INT2X: Explanation for the causes of prediction

Project Id: TMP-23-142

Project Proposal Report

Srinidee Methmal H.M.

B.Sc. (Hons) Degree in Information Technology

(Specializing in Software Engineering)

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2023

INT2X: Explanation for the causes of prediction

Project Id: TMP-23-142

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology

(Specializing in Software Engineering)

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2023

DECLARATION

We declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning, and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Srinidee Methmal H.M.	IT18161298	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date: 21/04/2023

ABSTRACT

With the advancement of machine learning techniques, the topic artificial intelligence becomes popular in recent years. With these advancements and automations, communities question the reliability of decision-making process which uses AI. Apart from that, major drawbacks of this decision-making processes are the lack of transparency and interpretability when it comes to real-world critical scenarios. The concept underlying this problem is called explainable AI (XAI). Nowadays, many countries come up rules and regulations to prohibit the use of black box models where the model interpretability is hidden (e.g.: European GDPR). Therefore, model interpretability (XAI) is the next step of AI. Even though the topic is popular, the number of studies done in this regard is less. The overview presented here is with the main objective of providing rule-based explanation method that can explain multivalued classification models as well as binary classification models with counterfactual explanations. The proposed method will specifically target the model logistic regression (LR).LR is a very popular classifier that performs well with linear data in a transformed space. However, logistic regression is not a fully black box model. When there is a non-linear relationship or greater data complexity between the features and the data, it turns into a "black box". Throughout the research, the drawbacks of the existing methods and weak points are identified and addressed using several techniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Description
AI	Artificial Intelligence
XAI	Explainable AI
LR	Linear Regression
GDPR	General Data Protection Regulation
VCS	Version Control System

LIST OF APPENDICES

1. INTRODUCTION

The explainable AI comes into the discussion with the evolution of artificial intelligence under machine learning domain. When AI is used in decision making process, the need for explainability has been emerged. Therefore, the interpretability of these systems is more important and must be proved. There are scenarios in which models have accountability, responsibility as well as transparency.

When the concept of AI becomes popular among large number of domains, they were encouraged to do decision making under the supervision of domain experts. These are some of the domains where AI for decision making is applied.

- Transportation
- Legal
- HealthCare
- Finance
- Military

In healthcare diagnosis, doctors and surgeons used AI models for predicting diseases. Since they are extremely sensitive models, they should produce proper explanations between inputs and outputs. Otherwise, decisions given by the model are not trustable. There can be situations where models can be biased on some features. In such situations, explainability is needed to identify and avoid bias in decision making.

Once machine learning models are used, accuracy is an important factor. Even though accuracy is high there should be a way to explain the path the decision has taken to produce such a decision. So there are two types of machine learning models called “White Box Models’’ and “Black Box Models”. Those are the two main taxonomies that can be used to divide AI models.

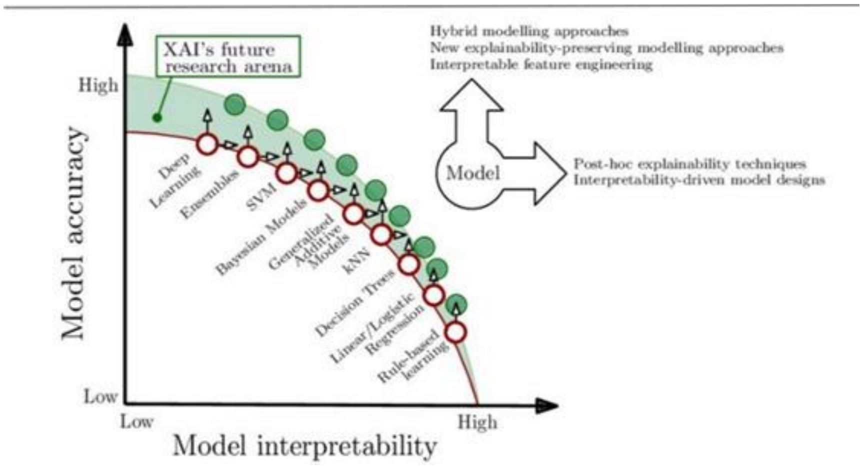
White box models are provided with a better explanation regarding the way it followed to provide a particular prediction and complexity is also less. In such situations, accuracy provided by the model cannot be enough. Examples of white box models are,

- Decision tree
- Naïve bayes
- Linear regression

In explanation point of view, they perform well.

Black box models are providing outstanding accuracies compared to white box models. But those models are less interpretable. Since General Data Protection Regulation (GDPR) required justifications to comply with global regulations, less interpretability is a drawback of black box models. Examples for black box models are,

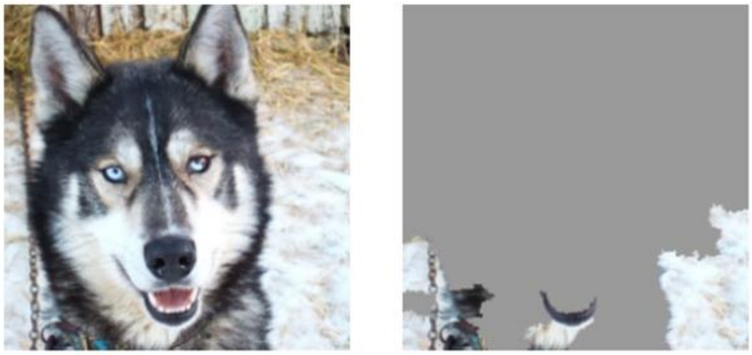
- Boosted trees
- Deep neural networks
- Random forests
- Support vector machine
- KNN



In [1] they propose four reasons why explainability is needed.

- Explain to justify
- Explain to control
- Explain to improve
- Explain to discover

There is a popular example to demonstrate the need of interpretability.This one is called “Husky” mistake [2].



(a) Husky classified as wolf

(b) Explanation

In the above situation, researchers trained a logistic regression neural network by feeding the images of huskies. The images of wolves had a white colour (snow) background and the images of huskies had not such a background. After the training process, they provide an image of a husky with snowy background. The model classified it as a wolf. In that situation what happened was the classifier predicts the image as Wolf if there is snow in the background. That means without looking at the animal colour, position, size, and other characteristics like that, the classifier predicts the given image based on the background snow in the image. So the model has trained as a bad classifier. With the help of XAI, these bad classifiers can be identified.

2. LITERATURE REVIEW

2.1 Background

The research field of XAI has taken the attention of the researchers and community with the development of AI models and related applications. The need and the importance of such a concept is raised with lawsuits published by the governments of some countries. As discuss in the introduction, the concept of XAI mainly focus on black box models. But Logistic Regression (LR) is a white box model. It becomes black box nature with the effect of the curse of dimensionality.

Among regression models, binary logistic regression is used to model the dichotomous dependent variable and multiple independent variables which are either continuous or categorical. There are some assumptions under it to give a valid result [1]. Among that,

- Explanatory variables should have a linear relationship with the logit of the response variable.
- Errors should not be correlated.
- Explanatory variables should not be highly correlated with each other (Multicollinearity).
- There should be no outliers, high leverage values or highly influential points.

