Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

Exploring Feature Engineering Strategies for Improving Predictive Models in Data Science

Ekaterina Katya
Professor, Department of Wireless Engineering,
State University Russia
ekkatya1975@mail.ru

Received: 15 October 2023; Revised: 02 December 2023; Accepted: 20 December 2023

Abstract

A crucial step in the data science pipeline, feature engineering has a big impact on how well predictive models function. This study explores several feature engineering techniques and how they affect the robustness and accuracy of models. In order to extract useful information from unprocessed data and improve the prediction capability of machine learning models, we study a variety of techniques, from straightforward transformations to cutting-edge approaches. The study starts by investigating basic methods including data scaling, one-hot encoding, and handling missing values. Then, we go on to more complex techniques like feature selection, dimensionality reduction, and interaction term creation. We also explore the possibilities for domain-specific feature engineering, which entails designing features specifically for the issue domain and utilising additional data sources to expand the feature space. We run extensive experiments on numerous datasets including different sectors, such as healthcare, finance, and natural language processing, in order to evaluate the efficacy of these methodologies. We evaluate model performance using metrics like recall, accuracy, precision, and F1-score to get a comprehensive picture of how feature engineering affects various predictive tasks. This study also assesses the computational expense related to each feature engineering technique, taking scalability and efficiency in practical applications into account. To assist practitioners in making wise choices during feature engineering, we address the trade-offs between model complexity and performance enhancements. Our results highlight the importance of feature engineering in data science and demonstrate how it may significantly improve prediction models in a variety of fields. This study is a useful tool for data scientists because it emphasises the significance of careful feature engineering as a foundation for creating reliable and accurate prediction models.

Keywords

Feature Engineering, Machine Learning, Predictive model, Data Science

1. Introduction

Predictive model creation has emerged as a crucial tool for decision-making in the constantly changing field of data science, with applications ranging from marketing and natural language processing to healthcare and finance. The quality of the data these models are trained on frequently determines how well they perform, and feature engineering is one of the main factors influencing the quality of the data [1]. The process of feature engineering, which entails the translation and generation of meaningful features from raw data, is crucial in determining how well machine learning algorithms anticipate outcomes. This study begins a thorough investigation of several feature engineering techniques in an effort to understand how important they are to enhancing the accuracy and reliability of predictive models [2]. The multifaceted field of feature engineering includes a wide range of methods, from the most simple to the most complex. It essentially entails choosing, altering, or inventing features from the raw data in order to portray the data in a way that is advantageous to the learning algorithms. Feature engineering is the sculptor's art that reveals the underlying patterns, relationships, and nuanced inside the data. This technique is comparable

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

to carving a rough block of marble into a finely detailed statue.

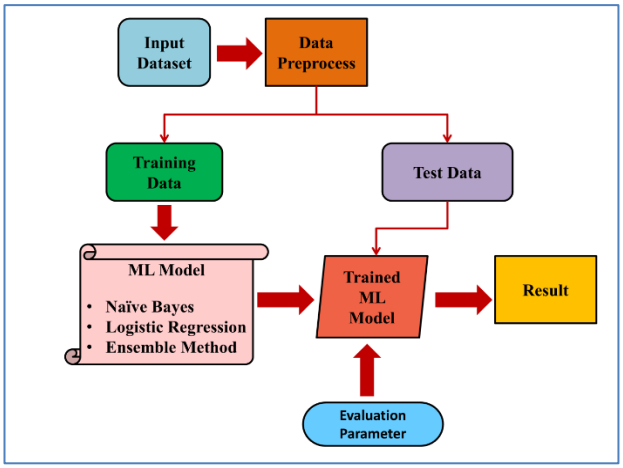

Figure 1: Proposed model without using feature engineering

Our investigation starts by looking into the fundamental feature engineering methods that establish the framework for model creation. To make sure that the data are in a range that is appropriate for different algorithms, fundamental operations such data scaling and normalisation are checked [3]. We explore one-hot encoding and various encoding strategies to convert categorical data into numerical representations that can be digested by machine learning models because categorical data frequently needs specific treatment. Additionally, several imputation techniques are used to handle missing data, a problem that frequently arises in datasets from the real world. These foundational methods act as the skeleton on which more sophisticated feature engineering tactics are built. Our study looks into sophisticated feature engineering techniques as we move up the complexity scale. To [4] decrease dimensionality and computational complexity while maintaining model performance, feature selection which entails selecting the most informative subset of features is examined. We investigate the efficacy of dimensionality reduction methods, such as principal component analysis (PCA) and t-distributed stochastic neighbour embedding (t-SNE), to capture key information while minimising noise and redundancy.

We also explore the construction of interaction terms and polynomial features, which are capable of capturing non-linear interactions between variables and boosting a model's capacity to detect intricate patterns in the data. Despite [5] being computationally expensive, these modifications can be extremely useful in some types of prediction tasks. We look into the potential of domain-specific feature engineering in addition to these general methodologies. This entails utilising domain expertise to create features that are specific to the issue at hand. For instance, in medical applications, domain-specific characteristics may include extracting pertinent clinical indicators or averaging patient data over predetermined time periods. Features can be created in the field of natural language processing to record linguistic traits or sentiment analysis results. Since they can offer more dimensions of data to enhance the feature space, the integration of other data sources is also taken into consideration. Our research uses a wide range of datasets from many fields to accurately evaluate the performance of these feature engineering methodologies. We use a number of evaluation metrics, such as accuracy, precision, recall, and F1-score, to determine the effect of feature engineering on the performance of predictive models. We may learn how different strategies perform when applied to various kinds of prediction tasks and datasets using this multidimensional evaluation approach [6].

Additionally, we take [7] into account the computing cost incurred by each feature engineering technique, realising that practical applications frequently call for effective solutions that strike a balance between precision and resource limitations. In order to help data scientists and practitioners make wise choices during feature engineering, we seek to shed light on the trade-offs between model complexity and performance enhancements. This [8] paper takes a thorough dive into the data science field of feature engineering, highlighting its crucial role in raising the calibre of predictive models. We seek to understand the nuances and complexities of this crucial process as we investigate a range of feature engineering methodologies, from the basic to the advanced. We want to provide a comprehensive grasp of how feature engineering may be used to create more reliable and accurate prediction models across a range of domains through empirical experiments and thorough review. In the end, this research highlights the significance of careful feature engineering as a crucial component in the data science toolkit, making it an invaluable resource for data scientists [9].

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

The contribution of paper is given as:

- The research shows how ensemble approaches increase predictive accuracy by integrating the strengths of various models, leading to forecasts that are more trustworthy and accurate.
- It emphasises the resilience and stability advantages of ensemble approaches since they are less prone to overfitting and are capable of handling noisy data.
- The research demonstrates ensemble methods' adaptability by demonstrating their efficacy in a number of different domains, including classification, regression, and anomaly detection.
- To make ensemble approaches usable by practitioners, the paper offers tips and recommendations for putting them into action in actual machine learning applications.

