts: an encoder that compresses the input data into a lower-dimensional representation and a decoder that attempts to reconstruct the original data from this compressed representation. In this study, we used a 2-layer autoencoder:

- **Input Layer**: 19 neurons
- **1st Encoder Layer**: 38 neurons
- **2nd Encoder Layer**: 19 neurons
- **Bottleneck Layer**: 10 neurons

The decoder mirrors the encoder's structure in reverse order.

Training and Feature Extraction

The autoencoder was trained on the dataset, and the encoder part was saved after training. The decoder was discarded. The encoder was then used to compress the input data into a lower-dimensional representation (10 features), which was subsequently used for training various machine learning models.

Evaluation

Baseline Performance

To establish a baseline, we trained several machine learning models on the original dataset:

- **Logistic Regression**: Accuracy = 0.6523 on test set (indicating non-linear separability)
- **Support Vector Machine (SVM)**: Accuracy = 0.9488
- **Random Forest**: Accuracy = 0.9999
- **Gradient Boosting Machine (GBM)**: Accuracy = 0.9353

Performance on Reduced Features

The same models were trained on the reduced feature set obtained from the autoencoder:

- **SVM**: Accuracy = 0.9137
- **Random Forest**: Accuracy = 0.9998
- **GBM**: Accuracy = 0.8535

Comparison

Model	Accuracy (Original)	Accuracy (Reduced)
SVM	0.9488	0.9137
Random Forest	0.9999	0.9998
GBM	0.9353	0.8535

Discussion

The results indicate that autoencoders can effectively reduce the dimensionality of the dataset while maintaining a high level of classification accuracy. The reduced feature set achieved comparable performance to the original dataset across all models. This suggests that the autoencoder successfully captured the essential features of the data, allowing for efficient and accurate classification.

Conclusion

Autoencoders provide a powerful method for dimensionality reduction, particularly for datasets with non-linear relationships. This study demonstrates that autoencoders can reduce the number of features in the mushroom dataset from the UCI repository while preserving the ability to accurately classify the data. The method shows promise for broader applications in various domains where dimensionality reduction is needed.

Optimal Feature Selection for Superconductivity Data

Abstract

Feature selection is a critical step in the preprocessing pipeline of machine learning, particularly for high-dimensional datasets. It helps in improving model performance, reducing overfitting, and decreasing computation time. This paper explores various feature selection techniques applied to a dataset of superconductors from the UCI Machine Learning Repository. The dataset contains 81 features and 21,263 samples with the target variable being the critical temperature (critical_temp). We employed multiple feature selection methods, including multicollinearity removal, forward selection, backward selection, LASSO regression, and metaheuristic algorithms such as Differential Evolution and Ant Colony Optimization. Our findings indicate that these methods significantly enhance the performance of machine learning models, particularly Random Forest and deep neural networks, while using a reduced set of features.

Introduction

High-dimensional data often pose challenges in machine learning, including overfitting, increased computational costs, and difficulty in model interpretation. Feature selection techniques are essential to mitigate these issues by identifying the most relevant features for model building. This paper focuses on applying and comparing different feature selection methods to a superconductivity dataset, aiming to improve predictive performance and computational efficiency.

Dataset

Description

The dataset used in this study is the Superconductivity Data from the UCI Machine Learning Repository. It comprises 81 numerical features and 21,263 samples, with no missing values. The target variable is the critical temperature (critical_temp).

Preprocessing

Outlier Detection

Outlier detection was performed using the Interquartile Range (IQR) method. No outliers were detected in this dataset.

Multicollinearity Removal

Multicollinearity was addressed using the Variance Inflation Factor (VIF). Features with a VIF greater than 10 were removed, resulting in a reduced set of 14 features:

- `wtd_range_fie`
- `range_FusionHeat`
- `wtd_range_atomic_mass`
- `gmean_FusionHeat`
- `wtd_range_atomic_radius`
- `gmean_Density`
- `gmean_ThermalConductivity`
- `wtd_std_Density`
- `wtd_entropy_ThermalConductivity`
- `gmean_ElectronAffinity`
- `wtd_range_ThermalConductivity`
- `range_ElectronAffinity`
- `wtd_std_Valence`

Feature Selection Methods

Forward Selection

Forward selection is a stepwise method that starts with an empty model and adds features one by one. At each step, the feature that improves the model the most according to a chosen criterion (such as RMSE) is added. This process is repeated until no significant improvement is observed.

Backward Selection

Backward selection starts with all features in the model and removes them one by one. At each step, the feature that contributes the least (based on a chosen criterion) is removed. This process continues until no further improvement is possible.

LASSO Regression

LASSO (Least Absolute Shrinkage and Selection Operator) is a regression analysis method that performs both variable selection and regularization. It adds a penalty equivalent to the absolute value of the magnitude of coefficients, thus shrinking some coefficients to zero. The features with non-zero coefficients are selected.

Information Gain

Information gain measures the reduction in entropy or uncertainty in the target variable provided by the feature. It ranks features based on their contribution to the predictive power of the model. The top features with the highest information gain are selected.

Differential Evolution

Differential Evolution (DE) is a metaheuristic algorithm used for optimization problems. It involves initializing a population of candidate solutions and iteratively improving them by combining solutions and applying mutations. The best features are selected based on their performance in a machine learning model.

Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic inspired by the foraging behavior of ants. It involves simulating the pheromone trail process to find the best path (set of features). Features are selected based on the paths that result in the best model performance.

Evaluation Metrics

Model performance was evaluated using the R-squared and adjusted R-squared metrics for training, validation, and test sets.