Without satisfying above assumptions, the model may have problems like unnecessarily inflated standard errors, spuriously low or high t-statistics [3], parameter estimates with illogical signs and lead to invalid statistical inferences [4]. Belsley [5] noted that "... in nonexperimental sciences, ..., collinearity is a natural law in the data set resulting from the uncontrollable operations of the data-generating mechanism and is simply a painful and unavoidable fact of life." In many surveys, correlated variables are collected for analysis. Shen and Gao [6] suggested a double penalized maximum likelihood estimator combining Firth’s penalized likelihood equation to stabilize the estimates in cases of multicollinearity. Azar [7] proposed a method to estimate the shrinkage in parameters of Liu-type logistic estimator. Apart from that Schaefer, Roi & Wolfe [8] proposed a ridge type estimator that has smaller total mean squared error than the maximum likelihood estimator under certain conditions.

Multi-collinearity can be detected with the help of tolerance and its reciprocal, which is known as variance inflation factor [9] (VIF). The tolerance of any specific explanatory variable is 1-R2 where R2is the coefficient of determination n for the regression of that explanatory variable on all remaining independent variables. Tolerance close to 1 is little multicollinearity and close to 0 suggest that multicollinearity becomes a threat. So, a tolerance of 0.1 or less is a cause for concern. VIF means reciprocal of tolerance. It shows how much the variance of the coefficient estimate is being inflated by multicollinearity. The values of VIF exceeding 10 are multicollinearity. But for weaker models like logistic regression values above 2.5 are concern.

Apart from that eigen values for the scaled, uncentered cross-product matrix, condition indices and variance proportions for each explanatory variable are also used to identify multi-collinearity. If the eigen value is large, regression parameters are greatly affected by small changes in the explanatory variables or outcome. If the eigen values are similar, then the fitted model is unchanged by small changes in the measured variables [10]. The condition indices also compute as the square root of the ratio of the largest eigen value to the eigen value of interest. When there is no collinearity the eigen values and condition indices equal to unity. An informal rule of thumb is that if the condition index is 15, multicollinearity is a concern; if it is greater than 30, multicollinearity is a very serious concerned [11].

2.2 Literature Survey

To overcome less model interpretability, researchers propose sophisticated techniques to explain black box models. Since the research area is new, most of the techniques are in research state and they build communities around the development of the technique to come up with better solutions.

The authors of [1], introduce a couple of methods as explainability strategies. They are ”Scooped Related Methods” and “Model Related Methods”. Scooped related methods are globally interpretable (focus on the whole mechanism of the model) and locally interpretable (focus on explaining a specific decision or prediction).The model-related methods are also classified as Model-Specific Interpretability (methods are limited to a selected model) and Model-Agnostic Interpretability (methods are not limited to the selected model and it considers the prediction and explain separately).

Among proposed techniques some have taken the attention on researchers and new contributions have been done on top of these techniques [12]. Those are LIME (Local Interpretable Model-Agnostic Explanation) [13] and SHAP (Shapley Additive explanations) [14].

SHAP is a unified approach that has developed based on coalitional game theory. For each feature, SHAP assigns a feature importance value for a given prediction to approach the explainability of the model. When considering LIME, it uses local surrogate models to explain individual predictions or decisions.

Different feature ranking methods (SHAP, SAGE, FSP and BSP)

Though SHAP and SAGE scores are based on Shapley theory, SHAP has better performance in terms of model performance. But SAGE scores have the least correspondence in terms of model performance. SHAP does not assume feature independence and leverages feature clustering based on correlations which explains why it performs well better than SAGE. When comparing with other feature ranking methods, forward single pass (FSP) has a lower correspondence between total performance and model performance. In here forward permutation starts with all features permuted and it breaks the relationship between features. Neglecting the important features and isolating individual importance decreases faithfulness of FSP. Other than feature correlations, backward single-pass (BSP) method is the most faithful method. In this method impact of the correlated features on permutation importance score is heavily dependent on correlations between feature and target variable. If the two features are correlated, but there's a stronger predictor of the target variable permutation importance scores may be negligibly impacted by the feature correlations. To determine the dataset has sufficient correlations with the target variable compared to correlations between features, we compare average feature correlation and the average correlation of a feature with the target variable.

Feature Relevance [15]

It is beneficial to figure out most impactful features which are crucial in decision making. For that purpose, feature importance is introduced. It shows the impact factor of each feature for decisions [16]. Along with feature importance correlation among features is also important. In AI based medical diagnosis, feature correlation in training data is one of the main forces for diagnosis.

Saliency map-based technique [15]

There is a strong correlation between feature score-based justifications and saliency-based Visual representations. More research-based demonstrations also choose it. Saliency-based visualizations are popular because they give visually perceptive explanations that can be easily understood by the end-users [17].

Explanations by example method [18]

This method considers extraction of data samples that relate to the result generated by a certain model. Similar, to how human behaves when explaining a given process, explanations by example are mainly focused on extracting representative samples that show inner relationships and correlations which are found by analysing the model.

Correlation coefficient method [19]

This is a kind of feature selection method that effectively choose the most appropriate feature and reduce data dimensions. There are several supervised correlation coefficient methods. But few can detect both linear correlation and non-linear correlations. In here we further discussed supervised correlation coefficient method called consistency detection. The results obtained from above method can better retain the dimensions of classifications and detect correlation than Pearson correlation coefficient method.

Dimension reduction methods can be divided into two categories feature extraction and feature selection [20]. When considering the generality, the correlation coefficient methods can be divided into two categories called supervised correlation coefficient method and unsupervised correlation coefficient method.

Liliana Forzani et.al. [21] had considered important information on the relation between independent variable and dependent variable may be lost when using principal component analysis. Wang et al. [22] had considered a feature selection method which can detect statistical significance between discrete dimensions and continuous index. Duan et al. [23] considered the dimensionality reduction methods which do feature selection before feature extraction. Korn F et al. [24] introduces dimensionality reduction method for real data sets with non-uniform distributions and incompletely independent dimensions.

Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations [25]

The concept of counterfactual needs imagination of hypothetical realities that happen other than the existing situation. In this research to understand the concept of counterfactuals it has given a real-time scenario. It will be discussed below.

In a situation of rejecting a loan, using interpretable models the organization may give explanation for that action. As an example, it may be due to “poor credit” history. When we consider in person point of view who has been applied for the loan, the explanation does not help him to decide “what should he/she do next?” If the system can suggest some solutions to improve the chance of getting the loan in future, that would be more effective for both sides.

However, there must be situations where the most important feature or features like gender, race may not be sufficient to flip or change the correlation of prediction features. Therefore, it is pretty much important to provide alternative rules which are actionable.

As the counterfactual explanation for above loan example, the paper has provided below information [25].

“You would have received the load if your income was higher by $10000”.

When the person gets that kind of explanation, he/she would be able to identify which features need to be improved to be eligible for the loan in future.