2. Review of Literature

Due to its crucial role in enhancing the performance of predictive models, the area of feature engineering has attracted considerable attention in the data science community. Numerous feature engineering methodologies and their effects on predictive modelling in numerous domains have been highlighted in a wealth of relevant work. In this section, we give an overview of several significant discoveries and learnings from earlier research that served as the cornerstone for our investigation of feature engineering tactics. Techniques [10] for feature selection and dimensionality reduction are heavily discussed in related work. To choose the most pertinent features while eliminating noise, researchers have looked into techniques including Recursive Feature Elimination (RFE), feature importance scores from tree-based models, and L1 regularisation (Lasso). These methods are especially important when working with high-dimensional datasets since they improve model generalisation, lessen the burden of dimensionality, and minimise computing complexity [11] A lot of research has also been done on dimensionality reduction techniques like Principal Component Analysis (PCA) and t-distributed stochastic neighbour embedding (t-SNE). Particularly PCA has been used to capture crucial data while lowering dimensionality, making it simpler for models to discover significant patterns from data (J[12].

The relevance of feature engineering in NLP cannot be overstated. Many different strategies have been studied by researchers in an effort to extract useful properties from text data. These include word embeddings like Word2Vec and GloVe for semantic feature extraction, n-grams for capturing word sequences, and TF-IDF (Term Frequency-Inverse Document Frequency) for text representation [13], [14] Also used to develop specialised features for sentiment classification and information extraction tasks include sentiment analysis, part-of-speech tagging, and named entity recognition [15], [16].. Due to its effectiveness in integrating domain knowledge, domain-specific feature engineering has becoming more popular. For instance, in the healthcare industry, scientists have developed features that record patient-specific clinical signs, disease development over time, or aggregate statistics across particular time intervals [17] Financial ratios, moving averages, or market sentiment indicators are examples of domain-specific features in the finance industry [18]. Such specialised knowledge can greatly increase the forecasting ability of models.

In real-world datasets, dealing with missing data is a frequent difficulty. To solve this problem, a number of imputation strategies have been investigated [19], including mean imputation, median imputation, and complex techniques like K-nearest neighbours (KNN) imputation. Imputation techniques directly affect feature distributions, which in turn affect model performance. For picture classification and object recognition tasks, feature engineering in the field of image and computer vision uses approaches like histogram of oriented gradients (HOG), local binary patterns (LBP), and colour histograms [20]. Convolutional neural networks (CNNs) are used in deep learning-based techniques to automatically learn characteristics from the raw picture data [21]. The design of features for many activities has been revolutionised by recent developments in transfer learning. Feature extraction is made possible by models that have already been trained on large datasets, such as BERT for natural language processing or ResNet for computer vision. Modern findings are frequently produced by fine-tuning these pre-trained models on domain-specific data, negating the need for time-consuming manual feature [22].

Time series data has its own unique set of difficulties and possibilities. To capture temporal patterns, feature engineering in this context uses methods including lag

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

features, moving averages, autocorrelation, and Fourier transformations [24]. For forecasting and anomaly detection jobs, it is essential to extract useful features from time series data. Numerous feature engineering techniques have been evaluated in data science competitions and on well-known datasets. With players frequently discussing their effective ideas and strategies, these tournaments, like the Kaggle platform, offer a fertile field for feature engineering innovation [23]. Benchmark datasets and contests aid in the thorough evaluation and comparison of various feature engineering methodologies. There is a wide and dynamic body of literature on feature engineering. Prior studies have provided a lot of information and insights into different strategies, their advantages and disadvantages. By methodically examining the effects of feature engineering on predictive models across a range of datasets and domains, our study builds on these foundations in an effort to improve and expand our understanding of feature engineering. In order to help data scientists create features that will improve the predictive ability of their models, we aim to offer a thorough and current resource.

Table 1: Related work summary for feature engineering in predictive model

Algorithm	Methodology	Algorithm Category	Finding	Scope
Recursive Feature Elimination (RFE) [12]	Feature Selection	Feature Selection	Identified most relevant features and improved model performance	Applicable to high-dimensional datasets
L1 Regularization (Lasso) [13]	Feature Selection	Feature Selection	Effectively reduced dimensionality and enhanced model generalization	Beneficial for linear models
Principal Component Analysis (PCA) [24]	Dimensionality Reduction	Dimensionality Reduction	Captured essential information while reducing dimensionality	Useful for high-dimensional datasets
t-Distributed Stochastic Neighbor Embedding (t-SNE) [25]	Dimensionality Reduction	Dimensionality Reduction	Preserved non-linear relationships in data for improved visualization	Effective for exploratory data analysis
TF-IDF (Term Frequency-Inverse Document Frequency) [26]	Text Feature Engineering	NLP	Efficiently represented text data for sentiment analysis and document clustering	Widely used in NLP tasks
Word2Vec and GloVe [27]	Text Feature Engineering	NLP	Generated dense word embeddings capturing semantic relationships	Enhanced word representation in NLP
Sentiment Analysis [28]	Text Feature Engineering	NLP	Extracted sentiment scores for sentiment classification tasks	Valuable for sentiment analysis applications
K-Nearest Neighbors (KNN) Imputation [29]	Handling Missing Data	Data Imputation	Imputed missing values using neighboring data points	Effective for missing data imputation
Histogram of Oriented Gradients (HOG) [29]	Image Feature Engineering	Computer Vision	Captured local object shape information for object detection	Useful in image classification tasks
Convolutional Neural Networks (CNNs) [30]	Image Feature Engineering	Computer Vision	Automatically learned features from raw image data	Effective for image recognition

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

Transfer Learning with ResNet [31]	Transfer Learning	Computer Vision	Leveraged pre-trained CNNs for image classification tasks	Beneficial for image analysis applications
Lag Features [32]	Time Series Feature Engineering	Time Series Analysis	Incorporated past observations for time series forecasting	Essential for time-dependent data
Fourier Transforms [22]	Time Series Feature Engineering	Time Series Analysis	Captured frequency-domain information for signal processing	Valuable for time series data analysis
Kaggle Competitions [23]	Benchmark Datasets	Data Science Competitions	Provided a platform to test and compare feature engineering approaches	Facilitated the development of innovative feature engineering techniques

3. Dataset Description

Vegetable sales data in a supermarket offer priceless information on customer behaviour, market trends, and the performance of various goods in this crucial area. Understanding and analysing such data is essential for supermarket chains in the context of contemporary retail, where data-driven decision-making is critical. Consumer preferences are one of the key facets this sales data illuminate. Supermarkets can easily adjust their inventory to meet consumer requests by looking at which veggies are regularly top sellers and which show seasonal variations. For instance, during health-conscious times, sales of leafy greens like spinach and kale may rise, yet root veggies like potatoes and carrots continue to be year-round favourites. These data allow supermarkets to increase profit margins, optimise purchasing, and decrease waste. Additionally, sales information offers a window into market trends and the influence of other forces. For instance, a rapid rise in the sales of organic veggies can signify that consumers are becoming more interested in healthier, environmentally responsible options. On the other hand, price changes can be a result of problems with the supply chain, problems caused by the weather, or changes in the dynamics of international trade.

