Uncovering Patterns in Stack Exchange Users A Data Mining Perspective

Arlene D’costa, Samyuktha Chaparla, Sohan Thakur and Krunal Gaikwad
aadcosta@syr.edu, schaparl@syr.edu, sothakur@syr.edu , kgaikwad@syr.edu

***Abstract* - "Uncovering Patterns in Stack Exchange Users - A Data Mining Perspective" delves into the analysis of Stack Exchange data since 2020 to discern prevalent topics and technological discussions within the community. Leveraging advanced topic modeling techniques such as Latent Dirichlet Allocation (LDA) and BERTopic, the project identifies key themes and interests among users. By employing predictive analytics, including linear regression, neural networks, and random forests, the study forecasts the value and engagement of posts based on upvote scores. This dual approach offers insights into current trends, facilitates strategic decision-making, and fosters community growth by enhancing user experience and content relevance. Through dynamic understanding and continuous improvement, the project contributes to business and educational domains by optimizing engagement and fostering strategic community development.**

Index Terms - Topic Modeling, LDA, BERTopic, Clustering unlabelled Textual Data, Data Technology Topic Analysis, LDA vs BERTopic , Upvote Score Prediction, Neural Network, Linear Regression, Random Forest Regressor

Introduction

Uncovering Patterns in Stack Exchange Users - A Data Mining Perspective focuses on analyzing different content of posts, questions and different answers, comments, titles, etc from the stack exchange dataset to identify the most discussed topic about technologies since 2020. This is accomplished through the use of advanced topic modeling techniques. Further goal is to use data-driven insights to thoroughly analyze user behavior on Stack Exchange and provide suggestions for improving the platform's efficiency and user experience. This makes it particularly valuable for researchers, IT developers, students and analysts looking to understand trends, behaviors, and expertise within various domains.[5]

This data is organized in XML format and undergoes processing and analysis to discern significant patterns and insights. The project aims to employ this data to conduct topic modeling and forecast the worth of posts using indicators of community interaction, such as upvotes. This strategy seeks to pinpoint and predict the most valuable discussions, offering a deeper understanding of user preferences and the intrinsic value of the content.

Topic modeling, specifically using Latent Dirichlet Allocation (LDA) and BERTopic techniques, is employed to identify the key technologies and topics discussed on Stack Exchange since 2020. These methods are effective in revealing the dominant themes and popular topics within the community, reflecting the main interests and technological dialogues among users.

The project aims to evaluate Stack Exchange posts to determine their potential benefit to the community. This is done by predicting which posts will be most valuable based on scores from upvotes. The prediction models used include linear regression, neural networks, and random forests, which assess the relationship between the content of the posts and the engagement they receive.

By combining topic modeling and predictive analytics, the project seeks to not only understand the current interests and challenges faced by the Stack Exchange community but also to forecast the future relevance and usefulness of posts. This dual approach allows for a dynamic understanding of content value, helping to enhance user engagement and satisfaction. Here we are using modeling techniques[7] like LDA, BERTopic,etc. This helps to identify hidden structures in the vast amount of data, categorizing them into topics based on the similarity of their content. Further we perform hierarchical clustering to discover and organize discussions into coherent topics through BERTopic. Dynamic Understanding and Continuous Improvement by optimizing user experience can be achieved through prediction of upvote scores based on post relevancy. This project focuses on enhancing business and educational implications by upgrading strategic decision making and community Growth.

Literature Review

The study of online behavior and discourses within communities is a trending research area of our time. Platforms like Stack Exchange, which serve as the hub for varied technical discussions spanning across different disciplines, have been the center of our research. This literature review collects the latest information and research on user behavior, the dynamics of topics, and the quality of content within the context of Stack Exchange communities.

1.User Behavior Analysis:

In their state-of-the-art study, Mondal et al. (BSEC 2023) dived into the span of technology consumption by exploring the question-and-answer traces of users on Stack Overflow. They applied complex algorithms[1] such as sequence mining and clustering to reveal patterns in user interactions, hence being able to outline a very rudimentary adoption and engagement scheme within the community. As a result of Mondal et al's work in pinpointing repeated patterns and classifying users based on their behavior, the authors presented insights into the changing nature of tech usage among Stack Exchange members.

2.Topic Dynamics and Content Quality:

Huang et al. (ASE) published revolutionary ways of extracting technology differences from discussions carried through large-scale online platforms. Through the utilization of natural language processing and machine learning algorithms, they were able to categorize conversations that related to specific technologies, leading to a deeper comprehension of the constantly changing technological landscape on Stack Exchange.[2] Their method, which was centered on detecting the small fluctuations in technology conversation, let it be an extra tool that complemented the traditional topic modeling techniques and made it possible for a better understanding of the community dynamics.