3. RESEARCH GAP

Already proposed local rule-based explanation methods are mainly focusing on object-based and grid-based datasets. The [18] literature dealing with explainability, has approached two different perspectives. Namely, [1] ML models that feature some degree of transparency, thereby interpretable to some extent [2] post-hoc XAI techniques devised to make ML models interpretable. The existing methods have highlighted several post-hoc explainability methods SHAP, LIME, permutation importance, partial dependence/ALE) that developed to improve ML models like Logistic Regression. In this research, we evaluate multiple feature ranking methods and how feature ranking faithfulness is impacted by dimensionality reduction. Further it illustrates the outcome of multi-collinearity such as unstable estimates, inaccurate variances, biasness and how model explainability increased by limiting correlated features using different machine learning model agnostic frameworks like SHAP, LIME and ELI5.Moreover, it provides how model explainability improves through dimensionality reduction and knowing relative faith-fullness of feature ranking methods for a text data set using Logistic regression as the ML model.

	Existing methods	**The method proposed by the study**
Text classification	***X***	**√**
Post-hoc explainability(SHAP, LIME, permutation importance, ALE variance and Logistic Regression (LR) coefficients)	**√**	**√**
Provide counterfactual rule generation-based explanation method	**√**	**√**
Provide user-friendly visualizations	**√**	**√**

Table 3.1: Comparison between existing methods and the proposed method

4. RESEARCH PROBLEM

When it comes to the applications of machine learning models, those models are used in decision classifier systems. Therefore, the importance of XAI is highlighted in those situations. Apart from that there are multiple approaches that have taken by researchers to provide proper explanations to explain these ML models. When it comes to existing methods post hoc explainability methods like SHAP, LIME, partial dependence/ALE and permutation importance have been developed to verify the faithfulness of the explainability methods. By giving the sensitivity of many of those methods to correlated features, we evaluate how feature ranking faithfulness is impacted by dimensionality reduction. Moreover, existing studies only discuss how model explainability can be improved through dimensionality reduction which is specifically done for object-based and grid-based datasets. In this study we further discuss improving the interpretability of Logistic regression (when it shows black box nature in curse of dimensionality) using text classification.

As discussed in [25], [26] counterfactual explanations are important in terms of model interpretability. To get user-friendly explanations, counterfactual explanations are performing a significant role. It provides the ability for decision making systems to provide suggestions to flip the prediction. According to [26], the existing methods focus only binary predictors only. Apart from that when decision making systems need counterfactual explanations, to text classifies, the existing methods are not sufficient.

Those existing post hoc explainability methods and counterfactual techniques are not properly applied to Logistic regression model when it shows the black box nature. So, to apply those techniques with LR model, there should be an evaluation done regarding the method compatibility with the model.

5. OBJECTIVES

5.1 Main Objective

Enhance the interpretability of Logistic Regression, especially when it shows black box nature under the curse of dimensionality by reducing multiple correlation between independent variables using text classifiers.

The current work has been done using post hoc explainability methods like SHAP, LIME, partial dependence/ALE and permutation importance. Other than that, feature ranking methods like forward single-pass (FSP), backward single-pass (BSP) and backward multiple-pass (BMP) is applied to correlated features and from there it evaluates how feature ranking faithfulness is impacted by dimensionality reduction. At the end of the research, more efficient method will be selected and applied for generating neighbourhood from training data.

The identified neighbourhood will be passed to logistic regression classifier to elicit decision rules and counterfactual rules. Through this approach, it is simple to understand as it follows the same process which a human follows while making any decision in real-life.

Other than that proposed solution will be implemented as a web application that provides better user experience to end users. Since existing methods are difficult to understand by the users, understandable user-friendly visualizations will be provided by the research.

5.2 Specific Objectives

To achieve the main objective, there are few milestones to reach.

- **Identify appropriate text dataset and apply data pre-processing techniques:**

  To perform experiment, we should identify suitable text data set that align with the requirements. It can be taken from open-source resources like Kaggle.

- **Generating neighbourhood from training data:**

By considering the instance to be predicted a neighbourhood is generated using training data.

- **Extracting the explanation rule from the logistic regression classifier:**
  The rule which explains the prediction needs to be extracted from the implemented logistic regression classifier.
- **Extract counterfactual explanations from logistic regression classifier:**
  After extracting the explanation rule, the relevant counterfactual explanation rules should be identified using logistic regression classifier.

5.3 Work Breakdown Structure

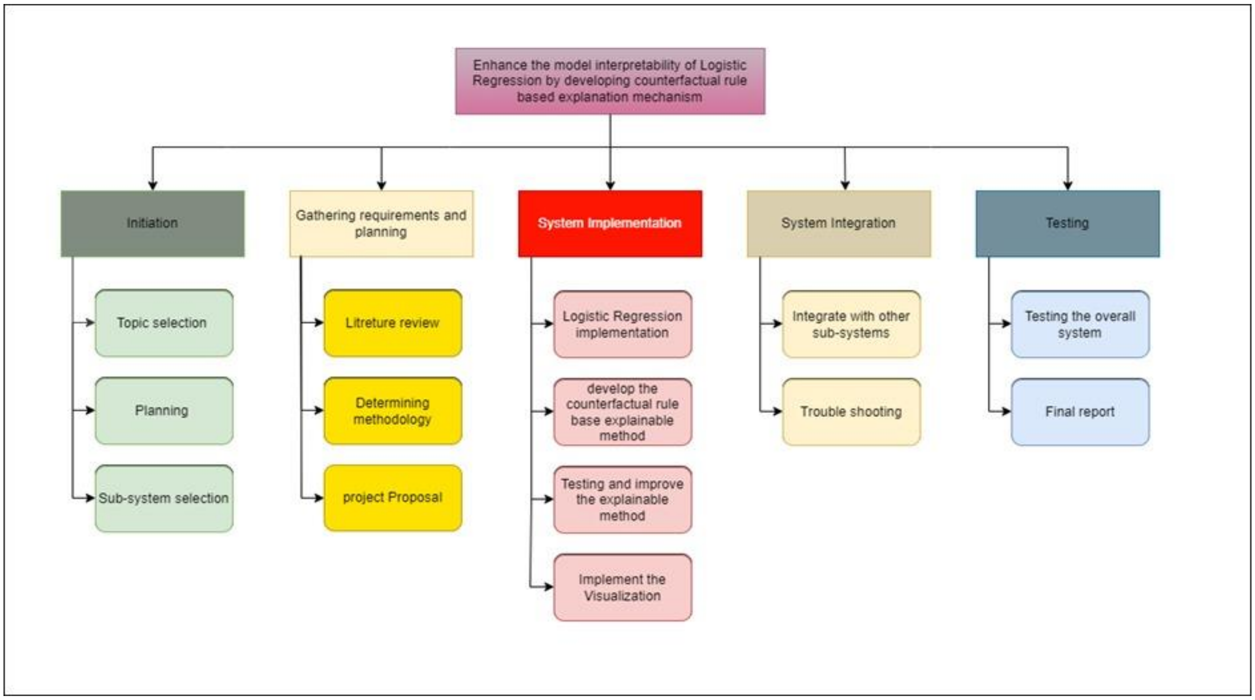

Enhance the model interpretability of Logistic Regression by developing counterfactual rule based explanation mechanism
Initiation
Topic selection
Planning
Sub-system selection
Gathering requirements and planning
Litreture review
Determining methodology
project Proposal
System Implementation
Logistic Regression implementation
develop the counterfactual rule base explainable method
Testing and improve the explainable method
Implement the Visualization
System Integration
Integrate with other sub-systems
Trouble shooting
Testing
Testing the overall system
Final report

Figure 5 1: Work Breakdown Structure

6. METHODOLOGY

To implement the proposed solution, there are few milestones to accomplish. In here it discussed about environments, techniques, requirements, tools and technologies are required. The steps that need to be followed to carry out the study will be discussed here.