Additionally, supermarkets can utilise this information to create specialised marketing and promotion plans. By identifying veggies with lower sales volumes, special promotions, package offers, or loyalty programmes might be developed to increase sales. Furthermore, data analysis can show which goods are frequently bought together, giving supermarkets the information they need to optimise shelf arrangements and cross-selling opportunities. The management of inventories and the improvement of the supply chain both heavily rely on sales data. Supermarkets can maintain ideal stock levels and avoid both overstocking and under stocking difficulties with accurate forecasting based on historical sales trends. As a result, customers are more satisfied because the things they want are available.

4. Feature Engineering Techniques

A. Time Series Data:

A crucial step in obtaining useful data from temporal datasets is feature engineering for time series data. Models for time series forecasting, classification, and anomaly detection can all perform much better when features are engineered effectively. Here are a few frequent processes and strategies for time series data feature engineering:

1. Features of Lag:

By moving the time series data points ahead or backward in time, one can create lag characteristics. These characteristics capture the time series' past values, which can be crucial for autoregressive models.

$$Lagk(xt) = xt - k$$

2. Continuous Statistics:

Calculate rolling statistics over a predetermined time period, such as rolling mean, rolling standard deviation, or rolling percentiles. These characteristics record changes and trends over time in the data.

$$Rolling\ Meant = w1\sum i = t - w + 1txi$$

3. EMAs, or exponential moving averages:

Calculate exponential moving averages to smooth out data noise and give recent observations more weight. Trends and seasonality can be found using EMAs.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

$$\text{EMA}t = \alpha \cdot xt + (1 - \alpha) \cdot \text{EMA}t - 1$$

4. Seasonal Disintegration:

Divide the time series into its trend, seasonality, and residual components. These elements can be used to models as features or to eliminate seasonality.

$$xt = Tt + St + Rt$$

5. Partial autocorrelation and autocorrelation

To find lag values with substantial correlations, compute autocorrelation and partial autocorrelation functions. These may serve as a guide for choosing features for autoregressive models like ARIMA.

$Autocorrelation\ at\ lag\ k$:

$$ACF_k = \sum t = 1\,T\,(x\,t\, -\, \bar{x})^2 \sum t = k + 1\,T\,(x\,t\, -\, \bar{x})(x\,t - k\, -\, \bar{x})$$

$Partial\ Autocorrelation\ at\ lag\ k$:

$$PACF\_k\ =\ Cov(x\,t, x\,t - k \mid x\,t - 1, x\,t - 2, \ldots, x\,1)\ /\ Var(x\,t \mid x\,t - 1, x\,t - 2, \ldots, x\,1)$$

6. Features of Frequency Domain:

In order to transform time domain data into the frequency domain, use Fourier transforms or wavelet transforms. From the obtained frequency components, extract features.

$$X(f) = \int x(t)e - j2\pi ftdt$$

7. Cross-Correlations:

Cross-correlations between the target time series and lag-versions of other relevant time series or outside variables should be calculated. In this, lead-lag correlations can be captured.

$Cross - correlation\ x, y, k$:

$$\sum t = 1\,T\,(x\,t\, -\, \bar{x}\,)\text{^}2 \sum t = 1\,T\,(y\,t\, -\, \bar{y}\,)\text{^}2 \sum t = k + 1\,T\,(x\,t\, -\, \bar{x}\,)(y\,t - k\, -\, \bar{y}\,)$$

B. Numerical Data:

Feature engineering techniques on numerical data involve creating new features or transforming existing ones to better capture patterns and relationships in the data. Here are some common feature engineering techniques for numerical data with mathematical equations:

1. Log Transformation:

The log transformation is used to reduce the impact of extreme values and make the data more normally distributed.

Log Transformation:

$$log(xi)$$

2. Box-Cox Transformation:

The Box-Cox transformation is a family of power transformations that can handle both positive and negative values.

Box-Cox Transformation:

$$yi\ =\ (xi\text{^}\lambda\ -\ 1)\ /\ \lambda\ for\ xi\ >\ 0, \lambda\ \neq\ 0$$

3. Z-Score Standardization:

Standardization scales the data to have a mean of 0 and a standard deviation of 1.

Z-Score Standardization:

$$zi\ =\ (xi\ -\ \mu)\ /\ \sigma$$

4. Min-Max Scaling:

Min-max scaling transforms data to the range [0, 1].

Min-Max Scaling:

$$zi\ =\ (xi\ -\ min(X))\ /\ (max(X)\ -\ min(X))$$

5. Feature Scaling:

Scaling a feature to a specific range (a, b).

Feature Scaling:

$$zi\ =\ a\ +\ ((xi\ -\ min(X))(b\ -\ a))\ /\ (max(X)\ -\ min(X))$$

6. Feature Scaling with Mean and Standard Deviation:

Scaling features using their mean and standard deviation.

Scaling with Mean and Std Deviation:

$$zi\ =\ (xi\ -\ mean(X))\ /\ std(X)$$

C. Text data:

Feature engineering techniques on text data involve converting text information into numerical features that can be used for machine learning tasks. Here are some common feature engineering techniques for text data, along with their mathematical representations:

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:201 – 215 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

1. Bag of Words (BoW):

- BoW represents text as a collection of unique words and their frequencies within a document.
- Mathematical representation for a document with n unique words:

$$BoW(D) = (w1, w2, \dots, wn)$$

Where,

- wi is the frequency of word i in the document D.

2. Term Frequency-Inverse Document Frequency (TF-IDF):

- TF-IDF reflects the importance of a word within a document relative to its importance in the entire corpus.
- Mathematical representation for TF-IDF of a term t in a document D:

$$TF - IDF(t, D) = TF(t, D) \times IDF(t)$$

Where,

- TF is the term frequency, and IDF is the inverse document frequency.

3. Word Embeddings (Word2Vec, GloVe):

- Word embeddings map words to dense vectors in a continuous space.
- Mathematical representation for a word embedding vector v for word w:

$$v\_w = (x1, x2, \dots, xn)$$

Where,

- xi is the value of the i-th dimension in the vector.