3.Content Analysis and Quality Assessment:

Upadhyay et al. (WSDM 2016) talked about crowd-learning dynamics and the power of community knowledge in the online education sector. Through the process of looking at user interactions and content contributions, they aimed to get to the bottom of how knowledge travels through the community and goes from one person to the next.[3] The outputs of their findings were able to expose the processes that drive crowdsourcing and emphasize the dissemination of information as the most vital for community growth and engagement.

4.Named Entity Recognition (NER) and Semantic Content Analysis:

In the case of Ye et al. (SANER 2016), software-specific named entity recognition in social media content from software engineering was the subject matter. Their work focused on the identification and extraction of software-specific entities found in the online conversation, which in turn made it easier to perform a more thorough analysis of software-related topics and discussion threads.[4] By establishing NER algorithms in the software engineering domain, they assisted with substantial content analysis and gave important information about the amount and context of software-related discussion forums within the online communities.

5.Comparative Analysis and Model Evaluation:

In ESEM 2016, Chen and Xing executed a competitive examination of technology scenery acquisition from Stack Overflow conversations. Their research was centered around the identification and analysis of aspects concerning technology and trends in posts made on Stack Overflow. Through using advanced text mining techniques, they figured out what the widely used technologies are and their interconnections among them, contributing to a complete picture of the technological landscape at Stack Exchange.

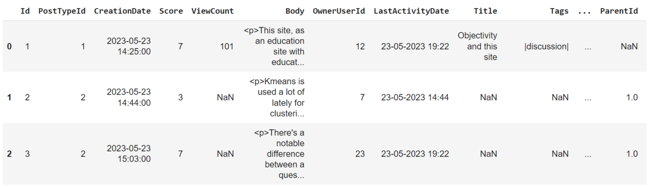
The works reviewed introduce a great variety of methods and techniques used to study the behavior and dynamics of both users and topics as well as the quality of contributions on Stack Exchange communities.[5] The researchers have used several types of techniques such as sequence mining, clustering, natural language processing, and named entity recognition, to reveal patterns in community dynamics, technological trends, and knowledge diffusion processes. Going forward, considering the use of sophisticated means such as transfer learning and model fine-tuning may lead to an even deeper understanding of online community interplay that would contribute to the creation of efficient strategies.

Methodology

I. Data Gathering

The Stack Exchange data, acquired from **Internet Archive**, is an invaluable resource for accessing historical data from various Stack Exchange websites[6], including Stack Overflow, Mathematics Stack Exchange, Cross Validated, and many others. This data is available in the form of XML files containing posts, comments, user information, and other relevant data.

Random Stack Exchange Posts Dataset has been selected from the Internet archive containing different attributes such as Id, PostTypeId, CreationDate, Score, ViewCount, Body,OwnerUserId, LastActivityDate, Title, Tags, AnswerCount,CommentCount, ContentLicense, ParentId, LastEditorUserId,LastEditDate, FavoriteCount, AcceptedAnswerId,OwnerDisplayName,LastEditorDisplayName, CommunityOwnedDate, ClosedDate,DeletionDate.



	Id	PostTypeId	CreationDate	Score	ViewCount	Body	OwnerUserId	LastActivityDate	Title	Tags	...	ParentId
0	1	1	2023-05-23 14:25:00	7	101	<p>This site, as an education site with educat...	12	23-05-2023 19:22	Objectivity and this site	|discussion|	...	NaN
1	2	2	2023-05-23 14:44:00	3	NaN	<p>Kmeans is used a lot of lately for clusteri...	7	23-05-2023 14:44	NaN	NaN	...	1.0
2	3	2	2023-05-23 15:03:00	7	NaN	<p>There's a notable difference between a ques...	23	23-05-2023 19:22	NaN	NaN	...	1.0

FIG I :XML Data Converted To Pandas Dataframe

In this report, we aim to analyze the content of posts, including questions and answers,comments and title from the Stack Exchange dataset using the Body(String) column in the dataset. By employing topic modeling techniques such as Latent Dirichlet Allocation (LDA) or BERTopic, our goal is to uncover common themes and topics prevalent within the dataset. We also want to identify the most meaningful posts by predicting the upvote scores for posts using the Score(Numerical) column.Understanding these topics can provide valuable insights into the interests, concerns, and discussions taking place within the Stack Exchange community.

II. Data Preprocessing

Text Data Overview:

As part of our EDA process, we conducted an overview of the text data present in the Stack Exchange dataset. This involved examining the content of posts, including questions, answers, comments, and other textual information contributed by users. By gaining a comprehensive understanding of the text data, we were able to assess its quality, structure, and relevance for further analysis and modeling tasks.