6.1 System Architecture Diagram

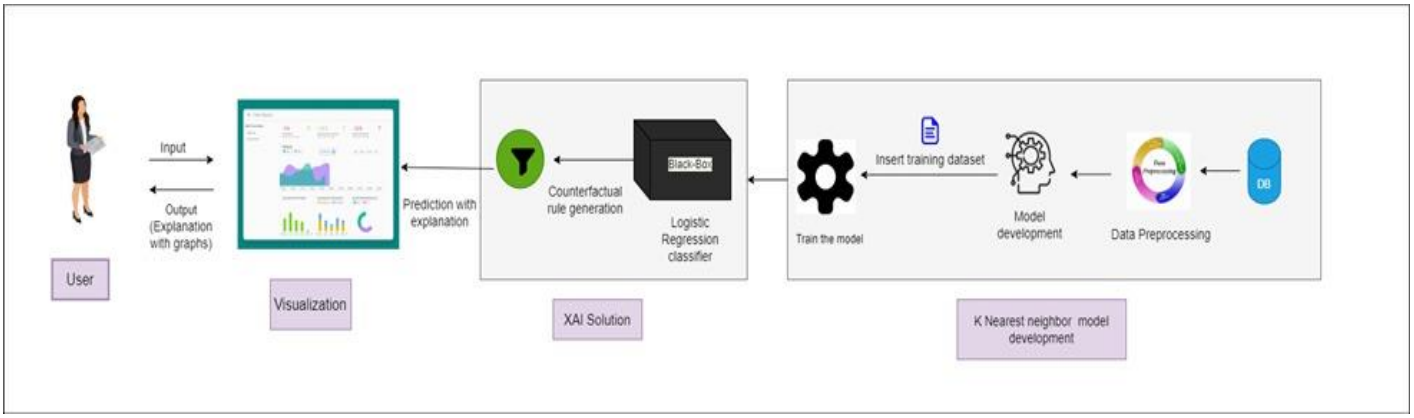


Figure 6 1: System Architecture Diagram

Above diagram illustrates the high-level system architecture of the proposed solution. As the first step, the user must provide the training text data set and the instance need to be predicted to the system through GUI. The provided data will be applied to Logistic Regression (LR) model to get the prediction. To extract the explanation rules, a neighbourhood from the given text data set is applied to the logistic regression classifier to get explanation rules. Meanwhile, counterfactual rules will be generated. The outcomes of the process (Explanation rules, counterfactual rules) will be transferred to the GUI with user-friendly visualizations to be more understandable to users.

6.2 Tools and Technologies

- **Front-end:**

  HTML, CSS, and JavaScript will be used as front-end technologies. Ajax Delta Communication technology will be used to handle the communication between front-end and back-end.

- **Back-end:**

  Python will be used as the programming language for back-end development.

- **Version Control:**

  Gitlab will be used as the VCS Platform.

7. PROJECT REQUIREMENTS

User Requirements

- Users should have a knowledge of decision-making systems based on machine learning.
- Dataset should be pre-processed, and appropriate data engineering techniques should be applied.

Functional Requirements

- System should be able to provide appropriate visualizations when needed.
- Quantify how model explainability improved through dimensionality reduction and knowing the faithfulness of feature ranking methods for a given text data input.
- Model accuracies should be provided by the system.

Non-Functional Requirements

- Output should be provided efficiently.
- Explanation rule generation steps should be precise.

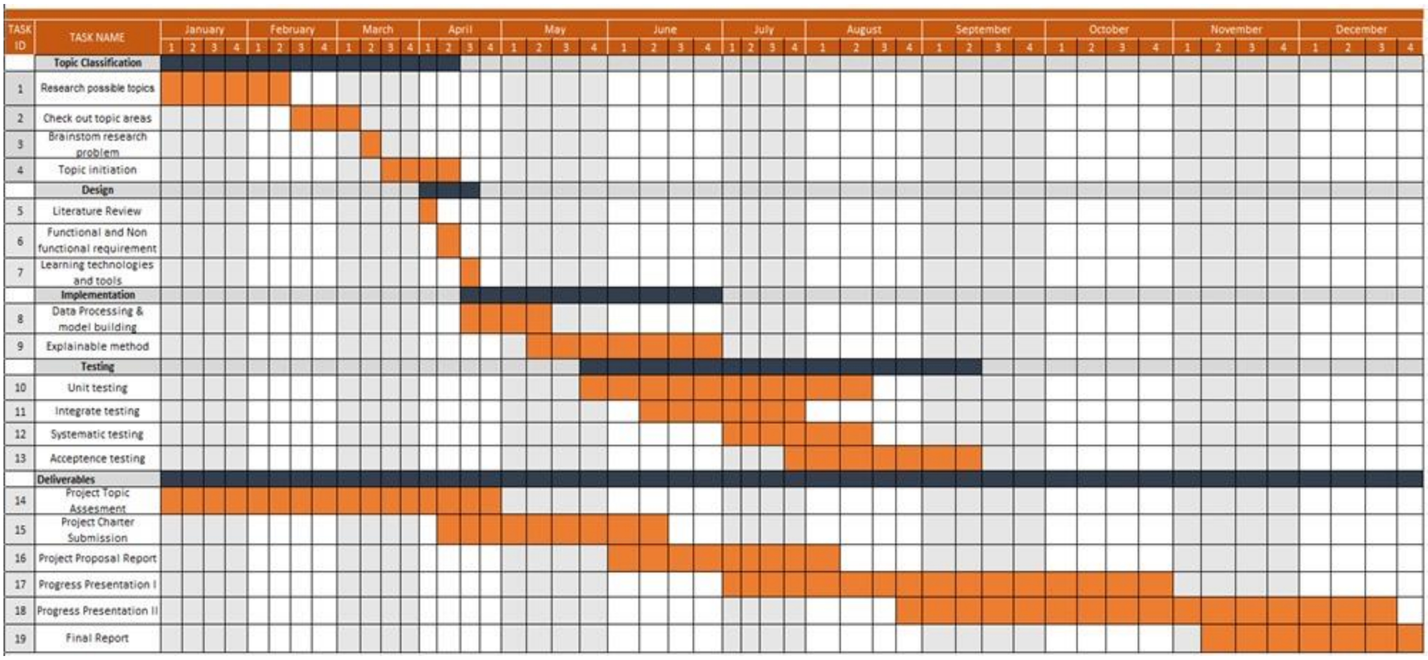
8. Budget and justification

For this research component, we do not aim to commercialize our individual components. Following is the budget justification for the whole research system.