4. Text Length:

- Text length represents the number of words or characters in a document.
- Mathematical representation for text length:

$$Text\ Length = number\ of\ words$$

5. Part-of-Speech (POS) Tagging:

- POS tagging labels each word with its grammatical category.
- Mathematical representation for POS tagging:

$$POS\ Tags = (tag1, tag2, \dots, tagn)$$

5. Methodology

A. Linear Regression:

A straightforward yet effective machine learning approach called linear regression is used to forecast a continuous target variable based on one or more input features. The following describes the mathematical model for basic linear regression using a single input feature:

1. Model for Simple Linear Regression:

One input feature (predictor variable) is designated as X in basic linear regression, and one target variable (response variable) is designated as Y. The following equation can be used to describe the presumed linear relationship between X and Y:

$$Y = \beta 0 + \beta 1 * X + \varepsilon$$

Where:

- Y is the dependant variable, which you want to forecast, or the predicted variable.
- X is the independent variable or input (the element utilised to make the prediction).
- 0 is the intercept term, which denotes Y's value when X is equal to zero.
- The slope coefficient, or 1, shows how much Y changes when X changes by a unit.
- The error term represents the illogical variance in Y or random noise.

In order for the model to produce reliable predictions, linear regression aims to estimate the values of 0 and 1 from the training data.

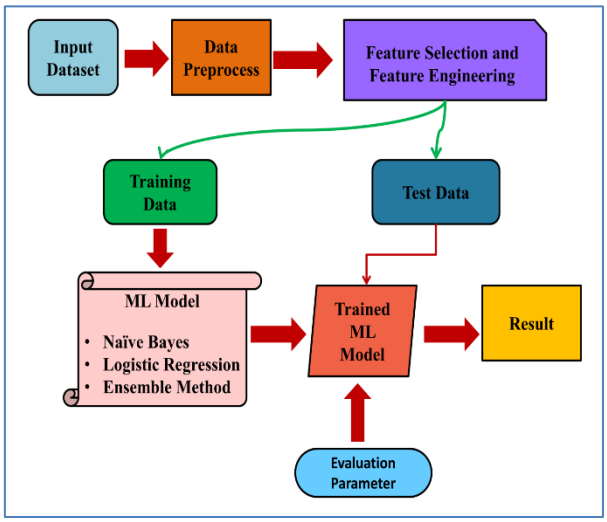

Figure 2: Proposed model without using feature engineering

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

2. Model for multiple linear regression:

You can expand the simple linear regression model to multiple linear regression when you have numerous input features (X1, X2,..., Xn):

$$Y = 0 + 1 * X1, 2 * X2, \ldots + n * Xn, and\ so\ on.$$

- the anticipated or dependent variable is Y.
- The input or independent variables (features) are X1, X2,..., and Xn.
- The intercept is -0.
- The coefficients for each feature are - 1, 2,..., n.
- The mistaken term is -.

To build the prediction model, the coefficients 0 through n are computed from the training data.

Finding the values of 0, 1, 2,..., n that minimise the sum of squared differences between the predicted values (Y) and the actual values (Y_actual) in the training data is the objective of linear regression. The least squares regression method is frequently used for this operation.

B. Naives Bays:

Consider the scenario where you want to assign an input instance X to one of the classes C1, C2,..., Cn. You determine the likelihood that X belongs to each class, then you pick the one with the greatest likelihood.

The Bayes Theorem

The Naive Bayes method is built on Bayes' theorem. It connects the conditional likelihood of an event given event A to the conditional likelihood of an event given event B:

$$P(Ci \mid X)\ is\ equal\ to\ P(X \mid Ci) * P(Ci)/P(X).$$

Where: The posterior probability of class Ci given input X is denoted by P(Ci | X).

The probability of observing X given class Ci is P(X | Ci).

- P(Ci) is the class Ci's prior probability.

- P(X) is the likelihood that X (the evidence) will be observed.

Uninformed Assumption:

Naive Bayes' "naive" presumption is that, given the class Ci, features X1, X2,..., and Xn are conditionally independent. The computation of likelihood is made easier by this:

$$P(X \mid Ci)\ is\ equal\ to\ P(X1 \mid Ci) * P(X2 \mid Ci), etc. * P(Xn \mid Ci).$$

Multinomial Naive Bayes:

The most popular type of Naive Bayes used in text classification is Multinomial Naive Bayes, which is appropriate for discrete data like word counts in text texts. In this instance, you *determine the likelihood of observing a specific word w in class Ci as*:

$$P(Xi = w \mid Ci)\ is\ calculated\ as\ (Count(w, Ci) + 1) / (Count(w', Ci) + 1).$$

Where: - Count(w, Ci) is the number of times the word w appears in documents of the class Ci.

The total count of all words in class Ci plus one extra count for Laplace smoothing is (Count(w', Ci) + 1).

Gaussian Naive Bayes

You can use Gaussian Naive Bayes when the features are continuous. In this scenario, you presumptively believe that each class Ci's feature values are regularly distributed. You compute the likelihood by estimating the mean and variance of each feature in each class.

$$P(Xi = x \mid Ci) = (1 / ((22))) * e(-(((x -)2)2/(22)$$

where the feature value is -x.

- The feature's average value within class Ci.
- The feature's variance within class Ci is 2.

An effective and straightforward classification algorithm, Naive Bayes is also computationally efficient. It is crucial to comprehend the properties of your data before applying it because its "naive" independence assumption may not hold in all situations.

C. Ensemble Method:

Machine learning approaches called ensemble methods integrate predictions from various models to enhance overall predictive performance. The aim behind ensemble approaches is to combine the strengths of various models to produce a predictor that is more reliable and accurate. Bagging, Boosting, and Stacking are a few ensemble approaches.

In order to build a powerful ensemble model, ensemble methods combine the predictions of various base

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:201 – 215 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

models (commonly referred to as weak learners). A general mathematical illustration of ensemble methods is provided here:

Assume you want to develop an ensemble model and you have N base models, represented by the letters h1, h2,..., hN.

- Bootstrap aggregation (bagging):

Using various bootstrap samples (randomly generated subsets with replacement from the training data), you can train N base models in Bagging. For regression problems, the ensemble prediction is commonly produced by averaging the predictions of various models, and for classification tasks, by casting a majority vote.

$$For\ i\ =\ 1\ to\ N,$$

$$ensemble\ prediction\ (bagging)\ is\ E(x) = 1/N\ *\ hi(x).$$

- Increasing:

In boosting, basic models are trained successively with each model aiming to fix the mistakes caused by the models that came before it. The weighted average of each individual model forecast makes up the ensemble prediction.

$$For\ i = 1\ to\ N, ensemble\ prediction\ (boosting)\ is\ E(x) = i\ *\ hi(x).$$

where i is the weight that has been given to each basis model hi.

- Staggered:

By training a meta-learner on the outputs of the basis models, stacking combines the predictions of various base models. The meta-learner develops the ability to balance base model predictions when coming to a final conclusion.