In addition to exploring the content of the posts, we filtered the dataset to include only those entries with a creation date later than 2020. This decision was motivated by the need to focus our analysis on recent discussions and trends in the Stack Exchange community, particularly in the context of emerging technologies and current topics of interest. By narrowing our scope to post-2020 data, we aimed to capture the most up-to-date insights and identify recent technologies that are the focal points of discussion among users. This filtering process ensured that our analysis remained relevant and aligned with the latest developments in the field, enabling us to extract actionable insights and valuable findings from the dataset.

Text Preprocessing:

Prior to applying any analytical techniques, we performed text preprocessing to clean and prepare the text data for analysis. This included several steps:
Lowercasing: Converting all text to lowercase to ensure consistency in word representation.
Tokenization: Breaking down the text into individual words or tokens to facilitate further analysis.
Lemmatization: Reducing words to their base or root form to normalize variations (e.g., "running" -> "run").
Removing Stopwords: Eliminating common words such as "the," "is," and "and" that do not contribute significant meaning to the text.
Handling Special Characters and Numbers: Removing or replacing special characters, punctuation, and numerical digits that may not be relevant for analysis.
Removing URLs: Eliminating URLs or web links present in the text, as they do not convey meaningful information and can introduce noise.
'technicalcorpus.txt' is the file containing the precreated technical corpus, where each line contains a single technical term. We read the contents of the file into a set called technical_terms. We then tokenize the stack exchange text data and filter out only the tokens that match any term in the technical_terms corpus set, resulting in a filtered list of technical tokens. Finally, we assemble these filtered tokens into a single string representing the technical corpus.
This approach allows us to leverage existing knowledge of technical terms to filter out technology-related terms from your text data, helping us focus our analysis on relevant content.

Feature Engineering

- Prior to model training, missing values in the target variable (y_train) are imputed with the median method to assure completeness.
- Unigrams(Bag of words) model are used as tokens and have been passed to TF-IDF for vectorization
- The TfidfVectorizer transforms text features into TF-IDF representations (X_tfidf).

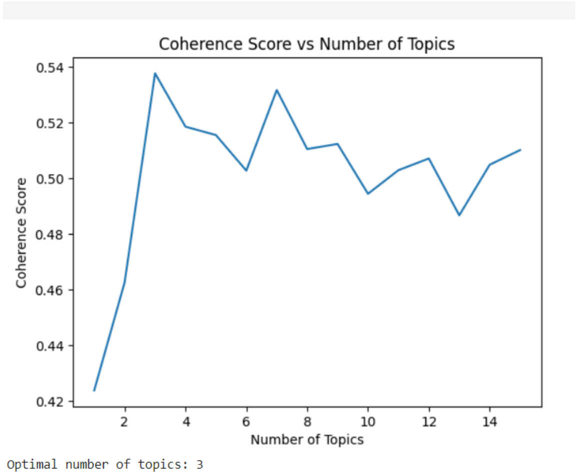
III. Topic Modeling

In our analysis, we employed **Latent Dirichlet Allocation (LDA)** and **BERTopic** to perform topic modeling on the Stack Exchange dataset.[7] These techniques allowed us to uncover latent topics within the text data and assign labels to previously unlabeled data points.

Latent Dirichlet Allocation (LDA):

LDA is a probabilistic context free generative model commonly used for topic modeling in text data. It assumes

that each document in the corpus is a mixture of various topics, and each word in the document is attributable to one of these topics. By applying LDA, we were able to identify clusters of words that frequently co-occur across documents, representing common themes and topics discussed within the Stack Exchange community. LDA provided us with a structured framework for organizing and understanding the textual content, enabling us to extract meaningful insights and identify prevalent topics of discussion.
We have calculated the optimal number of topics based on coherence score which turned out to be **3** for the given dataset.



```
Optimal number of topics: 3
```

FIG II: LDA COHERENCE SCORE VS NO OF TOPICS

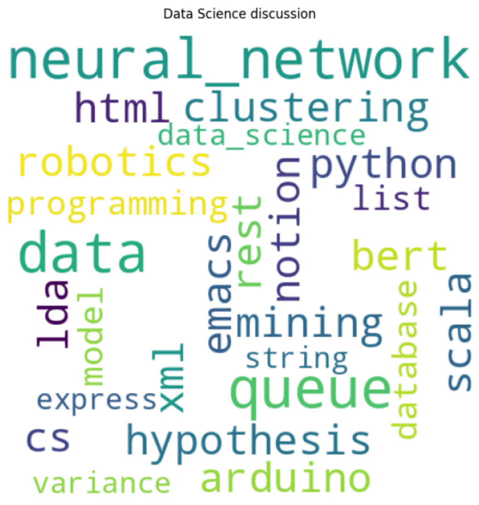
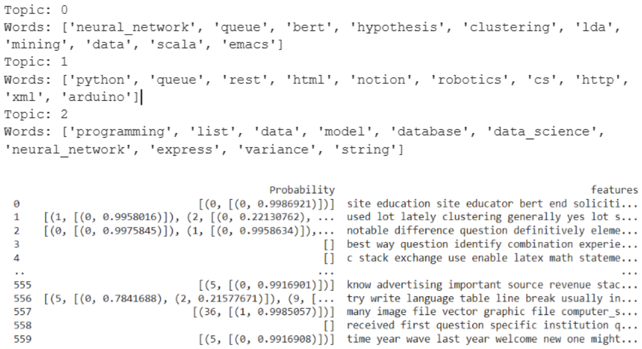