Item	Cost (Rs)
Solution publishing cost	5000.00
Backend hosting cost	10000.00
Front end hosting cost	5000.00
Research paper publishing cost	5000.00
Total	25000.00

Table 8.1 - Budget and budget justification

8. GANTT CHART



TASK ID	TASK NAME	January				February				March				April				May				June				July				August				September				October				November				December			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	Topic Classification	■	■	■	■	■	■	■	■	■	■	■	■	■	■																																		
1	Research possible topics	■	■	■	■	■	■																																										
2	Check out topic areas							■	■	■																																							
3	Brainstom research problem										■																																						
4	Topic initiation											■	■	■	■																																		
	Design													■	■	■																																	
5	Literature Review													■																																			
6	Functional and Non functional requirement														■																																		
7	Learning technologies and tools															■																																	
	Implementation															■	■	■	■	■	■	■	■	■	■																						
8	Data Processing & model building															■	■	■	■																														
9	Explainable method																		■	■	■	■	■	■	■																						
	Testing																				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■														
10	Unit testing																				■	■	■	■	■	■	■	■	■	■	■																		
11	Integrate testing																						■	■	■	■	■	■	■																				
12	Systematic testing																									■	■	■	■	■	■																		
13	Acceptence testing																												■	■	■	■	■	■	■														
	Deliverables	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
14	Project Topic Assesment	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																																
15	Project Charter Submission														■	■	■	■	■	■	■	■	■																										
16	Project Proposal Report																					■	■	■	■	■	■	■	■	■																			
17	Progress Presentation I																									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
18	Progress Presentation II																																■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
19	Final Report																																										■	■	■	■	■	■	■

REFERENCE LIST

[1] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2870052.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should i trust you?’ Explaining the predictions of any classifier,” in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, vol. 13-17-Augu, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[3] A. Field, Discovering statistics using IBM SPSS statistics., sage, 2013 Feb 20.

[4] D. Liao and R. Valliant, "Condition indexes and variance decompositions for diagnosing collinearity in linear model analysis of survey data," Survey Methodology 38, no. 2, pp. 189-202, 2012.

[5] D. Belsley, "A guide to using the collinearity diagnostics," Computer Science in Economics and Management 4, no. 1, pp. 33-50, 1991.

[6] J. Shen and S. Gao, "A solution to separation and multicollinearity in multiple logistic regression," Journal of data science: JDS 6, no. 4, p. 515, 2008 Oct 10.

[7] Y. Asar, "Some new methods to solve multicollinearity in logistic regression," Communications in Statistics-Simulation and Computation 46, no. 4, pp. 2576-2586, 2017.

[8] R. Schaefer, L. Roi and R. Wolfe, "A ridge logistic estimator," Communications in Statistics-Theory and Methods 13, no. 1, pp. 99-113, 1984 Jan 1.

[9] N. Senaviratna and T. Cooray, "Diagnosing multicollinearity of logistic regression model," Asian Journal of Probability and Statistics 5, no. 2, pp. 1-9, 2019 Oct 1.

[10] S. Rana, H. Midi and S. Sarkar, "Validation and performance analysis of binary logistic regression model.," in WSEAS Press, 2010.

[11] H. Midi, S. Sarkar and S. Rana, "Collinearity diagnostics of binary logistic regression model," Journal of interdisciplinary mathematics 13, no. 3, pp. 253-267, 2010.

[12] Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2019. https://christophm.github.io/interpretable-ml-book/

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should i trust you?’ Explaining the predictions of any classifier,” in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016, vol. 13-17-Augu, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[14] S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Section 2, pp. 4766–4775, 2017.

[15] P. Gohel, P. Singh and M. Mohanty, "Explainable AI: current status and future directions," arXiv preprint arXiv:2107.07045, 2021 Jul 12.

[16] N. Rajani, B. McCann, C. Xiong and R. Socher, "Explain yourself! leveraging language models for commonsense reasoning," arXiv preprint arXiv:1906.02361, 2019 Jun 6.

[17] N. Poerner, B. Roth and H. Schütze, "Evaluating neural network explanation methods using hybrid documents and morphological agreement," arXiv preprint arXiv:1801.06422, 2018 Jan 19.

[18] A. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina,, R. Benjamins and R. Chatila, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information fusion 58, pp. 82-115, 2020.

[19] S. Wang and L. Zhang, "A Supervised Correlation Coefficient Method: Detection of Different Correlation," 2020 12th International Conference on Advanced Computational Intelligence (ICACI), Dali, China, 2020, pp. 408-411, doi: 10.1109/ICACI49185.2020.9177709.

[20] X. Yao, X. Wang, Y. Zhang et al., "Overview of feature selection methods", Control and Decision, vol. 27, pp. 161-166, 2012.

[21] Liliana Forzani and Daniela Rodriguez, "Sufficient dimension reduction and prediction in regression: Asymptotic results", Journal of Multivariate Analysis, vol. 171, pp. 339-349, 2019.

[22] J. Wang and C. Xu, "Geodetector: Principle and prospective", Acta Geographica Sinica, vol. 72, pp. 116-134, 2017.

[23] Y. Duan and Q. Wang, "Optimization of rapid detection model of egg freshness spectrum based on feature selection and feature extraction", Food Science, pp. 1-9.

[24] B U Pagel, F Korn and C Faloutsos, "Deflating the dimensionality curse using multiple fractal dimensions", in Proceeding International Conference on Data Engineering, vol. 1, pp. 589-598, 2000.

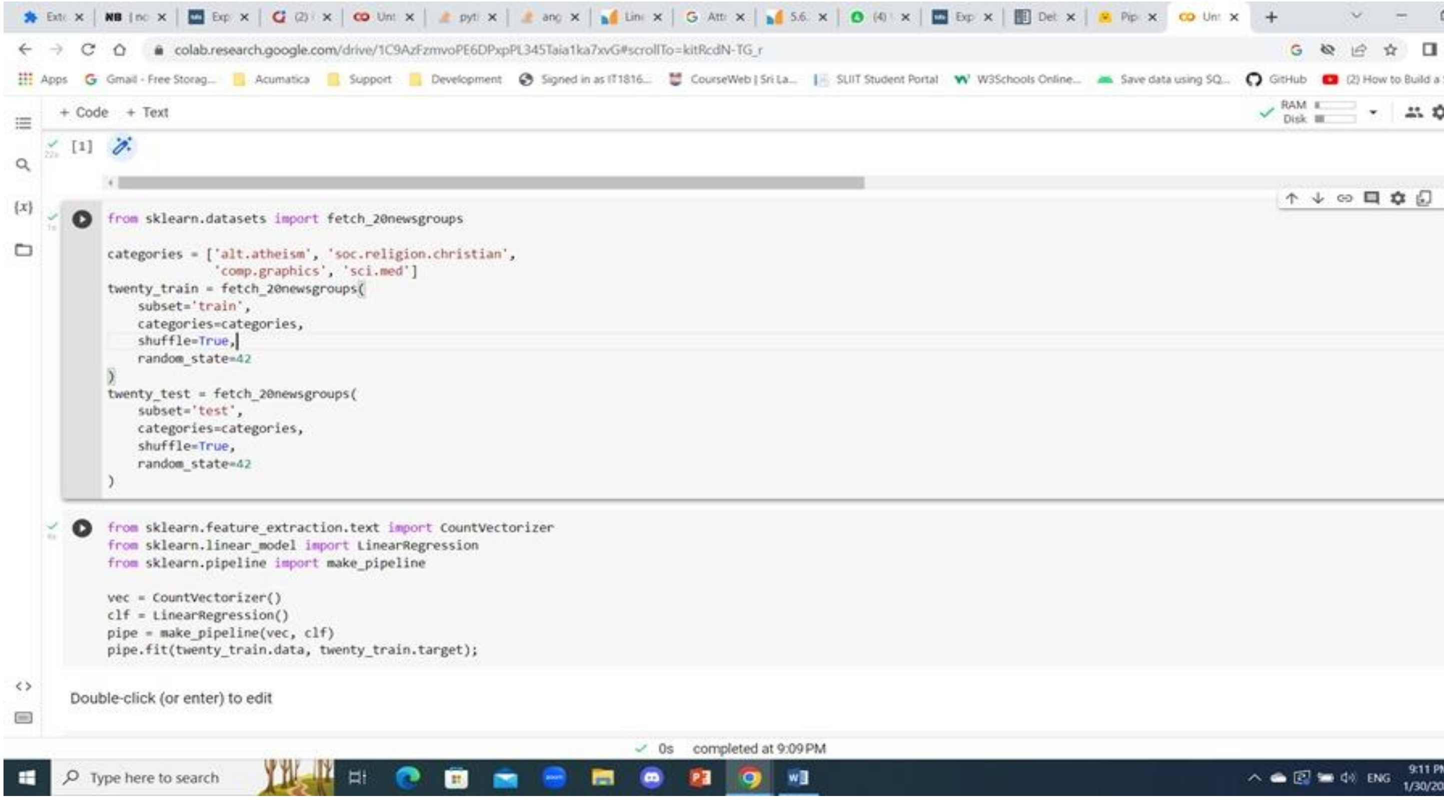
[25] R. K. Mothilal and C. Tan, “Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations.”, 2019