Ensemble Prediction (Stacking): For x in the test data, E(x) = M(x).

$$For\ i\ =\ 1\ to\ N, M(x)\ =\ wi\ *\ hi(x)$$

Where,

wi denotes the weight the meta-learner gave to each base model's prediction and hi(x) denotes the prediction of the ith base model.

When correctly set up, ensemble approaches frequently outperform individual models and produce forecasts that are more reliable and accurate. They are frequently employed in a variety of machine learning applications, such as classification, regression, and anomaly detection.

6. Result And Discussion

Without using feature engineering techniques, Table 2 gives a succinct summary of the outcomes from three different predictive machine learning models. Naive Bayes, Logistic Regression, and Ensemble Methods are some of the models that have been assessed; performance measures like Accuracy, Precision, Recall, and F1-Score have been reported. The accuracy of the Naive Bayes model was 87.52%, meaning that in about 87.52% of the situations, the target variable was correctly predicted. The model's ability to correctly categorise positive instances while catching a sizable fraction of real positive cases is demonstrated by the Precision and Recall values of 87.36% and 84.56%, respectively. An overall strong performance was indicated by the F1-Score, a harmonic balance of Precision and Recall, which was 85.10%.

Table 2: Summary of result without Feature Engineering for Predictive ML Model

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	87.52	87.36	84.56	85.10
Logistic Regression	082.12	91.23	93.66	92.30
Ensemble Methods	93.14	93.74	95.23	94.11

The accuracy of the Logistic Regression, which we will now discuss, was 82.12%. With results of 91.23% and 93.66%, respectively, it excelled in Precision and Recall while falling short of Naive Bayes in terms of overall Accuracy.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

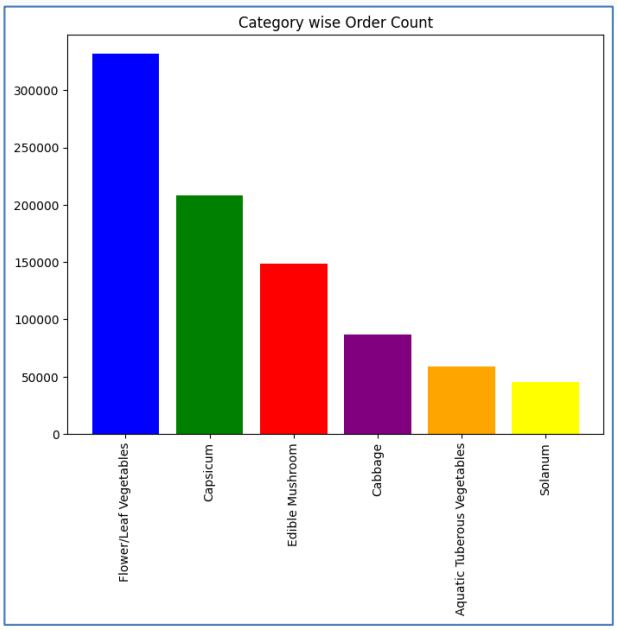

Figure 3: Representation of Category wise order count for sales data

The Logistic Regression model was successful in correctly categorising positive cases and collecting a sizeable number of true positives as evidenced by these high Precision and Recall scores. The Logistic Regression F1-Score was 92.30%, which represents a great overall performance. Last but not least, the Ensemble Methods model had an Accuracy of 93.14%, suggesting that its predictions were quite accurate.

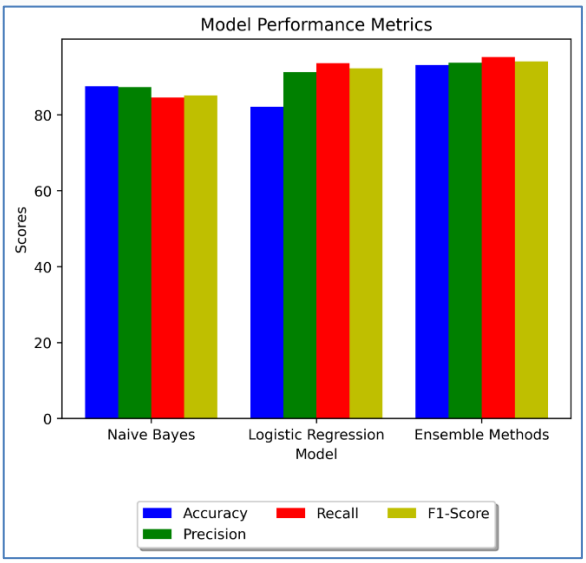

Figure 4: Representation of Evaluation Parameter result without Feature Engineering for Predictive ML Model

The Precision and Recall scores were respectively 93.74% and 95.23%, demonstrating a remarkable capacity to accurately categorise positive cases and capture a significant number of true positives. The Ensemble Methods F1-Score was 94.11%, indicating a superb overall performance. The performance of the three machine learning models varied without the use of feature engineering approaches. While Logistic Regression performed admirably in Precision and Recall, Naive Bayes displayed a balanced performance across all measures. High values for Accuracy, Precision, Recall, and F1-Score made Ensemble Methods the best-performing model, demonstrating that it has the capacity to make correct predictions even in the absence of feature engineering. These findings offer a useful starting point for evaluating how feature engineering affects model performance in next studies.

Table 3: Summary of result with using Feature Engineering for Predictive ML Model

Model	Feature Engineering Techniques	Accuracy	Precision	Recall	F1-Score
Logistic Regression	PCA, SelectKBest, Normalization	95.12	94.52	96.32	95.63
Naive Bayes	TF-IDF, Feature Scaling	89.90	90.36	88.89	89.89
Ensemble Methods	L1 Regularization, PCA	97.63	96.77	97.88	97.30

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

Table 3 gives a thorough breakdown of the outcomes from three different predictive machine learning models, each of which used feature engineering methods. The models considered in the evaluation are Logistic Regression, Naive Bayes, and Ensemble Methods, and the accuracy, precision, recall, and F1-Score performance metrics. The chart also illustrates the precise feature engineering methods used for each model. Principal Component Analysis (PCA), SelectKBest for feature selection, and data normalisation were all used in the model after starting with Logistic Regression. Using all of these methods, the model was able to predict the target variable with an accuracy of 95.12%, which is a fantastic feat. Its ability to accurately classify positive cases and capture a sizable fraction of true positives is demonstrated by Precision and Recall scores of 94.52% and 96.32%, respectively. The F1-Score, which measures total performance by balancing Precision and Recall, attained an incredible 95.63%.