FIG III: DATASET WORDCLOUD USING LDA



```
Topic: 0
Words: ['neural_network', 'queue', 'bert', 'hypothesis', 'clustering', 'lda',
'mining', 'data', 'scala', 'emacs']
Topic: 1
Words: ['python', 'queue', 'rest', 'html', 'notion', 'robotics', 'cs', 'http',
'xml', 'arduino']
Topic: 2
Words: ['programming', 'list', 'data', 'model', 'database', 'data_science',
'neural_network', 'express', 'variance', 'string']

                                         Probability                                            features
0                            [(0, [(0, 0.9986921)])]  site education site educator bert end soliciti...
1    [(1, [(0, 0.9958016)]), (2, [(0, 0.22130762), ...  used lot lately clustering generally yes lot s...
2    [(0, [(0, 0.9975845)]), (1, [(0, 0.9958634)]),...  notable difference question definitively eleme...
3                                                   []  best way question identify combination experie...
4                                                   []  c stack exchange use enable latex math stateme...
..                                                 ...                                                ...
555                          [(5, [(0, 0.9916901)])]  know advertising important source revenue stac...
556  [(5, [(0, 0.7841688), (2, 0.21577671)]), (9, [...  try write language table line break usually in...
557                         [(36, [(1, 0.9985057)])]  many image file vector graphic file computer_s...
558                                                 []  received first question specific institution q...
559                          [(5, [(0, 0.9916908)])]  time year wave last year welcome new one might...
```

FIG IV: LDA DOCUMENT PREDICTION

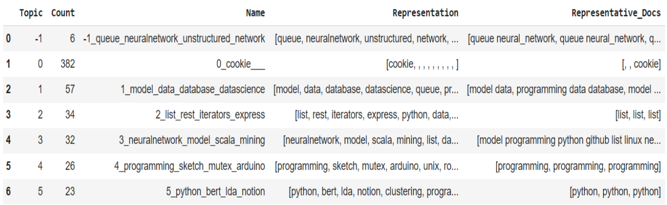
LDA result Analysis:

Based on the identified topics and their corresponding words, it appears that the topics are related to data science and programming. Here's a breakdown of the analysis:

Topic 0: This topic contains words such as *neural_network, clustering, lda,* and *data,* which are commonly associated with machine learning and data mining. The presence of terms like "scala" and "emacs" may indicate discussions related to programming tools and environments.
Topic 1: The words in this topic include *python,html, http*, and *arduino*, suggesting discussions on programming languages, web development, and robotics. Terms like *notion* and *xml* could also be related to software development and data interchange formats.
Topic 2: This topic includes terms like *programming, database, data science*, and *neural network*, indicating discussions on programming concepts, database management, and data science techniques. The presence of words like "list," "model," and "string" further supports the notion of discussions related to programming and data manipulation.

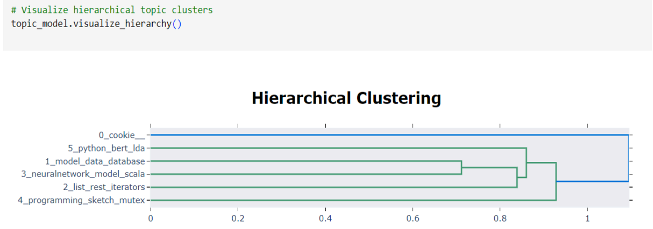
BERTopic:

BERTopic is a context-based topic modeling technique that leverages the BERT (Bidirectional Encoder Representations from Transformers) language model to generate document embeddings. These embeddings are then clustered using hierarchical clustering to identify coherent topics within the text data. BERTopic offers several advantages, including the ability to handle large datasets efficiently and capture complex semantic relationships between words.[9] By applying BERTopic, we were able to extract high-quality topics from the Stack Exchange dataset and assign labels to previously unlabeled data points based on their underlying topics.