[26] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, “Local rule-based explanations of black box decision systems,” arXiv, no. May, 2018.

APPENDICES

A. Eli5 explanation for linear regression

Below diagrams illustrate Sample code output (Here we select 20 newsgroups dataset as our text classification dataset). By using the machine learning framework ELI5 we generate visualization sample outputs to interpret the predictions of the regression model.

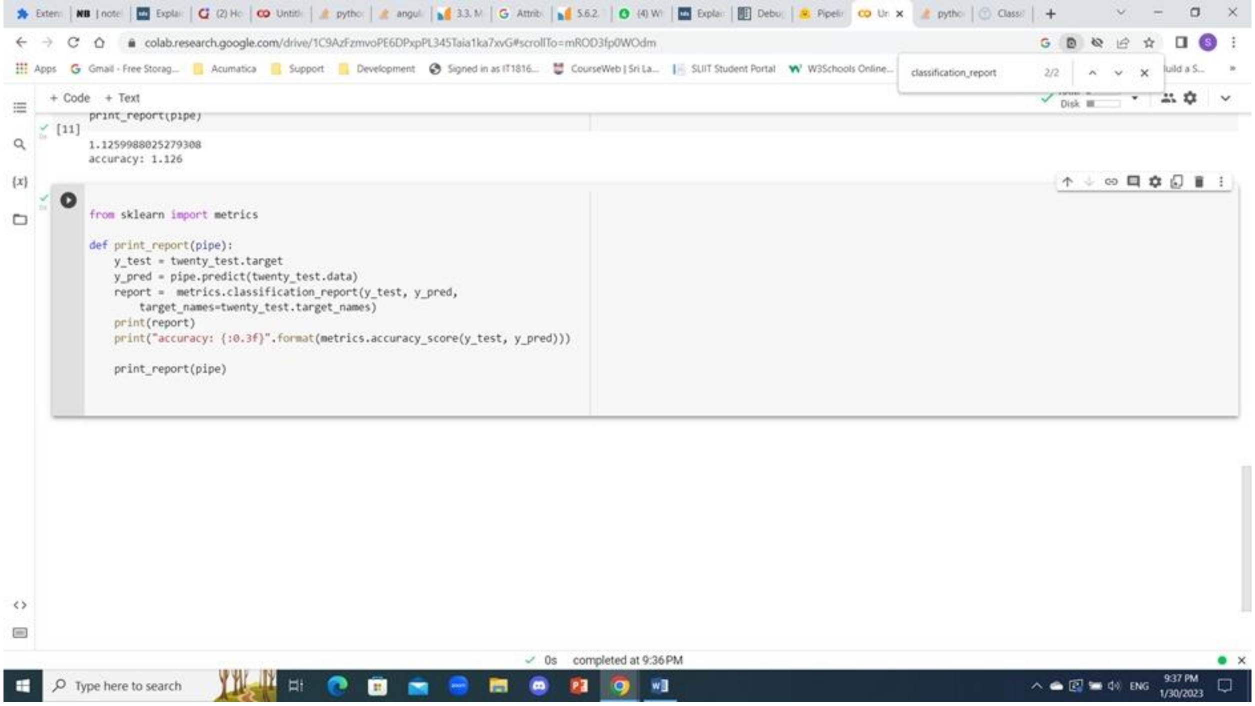


```
from sklearn.datasets import fetch_20newsgroups

categories = ['alt.atheism', 'soc.religion.christian',
              'comp.graphics', 'sci.med']
twenty_train = fetch_20newsgroups(
    subset='train',
    categories=categories,
    shuffle=True,
    random_state=42
)
twenty_test = fetch_20newsgroups(
    subset='test',
    categories=categories,
    shuffle=True,
    random_state=42
)
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline

vec = CountVectorizer()
clf = LinearRegression()
pipe = make_pipeline(vec, clf)
pipe.fit(twenty_train.data, twenty_train.target);
```