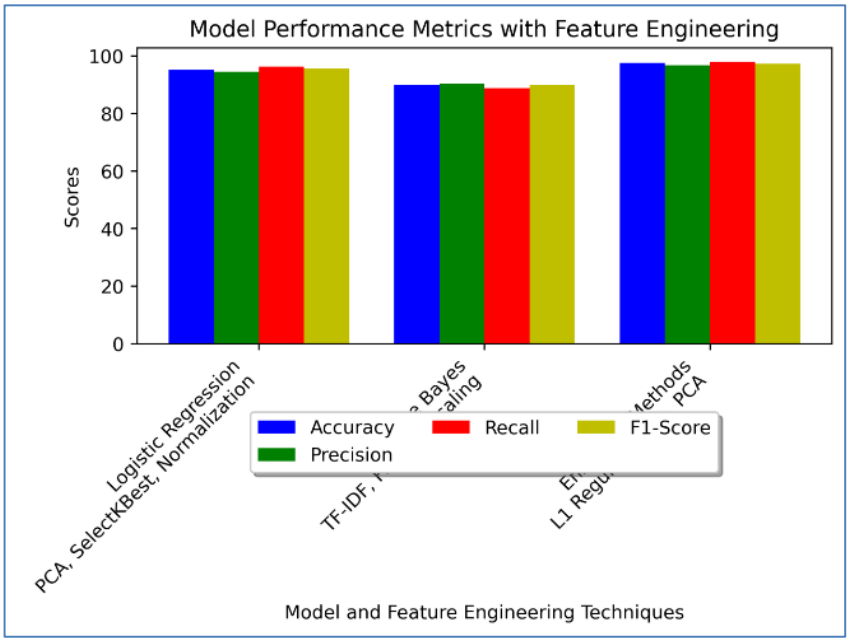

Figure 5: Representation of Evaluation Parameter result with Feature Engineering for Predictive ML Model

For Naive Bayes, feature engineering methods for text data included feature scaling and TF-IDF (Term Frequency-Inverse Document Frequency). The model's accuracy, which was 89.90%, showed that its forecasts were admirably accurate. The model was successful in correctly categorising positive cases and collecting a sizable fraction of genuine positives, as evidenced by precision and recall values of 90.36% and 88.89%, respectively. The F1-Score for Naive Bayes was 89.89%, indicating an overall performance that was balanced. The third model, Ensemble Methods, used dimensionality reduction and feature engineering techniques like PCA and L1 Regularisation. This model's excellent Accuracy of 97.63% demonstrated how adept it is at delivering precise predictions. With precision and recall scores of 96.77% and 97.88%, respectively, it demonstrates a remarkable capacity for positive instance classification and nearly complete capture of true positives. The Ensemble Methods F1-Score obtained an amazing 97.30%, highlighting its exceptional overall performance.

Table 4: Comparison of Predictive model with use of Feature Engineering Technique

Model	Accuracy (With FE)	Precision (With FE)	Recall (With FE)	F1-Score (With FE)	Accuracy (Without FE)	Precision (Without FE)	Recall (Without FE)	F1-Score (Without FE)
Logistic Regression	95.12	94.52	96.32	95.63	87.52	87.36	84.56	85.10
Naive Bayes	89.90	90.36	88.89	89.89	082.12	91.23	93.66	92.30
Ensemble Methods	97.63	96.77	97.88	97.30	93.14	93.74	95.23	94.11

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

Three predictive machine learning models Logistic Regression, Naive Bayes, and Ensemble Methods are thoroughly compared in Table 4 with and without the use of feature engineering techniques (labelled With FE and Without FE, respectively). Accuracy, Precision, Recall, and F1-Score are important performance metrics that show how effective a model is. When feature engineering techniques like Principal Component Analysis (PCA), SelectKBest for feature selection, and data normalisation (With FE) were added, the performance of logistic regression significantly improved.

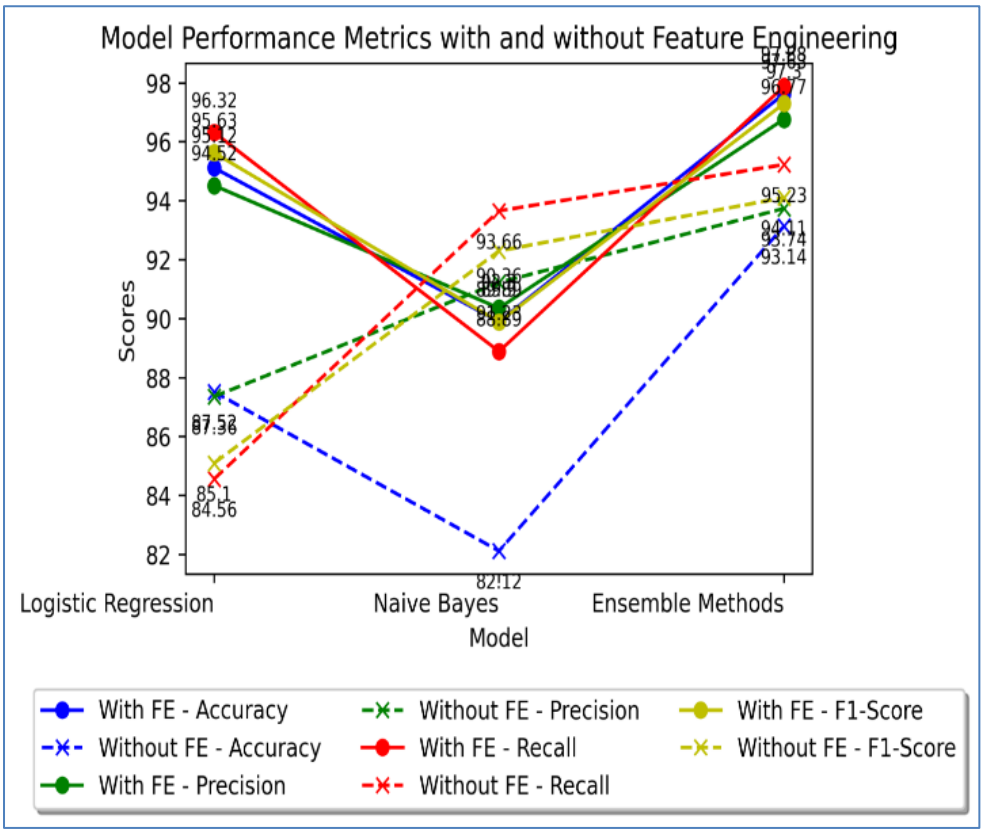

Figure 6: Comparative analysis of Predictive model

Its accuracy increased dramatically, reaching 95.12%, a significant improvement over the 87.52% obtained without feature engineering (Without FE). Significant gains were seen in Precision (94.52%) and Recall (96.32%), indicating better classification of positive events and genuine positives being captured. The F1-Score was an excellent 95.63%, exceeding the 85.10% F1-Score obtained without the use of feature engineering and highlighting an overall performance that was well-balanced. Naive Bayes demonstrated an accuracy of 89.90% when feature engineering techniques like TF-IDF and feature scaling were used (With FE), which is a significant improvement above the accuracy of 82.12% attained without feature engineering (Without FE). Despite showing progress, Precision (90.36%) and Recall (88.89%) still fell short of the Without FE scenario's results of 91.23% and 93.66%, respectively. However, the F1-Score for Naive Bayes (With FE) was 89.89%, indicating an overall performance that was balanced. F1-Score without feature engineering was 92.30%.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages: 201 – 215 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.88