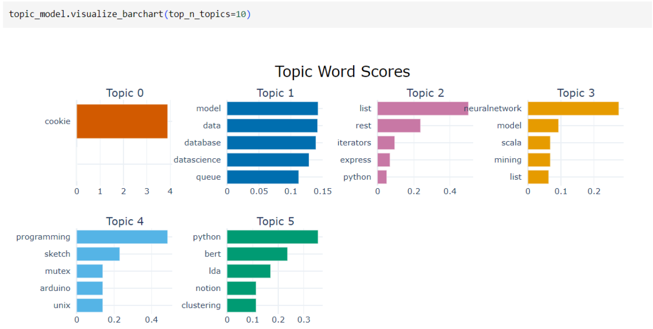
	Topic	Count	Name	Representation	Representative_Docs
0	-1	6	-1_queue_neuralnetwork_unstructured_network	[queue, neuralnetwork, unstructured, network, ...	[queue neural_network, queue neural_network, q...
1	0	382	0_cookie___	[cookie, , , , , , , , ,]	[, , cookie]
2	1	57	1_model_data_database_datascience	[model, data, database, datascience, queue, pr...	[model data, programming data database, model ...
3	2	34	2_list_rest_iterators_express	[list, rest, iterators, express, python, data,...	[list, list, list]
4	3	32	3_neuralnetwork_model_scala_mining	[neuralnetwork, model, scala, mining, list, da...	[model programming python github list linux ne...
5	4	26	4_programming_sketch_mutex_arduino	[programming, sketch, mutex, arduino, unix, ro...	[programming, programming, programming]
6	5	23	5_python_bert_lda_notion	[python, bert, lda, notion, clustering, progra...	[python, python, python]

FIG V: BERT TOPIC CLUSTERS INFORMATION



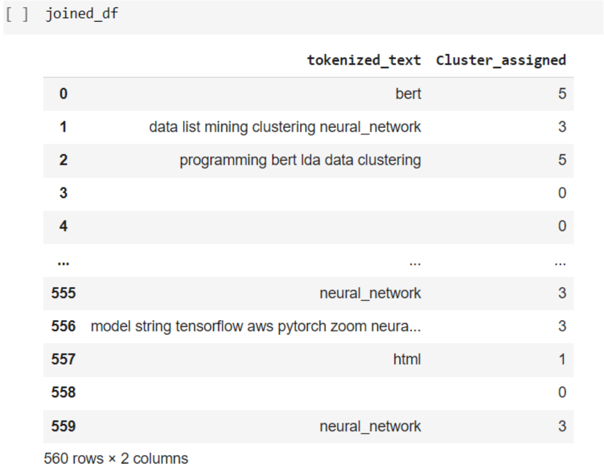
```
# Visualize hierarchical topic clusters
topic_model.visualize_hierarchy()
```

FIG VI: BERTopic Hierarchical Clustering



```
topic_model.visualize_barchart(top_n_topics=10)
```

FIG VII: BERTopic Word Scores



```
[ ]  joined_df
```

	tokenized_text	Cluster_assigned
0	bert	5
1	data list mining clustering neural_network	3
2	programming bert lda data clustering	5
3		0
4		0
...	...	...
555	neural_network	3
556	model string tensorflow aws pytorch zoom neura...	3
557	html	1
558		0
559	neural_network	3

560 rows × 2 columns

FIG VIII: BERTopic CLUSTER ASSIGNMENT

BERTopic Result Analysis:

Topic -1 : Keywords: queue, neural network, unstructured, network

Documents: A cluster ID of -1 in BERTopic represents documents that are not well-represented by any of the main clusters identified by the model and are often considered as noise or outliers. Documents within this topic contain discussions related to above keywords, albeit in a diverse or ambiguous context.

Topic 0: Keywords: cookie

Documents: This cluster appears to be focused solely on the topic of cookies. It might contain discussions related to web cookies

Topic 1: Keywords: model, data, database, datascience

Documents: This cluster likely represents topics related to data modeling, databases, and data science. It covers various aspects of handling and analyzing data, including modeling techniques and database management.

Topic 2: Keywords: list, rest, iterators, express, python, data

Documents: This cluster seems to revolve around programming topics, particularly in Python, involving lists, iterators, RESTful APIs, and data processing. It's a technical cluster focusing on programming concepts and data handling.

Topic 3: Keywords: neural network, model, scala, mining, list, data

Documents: This cluster might involve discussions related to neural networks, machine learning models, data mining, and possibly using Scala programming language. It overlaps with Cluster 0 but seems to have a stronger focus on data mining and machine learning models.

Topic 4: Keywords: programming, sketch, mutex, arduino, unix

Documents: This cluster encompasses discussions related to programming concepts such as sketches, mutex (mutual exclusion), Arduino microcontroller programming, and Unix operating system. It's a technical cluster focused on programming and embedded systems.