Double-click (or enter) to edit

Figure 1.1

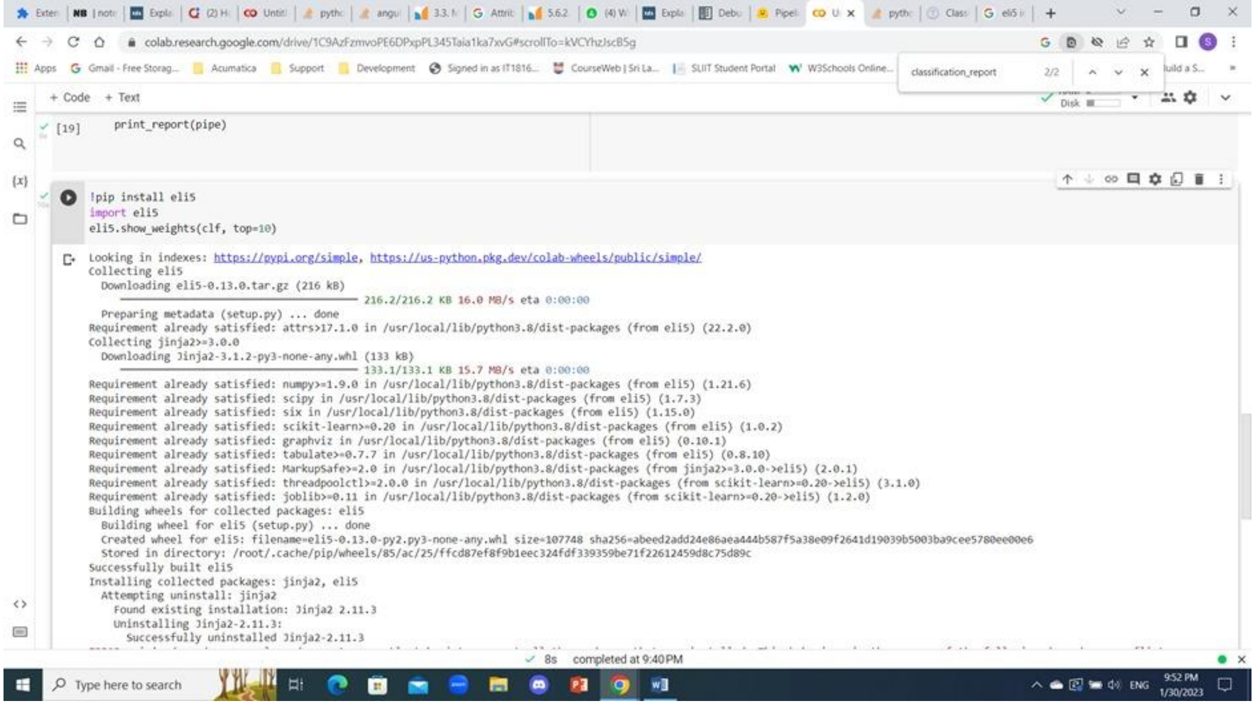


colab.research.google.com/drive/1C9AzFzmvoPE6DPxpPL345Taia1ka7xvG#scrollTo=mROD3fp0WOdm
classification_report 2/2
+ Code + Text
completed at 9:36 PM

```
print_report(pipe)
[11]
1.1259988025279308
accuracy: 1.126
```

```
from sklearn import metrics

def print_report(pipe):
    y_test = twenty_test.target
    y_pred = pipe.predict(twenty_test.data)
    report =  metrics.classification_report(y_test, y_pred,
        target_names=twenty_test.target_names)
    print(report)
    print("accuracy: {:0.3f}".format(metrics.accuracy_score(y_test, y_pred)))

    print_report(pipe)
```

Figure 1.2



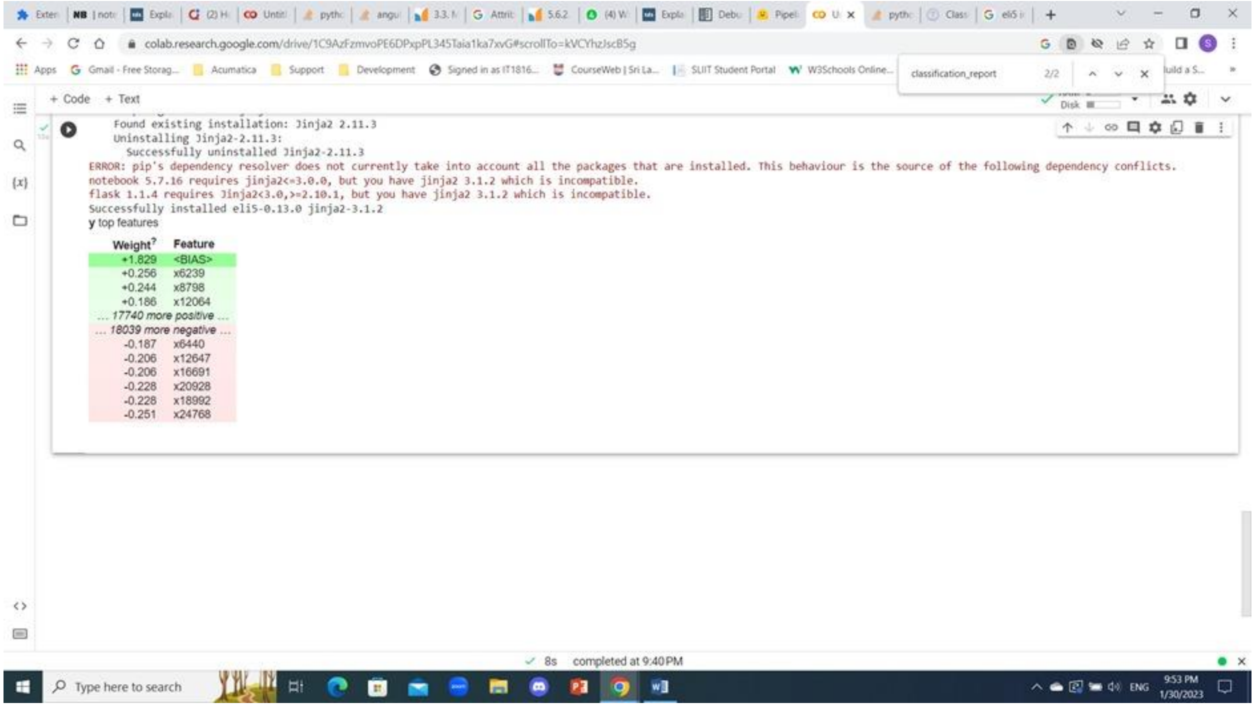
colab.research.google.com/drive/1C9AzFzmvoPE6DPxpPL345Taia1ka7xvG#scrollTo=kVCYhzJscB5g
classification_report 2/2
+ Code + Text
8s completed at 9:40 PM

```
[19]    print_report(pipe)
```

```
!pip install eli5
import eli5
eli5.show_weights(clf, top=10)
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting eli5
  Downloading eli5-0.13.0.tar.gz (216 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 216.2/216.2 KB 16.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: attrs>17.1.0 in /usr/local/lib/python3.8/dist-packages (from eli5) (22.2.0)
Collecting jinja2>=3.0.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.1/133.1 KB 15.7 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.8/dist-packages (from eli5) (1.21.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from eli5) (1.7.3)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from eli5) (1.15.0)
Requirement already satisfied: scikit-learn>=0.20 in /usr/local/lib/python3.8/dist-packages (from eli5) (1.0.2)
Requirement already satisfied: graphviz in /usr/local/lib/python3.8/dist-packages (from eli5) (0.10.1)
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.8/dist-packages (from eli5) (0.8.10)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.8/dist-packages (from jinja2>=3.0.0->eli5) (2.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn>=0.20->eli5) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn>=0.20->eli5) (1.2.0)
Building wheels for collected packages: eli5
  Building wheel for eli5 (setup.py) ... done
  Created wheel for eli5: filename=eli5-0.13.0-py2.py3-none-any.whl size=107748 sha256=abeed2add24e86aea444b587f5a38e09f2641d19039b5003ba9cee5780ee00e6
  Stored in directory: /root/.cache/pip/wheels/85/ac/25/ffcd87ef8f9b1eec324fdf339359be71f22612459d8c75d89c
Successfully built eli5
Installing collected packages: jinja2, eli5
  Attempting uninstall: jinja2
    Found existing installation: Jinja2 2.11.3
    Uninstalling Jinja2-2.11.3:
      Successfully uninstalled Jinja2-2.11.3
```