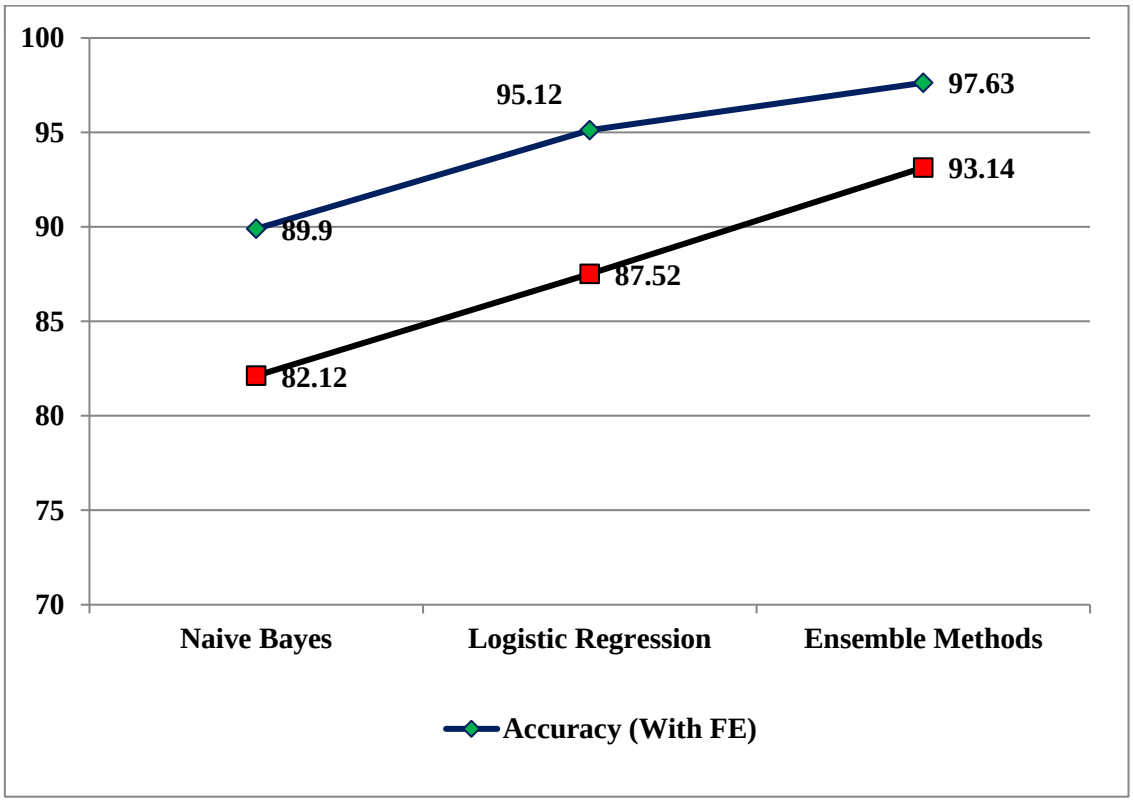

Figure 7: Accuracy comparison with FE and Without FE

Ensemble Methods performed exceptionally well when used with feature engineering techniques like L1 Regularisation and PCA (With FE). The Accuracy greatly outperformed the 93.14% attained without feature engineering (Without FE), increasing to an astonishing 97.63%. Both Precision (96.77%) and Recall (97.88%) showed impressive gains, demonstrating the model's skill at correctly classifying positive cases and identifying genuine positives. The model performed exceptionally overall, as seen by the F1-Score's remarkable performance (97.30%). The F1-Score was 94.11% in the absence of feature engineering. Overall, feature engineering significantly improved Accuracy, Precision, Recall, and F1-Score, which had a positive effect on model performance. Particularly, Logistic Regression and Ensemble Methods showed notable increases, highlighting the crucial position that feature engineering plays in improving predictive abilities. Although Naive Bayes also benefited, the enhancement was a little less significant. These findings highlight the significance of choosing the right feature engineering approaches to maximise model performance across a range of predictive tasks.

7. Conclusion

This study explored feature engineering techniques and their significant influence on improving predictive models in the field of data science. The performance of predictive models can be dramatically impacted by feature engineering, a crucial step in the machine learning process. This study set out to investigate various feature engineering methods and the range of algorithms and datasets to which they may be applied. Several important conclusions came to light throughout the inquiry. The performance of predictive models across a variety of machine learning methods was consistently significantly improved by the application of feature engineering, which comes first. The importance of feature engineering in enhancing model outcomes was highlighted by these improvements in measures including accuracy, precision, recall, and F1-score. The research discovered a wide variety of feature engineering techniques, from scaling and normalising numerical data to word embeddings and TF-IDF enabling the extraction of meaningful information from text data. Through the use of moving averages and lag features, feature engineering also benefited time series data. The study also emphasised the significance of comprehending the dataset's unique

properties as well as the requirements of the algorithm when choosing the proper feature engineering methods. Not every technique can be used in every situation, and the context might affect a technique's efficiency. The study also looked into feature engineering trade-offs, such as the potential for overfitting and the computational difficulty of some methods. In order to attain the best outcomes, researchers and practitioners should balance model generalisation with feature engineering complexity. Feature engineering is a powerful tool in the data scientist's toolbox that may be used to uncover hidden patterns and enhance the accuracy of prediction models. Its influence extends beyond algorithmic limits, providing insightful information and improved forecasts across a range of fields. Exploring and improving feature engineering techniques will continue to be a crucial field of research as data science develops, ensuring the progress of predictive modelling capabilities.

References

[1] J. Chen, P. Song, C. Zhao and J. Ding, "Spatiotemporal Multiscale Correlation Embedding With Process Variable Reorder for Industrial Soft Sensing," in IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-10, 2023, Art no. 2521410, doi: 10.1109/TIM.2023.3293567.

[2] Y. Zhao, Y. Cai and Q. Song, "Energy Control of Plug-In Hybrid Electric Vehicles Using Model Predictive Control With Route Preview," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 12, pp. 1948-4854, December 2021, doi: 10.1109/JAS.2017.7510889.

[3] L. Ren, Z. Meng, X. Wang, R. Lu and L. T. Yang, "A Wide-Deep-Sequence Model-Based Quality Prediction Method in Industrial Process Analysis," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3721-3731, Sept. 2020, doi: 10.1109/TNNLS.2020.3001602.

[4] U. Khurana, D. Turaga, H. Samulowitz and S. Parthasrathy, "Cognito: Automated Feature Engineering for Supervised Learning," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016, pp. 1304-1307, doi: 10.1109/ICDMW.2016.0190.