Topic 5: Keywords: python, bert, lda, notion, clustering, programming

Documents: This cluster appears to be centered around the Python programming language and various related topics such as BERT (Bidirectional Encoder Representations from Transformers), LDA (Latent Dirichlet Allocation), Notion (possibly referring to the productivity software), and clustering algorithms. It's a technical cluster with a focus on Python programming and related tools/algorithms.

Comparison of LDA and BERTopic:

Clusters of **BERTopic provide better semantic insights** and **more variation in capturing topics** due to following reasons:

- BERTopic is context based while LDA is context free and operates on frequency of words.
- Improved topic coherence
- Robustness to noise
- Reduced sensitivity to hyperparameters
- Support for hierarchical modeling make it interpretable and better choice over LDA for many text clustering tasks

IV. *Data Modeling and Prediction*

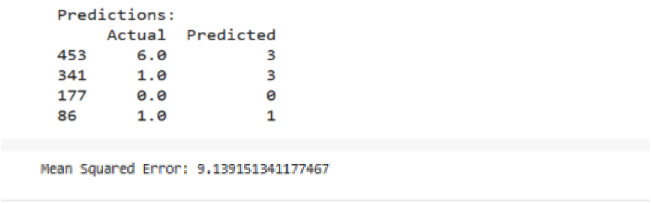
In our research to predict upvotes in Stack Overflow using text mining techniques, three different modeling approaches have been investigated namely: linear regression, neural network, and random forest. Each of these models has distinct strengths and concerns that are relevant to the complex character of our dataset. Linear Regression, which is well-known for its simplicity and interpretability, provides a natural starting point for investigating the linear relationship between text attributes and upvotes.Meanwhile, Neural Networks explore nonlinearity, utilizing their ability to detect complicated patterns and hierarchical representations in text data. Random Forest, on the other hand, takes advantage of ensemble learning to excel at handling high-dimensional feature spaces and generating solid predictions by aggregating numerous decision trees. Through this comprehensive study, we aim to discover the most effective approach for improving user engagement and content quality on the Stack Overflow platform.

To assist model training and evaluation, the dataset has been divided into three sets: training, validation, and testing. The training set is further partitioned to save a portion for validation.

Sparse TF-IDF matrices are converted to dense arrays (X_train_dense, X_val_dense, and X_test_dense) to make them compatible with neural networks.

1. Linear Regression

The simplicity of linear regression lies in its basic modeling of the relationship between independent variables (in this case, text attributes) and the dependent variable (upvote scores). The model estimates the coefficients for each attribute, indicating the magnitude and direction of their impact on the target variable, based on a linear relationship assumption. This interpretability is crucial for determining which components of text content contribute the most significantly to upvote values, giving practical insights for content creators and platform administrators alike. Furthermore, linear regression's ease of implementation and computational efficiency make it ideal for analyzing huge datasets, such as Stack Overflow's extensive text corpus, without compromising performance or scalability.



```
Predictions:
     Actual  Predicted
453     6.0          3
341     1.0          3
177     0.0          0
86      1.0          1

Mean Squared Error: 9.139151341177467
```

FIG IX: LINEAR REGRESSION MODEL RESULTS

2.Neural Networks

Neural networks are ideal for processing text data because of the ability they have to recognize complex, non-linear relationships that exist in language. Neural networks excel in learning detailed patterns and semantic representations from text by layering interconnected neurons, allowing them to make accurate predictions and draw valuable insights. Their adaptable architecture enables the use of specialized layers and techniques for text processing, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), which are designed to handle sequential and spatial patterns in text, respectively.Furthermore, neural networks can use word embeddings, which are dense vector representations of words, to encode semantic similarity and increase model performance. Furthermore, the availability of pre-trained models such as BERT and GPT facilitates transfer learning, allowing neural networks to use information from vast text corpora and adapt it to new tasks using minimum data and computational resources. Collectively, these qualities make neural networks an effective tool for natural language processing, sentiment analysis, machine translation, and a variety of other text-related applications. With appropriate regularization techniques and hyperparameter tuning, neural networks can generalize well to unseen data, making them suitable for tasks where the model needs to perform well on new, unseen examples.

Model Architecture:

- Using the Keras Sequential API, we create a neural network model. It consists of densely connected layers with rectified linear unit (ReLU) activation functions, which allow for nonlinear transformations of input features.
- To reduce overfitting, dropout layers are added after each thick layer at a rate of 0.7 and 0.5, respectively.
- The dense layers are regularized by L2 regularization at a regularization strength of 0.001,

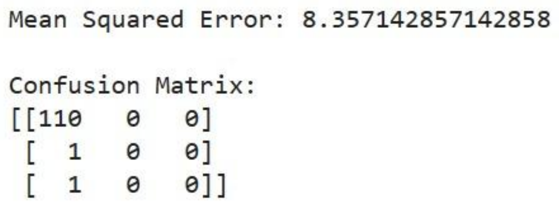
which promotes model generalization and reduces overfitting.