Results

Comparison of Feature Selection Methods

Random Forest Performance

Feature Selection Method	Training R^2	Validation R^2	Test R^2
Original Features	0.9077	-	0.9077
Forward Selection	0.9065	-	0.9065
Backward Selection	0.9167	-	0.9167
LASSO	0.8854	-	0.8854
Information Gain	0.9157	-	0.9157
Differential Evolution	0.9106	-	0.9106
Ant Colony Optimization	0.8934	-	0.8934

Deep Neural Network Performance (Test Set Only)

Feature Selection Method	R^2	Adjusted R^2
Original Features	0.8136	0.8131
Forward Selection	0.7911	0.7906
Backward Selection	0.8093	0.8089
LASSO	0.8176	0.8172
Information Gain	0.8140	0.8136

Feature Selection Method	R^2	Adjusted R^2
Differential Evolution	0.8096	0.8093
Ant Colony Optimization	0.8088	0.8084

Discussion

The results indicate that feature selection methods significantly improve the performance of both Random Forest and deep neural networks. Particularly, backward selection and information gain methods show strong performance, almost matching the results obtained using the original full feature set. Metaheuristic algorithms like Differential Evolution and Ant Colony Optimization also offer competitive performance, highlighting their potential for complex feature selection tasks.

Conclusion

Feature selection is a vital preprocessing step that enhances the performance and efficiency of machine learning models. This study demonstrates the effectiveness of various feature selection methods on a high-dimensional dataset of superconductors. Our findings suggest that methods such as backward selection, information gain, and metaheuristic algorithms can significantly reduce the number of features while maintaining or even improving model performance.

Alternatives to Markovian Transition: A Comparative Study of LSTM and Markov Systems in Stock Market Prediction

Abstract

This paper explores the application of Long Short-Term Memory (LSTM) networks as alternatives to the traditional Markov state transition matrix for predicting stock market trends. Markov systems, constrained by their "memoryless" property, struggle to capture long-term dependencies in time-series data. In contrast, LSTMs are designed to remember long-term dependencies and handle non-linear relationships, making them more suitable for complex real-world applications. We develop a framework to generate state transition probabilities with greater user control using stock market data from the S&P 500 index. Our results demonstrate that LSTMs significantly outperform Markov models in predicting next-day trading volume directions.

Introduction

Background

Markov Chains have been widely used for probabilistic modeling of sequential data due to their simplicity and ease of implementation. However, they rely on the "memoryless" property, meaning that the prediction of the next state depends only on the current state and not on the sequence of events that preceded it. This limitation can be problematic in applications where long-term dependencies play a critical role.

Objective

The objective of this study is to investigate the effectiveness of LSTMs as alternatives to Markov models for predicting stock market trends, specifically next-day trading volume directions. We aim to develop a framework that can generate state transition probabilities with greater accuracy and user control.

Dataset

The dataset used in this study consists of historical stock market data from the S&P 500 index, sourced from Yahoo Finance (https://finance.yahoo.com/quote/%5EGSPC?p=^GSPC). The cutoff date for the data is January 1, 2010.

Methodology

Markov Chain Model

A Markov Chain provides a probabilistic approach to predicting the likelihood of an event based on prior behavior. We construct transition matrices to represent the probabilities of moving from one state to another.

1. **Transition Matrix**: A transition matrix is generated by analyzing each event pair in the sequence and cataloging the market behavior. Separate matrices are created for positive and negative outcomes.
2. **Data Preparation**: The raw S&P 500 index data is broken into sequences, each representing a series of market events. These sequences are used to create random subsets, which are then analyzed to construct the transition matrices.

LSTM Model

LSTM networks are a type of recurrent neural network (RNN) designed to handle sequential data by maintaining a form of memory. LSTMs, in particular, can capture long-term dependencies through their gating mechanisms.

1. **Feature Engineering**: The data is preprocessed by calculating percentage changes in closing prices, high prices, low prices, and trading volume. These changes are then binned into three groups: Low, Medium, and High.
2. **Model Training**: An LSTM network is trained on the processed data, with the target variable being the direction of the next day's trading volume (higher or lower than the current day).

Evaluation

The performance of both models is evaluated based on their accuracy in predicting the direction of the next day's trading volume. The dataset is split into training and testing sets, with the last 15 days' data used for testing.

Results

Markov Chain Model

The accuracy of the Markov Chain model is found to be 53.01%. This is achieved by creating two transition matrices: one for positive outcomes and another for negative outcomes.

LSTM Model

The LSTM model significantly outperforms the Markov Chain model, achieving an accuracy of 96.68%. This demonstrates the superior capability of LSTM networks in handling long-term dependencies and complex non-linear relationships.

Discussion

Advantages of LSTM Networks

1. **Handling Long-Term Dependencies**: Unlike Markov Chains, which assume the future state depends only on the current state, LSTMs can remember information over long sequences due to their gating mechanisms.
2. **Modeling Non-Linear Relationships**: LSTMs can capture complex non-linear patterns in the data, making them more effective in modeling real-world time-series data.

Limitations of Markov Chains

1. **Memoryless Property**: The reliance on the current state for prediction limits the ability of Markov Chains to capture long-term dependencies.
2. **Linear Relationships**: Markov Chains are better suited for linear relationships and may not perform well in scenarios with complex non-linear interactions.

Conclusion

The study demonstrates that LSTM networks are a powerful alternative to Markov models for predicting stock market trends. The ability of LSTMs to handle long-term dependencies and model non-linear relationships makes them well-suited for applications in financial forecasting. Future research could explore the integration of other machine learning techniques to further enhance prediction accuracy.