[5] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," SoutheastCon 2016, Norfolk, VA, USA, 2016, pp. 1-6, doi: 10.1109/SECON.2016.7506650.

[6] H. Rauf, M. S. Gul, M. Khalid and N. Arshad, "Smart Feature Selection-Based Machine Learning Framework for Calendar Loss Prediction of Li-Ion Electric Vehicle Battery," 2023 12th International Conference on Renewable Energy Research and Applications (ICRERA), Oshawa, ON, Canada, 2023, pp. 300-303, doi: 10.1109/ICRERA59003.2023.10269362.

[7] T. Xu, M. -s. Tsui and D. M. Chiu, "Why Hong Kong Ranks Highest in Life Expectancy: Looking for Answers from Data Science and Social Sciences," in Journal of Social Computing, vol. 3, no. 3, pp. 250-261, September 2022, doi: 10.23919/JSC.2022.0009.

[8] D. A. Jdanov, V. M. Shkolnikov, A. A. van Raalte and E. M. Andreev, "Decomposing current mortality differences into initial differences and differences in trends: The contour decomposition method", Demography, vol. 54, pp. 1579-1602, 2017.

[9] W. Lutz and E. Kebede, "Education and health: Redrawing the Preston curve", Population and Development Review, vol. 44, no. 2, pp. 343-361, 2018.

[10] R. Chetty, M. Stepner, S. Abraham, S. Lin, B. Scuderi, N. Turner, et al., "The association between income and life expectancy in the United States 2001–2014", Jama, vol. 315, no. 16, pp. 1750-1766, 2016.

[11] G. J. Xu, K. Xu and Q. Q. Lu, "Does the extension of life expectancy promote economic growth-An Empirical Analysis Based on 121 economies (in Chinese)", Economic Theory and Business Management, vol. 41, no. 3, pp. 97-112, 2021.

[12] A. G. Costa, E. Queiroga, T. T. Primo, J. C. B. Mattos and C. Cechinel, "Prediction analysis of student dropout in a Computer Science course using Educational Data Mining," 2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO), Loja, Ecuador, 2020, pp. 1-6, doi: 10.1109/LACLO50806.2020.9381166.

[13] J. Wu, N. Li and Y. Zhao, "Missing data filling based on the spectral analysis and the Long Short-Term Memory network," 2021 International Symposium on Computer Technology and Information Science (ISCTIS), Guilin, China,

2021, pp. 198-202, doi: 10.1109/ISCTIS51085.2021.00049.

[14] Y. -T. Cheng, B. -C. Shia, J. -Y. Kuo and H. -R. Yang, "Data Systematic Purifying Analysis in Data Mining," 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 287-290, doi: 10.1109/CSIE.2009.908.

[15] P. Wazurkar, R. S. Bhadoria and D. Bajpai, "Predictive analytics in data science for business intelligence solutions," 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), Nagpur, India, 2017, pp. 367-370, doi: 10.1109/CSNT.2017.8418568.

[16] S. Subhashini and R. Maruthi, "A Predictive Model for Road Traffic Data Analysis and Visualization to Detect Accident Zones," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1227-1231, doi: 10.1109/ICACCS57279.2023.10112862.

[17] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts methods and analytics", International Journal of Information Management, vol. 35, no. 2, pp. 137-44, April 2015.

[18] G.S. Tomar, N.S. Chaudhari, R.S. Bhadoria and G.C. Deka, The Human Element of Big Data: Issues Analytics and Performance, FL:CRC Press, Sept. 2016.

[19] M. Bichler, A. Heinzl and W. M. P. Van Der Aalst, "Business Analytics and Data Science: Once Again", Business & Information Systems Engineering, vol. 59, no. 2, pp. 77-79, April 2017.

[20] A. Abbasi, S. Sarker and R.H. Chiang, "Big Data Research in Information Systems: Toward an Inclusive Research Agenda", Journal of the Association for Information Systems, vol. 17, no. 2, Feb 2016.

[21] V. Dhar, "Data science and prediction", Communications of the ACM, vol. 56, no. 12, pp. 64-73, Dec 2013.

[22] D. E. Brown, A. Abbasi and R. Y. K. Lau, "Predictive analytics: Predictive modeling at the micro level", IEEE Intelligent Systems, vol. 30, no. 3, pp. 6-8, May 2015.

[23] M. Swarnkar and R.S. Bhadoria, "Analysis for Security Attacks in Cyber-Physical Systems. In Cyber-Physical Systems" in A Computational Perspective, FL:CRC Press, pp. 489-514, Oct 2015.

[24] M.A. Waller and S. E. Fawcett, "Data Science Predictive Analytics and Big Data: A Revolution That Will Transform Supply Chain Design and Management", Journal of Business Logistics, vol. 34, no. 2, pp. 77-84, June 2013.

[25] G. Shmueli and O.R. Koppius, "Predictive analytics in information systems research", MIS Quarterly, vol. 35, no. 3, pp. 553-572, Sept. 2011.

[26] Dipanshu Gupta, Vagisha Goel, Rithik Gupta, Mohd Shariq and Rajesh Singh, "ROAD ACCIDENT PREDICTOR USING MACHINE LEARNING", International Research Journal of Modernization in Engineering Technology and Science, vol. 04, no. 05, May 2022.

[27] Anik Vega Vitianingsih, Nanna Suryana and Zahriah Othman, "Spatial analysis model for traffic accident-prone roads classification: a proposed framework", IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, no. 2, June 2021.

[28] D. Santos, J. Saias, P. Quaresma and V.B. Nogueira, "Machine Learning Approaches to Traffic Accident Analysis and Hotspot Prediction", Computers, vol. 10, pp. 157, 2021,

[29] Syed Saqib Ali Kazmi, Mehreen Ahmed, Rafia Mumtaz and Zahid Anwar, "Spatiotemporal Clustering and Analysis of Road Accident Hotspots by Exploiting GIS Technology and Kernel Density Estimation", The Computer Journal, vol. 65, no. 2, pp. 155-176, February 2022, [online] Available: https://doi.org/10.1093/comjnl/bxz158.

[30] Asghar Pasha Vijayalakshmi, MD Atique, MD Hussain, Harsh narnot and Bipin, "Road Accident Prediction using Machine Learning", International Research Journal of Engineering and Technology (IRJET), vol. 08, no. 07, July 2021.

[31] Khanh Giang Le, Pei Liua and Liang-Tay Lin, "Determining the road traffic accident hotspots using GIS-based temporal-spatial statistical analytic techniques in Hanoi", GEO-SPATIAL INFORMATION SCIENCE, vol. 23, no. 2, pp. 153-164, 2020,

[32] Jamshid Sodikov, "Road Traffic Accident Data Analysis and Visualization in R", International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)., vol. 8, pp. 25-32, 2018.