Model Compilation and Training:

- The model is built with the Adam optimizer and the mean squared error (MSE) loss function, with mean absolute error (MAE) serving as the evaluation metric.
- Early stopping is used for 10 epochs to monitor validation loss and end training when performance stagnates, reducing overfitting.
- Training takes place across 120 epochs with a batch size of 32, using training data (X_train_dense, y_train_imputed) and validating against validation data (X_val_dense, y_val_imputed).

Prediction and evaluation:

- The trained neural network model is used to predict upvote scores using the test data (X_test_dense).
- Predicted upvote scores are adjusted to the nearest integer in order to reflect the discontinuous nature of upvotes.
- Model performance is evaluated using the **mean squared error (MSE)** between expected and real upvote scores, which provides a quantitative measure of prediction accuracy.



```
Mean Squared Error: 8.357142857142858

Confusion Matrix:
[[110   0   0]
 [  1   0   0]
 [  1   0   0]]
```

FIG X: NEURAL NETWORK MODEL RESULTS

3.Random Forest Model Construction and Optimization Pipeline

This section delves into the development of a strong prediction model for upvote scores in Stack Exchange postings, using a comprehensive pipeline that smoothly blends preprocessing and modeling phases. The procedure begins with TF-IDF vectorization, which transforms textual information into numerical representations that are compatible with machine learning techniques. The dataset is then divided into separate training and testing sets to assist model evaluation.[8] The heart of our process is a rigorously developed pipeline, carefully crafted to enhance model performance. This pipeline includes critical phases such as feature scaling with MaxAbsScaler to standardize feature magnitudes and imputation of missing values with the median technique via SimpleImputer to assure data completion. Feature selection is then carried out using SelectKBest, which uses the F-regression score function to keep the most informative features, improving model efficiency and interpretability.Finally, the Random Forest Regressor, a powerful ensemble learning algorithm capable of capturing complicated associations in data, is used to accurately forecast upvote scores. Hyperparameters are methodically tweaked using grid search cross-validation to get the ideal configuration, which improves the model's predictive performance. Evaluation on the test set, defined by mean squared error, provides a full assessment of the model's ability to capture the intricacies of upvote score prediction.

The pipeline consists of several stages:

MaxAbsScaler: This step scales each feature to the range [-1, 1] by dividing it by its highest absolute value. Scaling is critical for models such as Random Forest, which are sensitive to the scale of input features.

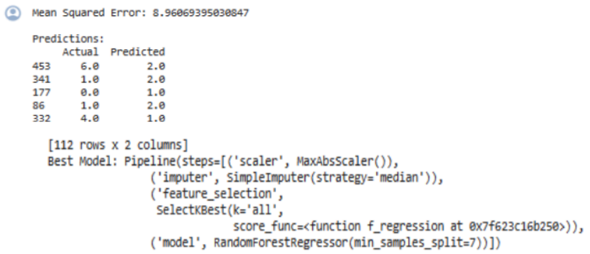
SimpleImputer: Missing data values are imputed using the median method. Imputation ensures that the dataset is full and ready to model.

SelectKBest: This stage uses the f_regression scoring function to select the top k features based on their F-score (a measure of feature relevance). Selecting relevant features can increase model performance while lowering computational overhead.

RandomForestRegressor: This stage depicts the Random Forest model, which is trained on the chosen features to predict upvote scores. Random Forest is a powerful ensemble learning method that aggregates predictions from numerous decision trees, resulting in robust forecasts and good handling of nonlinear relationships in the data.

The param_grid dictionary defines hyperparameters for tuning, such as the number of features to select (feature_selection__k), the number of trees in the forest (model__n_estimators), the maximum depth of each tree (model__max_depth), and the minimum number of samples needed to split a node. Grid search cross-validation (GridSearchCV) is used to find the optimal combination of hyperparameters and optimize the model's performance using negative mean squared error.[8]

Following grid search, the best model is evaluated on the test set, and its performance is measured using mean squared error. Finally, the predictions are generated, including the actual and expected upvote ratings for comparison.



```
Mean Squared Error: 8.96069395030847

Predictions:
     Actual  Predicted
453     6.0        2.0
341     1.0        2.0
177     0.0        1.0
86      1.0        2.0
332     4.0        1.0

  [112 rows x 2 columns]
  Best Model: Pipeline(steps=[('scaler', MaxAbsScaler()),
                  ('imputer', SimpleImputer(strategy='median')),
                  ('feature_selection',
                   SelectKBest(k='all',
                               score_func=<function f_regression at 0x7f623c16b250>)),
                  ('model', RandomForestRegressor(min_samples_split=7))])
```

FIG XI: RANDOM FOREST REGRESSOR MODEL RESULTS

Results And Discussion

On comparing the performance of Linear Regression, Neural Networks, and Random Forest for predicting upvote ratings, we found significant differences, particularly for our stack exchange text dataset, **Linear regression**, a simple and interpretable model, tries to fit a straight line to the data. However, text data frequently contains hidden patterns and relationships that a linear model cannot fully capture. This limitation is reflected in the **mean squared error (MSE) score of 9.14,** which indicates moderate prediction error.

In comparison, **Neural Networks** have more flexibility and can identify complicated patterns within text data. Neural networks are particularly good at capturing non-linear interactions and hierarchical structures because neural networks are probabilistic in nature. This adaptability is shown in its lower **MSE score of 8.36**, which demonstrates its capacity to better understand the nuances of the text dataset when compared to linear regression.

Random Forest is more advanced than Linear Regression, although it performs somewhere in the middle. It works as a decision tree team, making predictions using ensemble learning. While Random Forest is better at handling non-linear correlations than Linear Regression it functions efficiently when feature space can be logically split into rectangular regions , it may fail to capture the complex patterns seen in text data as successfully as Neural Networks. This is noticeable from its **MSE score of 8.96**, which is greater than Neural Networks but lower than Linear Regression.

Conclusion

The study throughout our analysis of Stack Exchange of data with BERTopic and LDA, we found relevant patterns and clusters through BERTopic that could efficiently be used to identify major technological topics of discussions from the year 2020 till date which turned out to be neural networks model and python programming discussions focusing more on technologies involved in data science. Some discussions were also drawn towards arduino, session and cookies, mutex and unix programming. Through LDA, we could capture technologies that were part of major discussions on a broader perspective throughout the dataset while BERTopic helped in further segregation of topics into clusters capturing sub-domains within technologies.

Although we weren't able to identify the targeted age group indulged in such discussions due to lack of adequate data for analysis, we used the textual data to predict the upvote scores for the discussion posts using 3 different models namely Linear Regression, Neural Network and Random Forest performing comparative analysis. Data was noisy and consisted of non-linear complex patterns efficiently captured by neural networks providing the best results among the 3 yielding in lowest MSE.

Future work

A limited corpus for identifying the important keywords for topic modeling has been used, however Named Entity Recognition strategy from NLP can be used for capturing technical terms.

Models pre-trained on technical terms(Transfer learning) can be used for better and more accurate results.

Fine-tuning of ML models and setting up better hyperparameters can be used.

Data cleansing and preprocessing needs to be done efficiently and multiple datasets can be used for topic modeling and training ML models.

Since most of the data was in XML, a better parser for eliminating redundant data can be used for performing analysis and prediction.

References

[1] Surveys, 50(3), Article 35, 1–37. doi:10.1145/3068281 Saikat Mondal, Debajyoti Mondal, Chanchal K. Roy. 'Investigating Technology Usage Span by Analyzing Users' Q&A Traces in Stack Overflow'. 30th Asia-Pacific Software Engineering Conference (APSEC 2023)

[2] Yi Huang, Chunyang Chen, Zhenchang Xing, Tian Lin and Yang Liu. 'Tell Them Apart: Distilling Technology Differences from Crowd-Scale Comparison Discussions'. The 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE).

[3] Utkarsh Upadhyay, Isabel Valera, Manuel Gomez-Rodriguez. 'Uncovering the Dynamics of Crowdlearning and the Value of Knowledge'. 10th ACM International Conference on Web Search and Data Mining Conference (WSDM 2016)

[4] Chunyang Chen, Zhenchang Xing.' Mining Technology Landscape from Stack Overflow'. The 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2016)

[5] Deheng Ye, Zhenchang Xing, Chee Yong Foo, Zi Qun Ang, Jing Li, and Nachiket Kapre. 'Software-specific Named Entity Recognition in Software Engineering Social Content'. The 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2016)

[6] https://archive.org/download/stackexchange

[7] L. E. George and L. Birla, "A Study of Topic Modeling Methods," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 109-113, doi: 10.1109/ICCONS.2018.8663152. keywords: {Computational modeling;Semantics;Resource management;Analytical models;Probabilistic logic;Electronic mail;Conferences;Topic

modeling;Hidden Thematic Structure;Latent Semantic Analysis (LSA);Probabilistic Latent Semantic Analysis (PLSA);Latent Dirichlet Allocation (LDA)},

[8] J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, India, 2017, pp. 65-68, doi: 10.1109/WCCCT.2016.25.

[9] S. Sawant, J. Yu, K. Pandya, C. -K. Ngan and R. Bardeli, "An Enhanced BERTopic Framework and Algorithm for Improving Topic Coherence and Diversity," 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Hainan, China, 2022