Figure 1.3



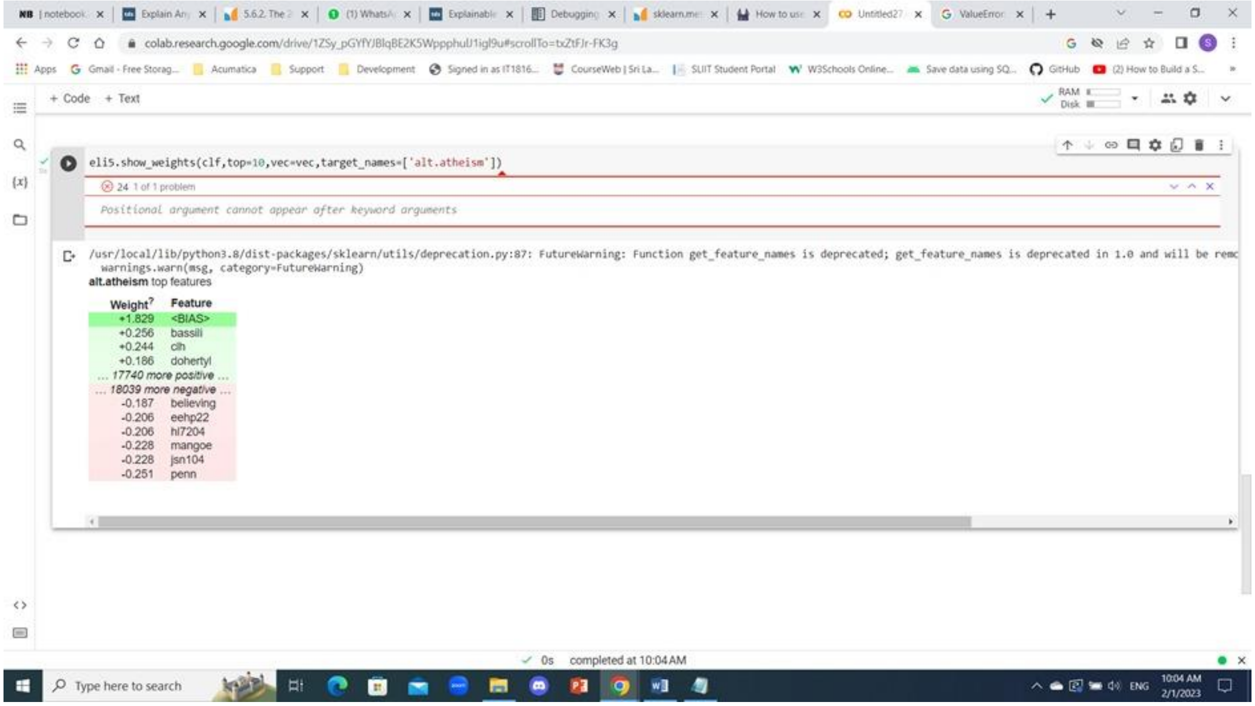
```
    Found existing installation: Jinja2 2.11.3
    Uninstalling Jinja2-2.11.3:
      Successfully uninstalled Jinja2-2.11.3
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
notebook 5.7.16 requires jinja2<=3.0.0, but you have jinja2 3.1.2 which is incompatible.
flask 1.1.4 requires Jinja2<3.0,>=2.10.1, but you have jinja2 3.1.2 which is incompatible.
Successfully installed eli5-0.13.0 jinja2-3.1.2
```

y top features

Weight?	Feature
+1.829	<BIAS>
+0.256	x6239
+0.244	x8798
+0.186	x12064
... 17740 more positive ...	
... 18039 more negative ...	
-0.187	x6440
-0.206	x12647
-0.206	x16691
-0.228	x20928
-0.228	x18992
-0.251	x24768

colab.research.google.com/drive/1C9AzFzmvoPE6DPxpPL345Taia1ka7xvG#scrollTo=kVCYhzJscB5g
classification_report 2/2
+ Code + Text
Disk
✓ 8s completed at 9:40 PM
Type here to search
9:53 PM
1/30/2023

Figure 1.4



```
eli5.show_weights(clf,top=10,vec=vec,target_names=['alt.atheism'])
```

24 1 of 1 problem

Positional argument cannot appear after keyword arguments

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be remo
  warnings.warn(msg, category=FutureWarning)
```

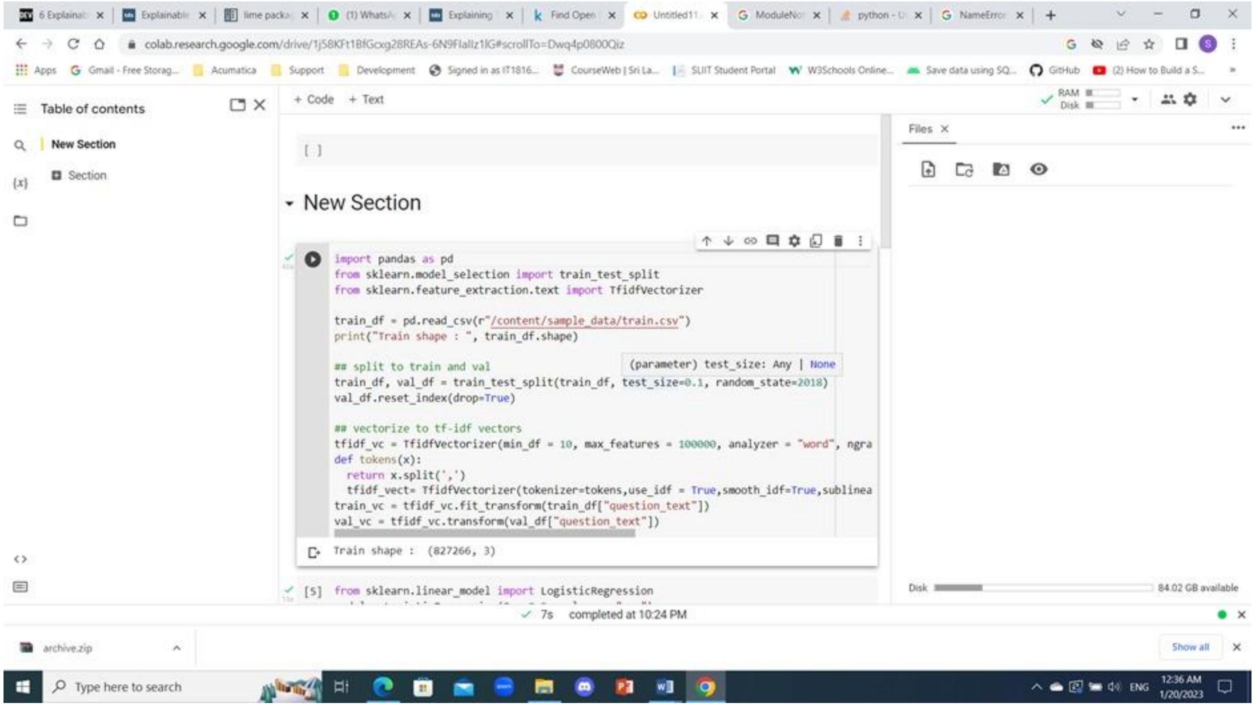
alt.atheism top features

Weight?	Feature
+1.829	<BIAS>
+0.256	bassili
+0.244	clh
+0.186	dohertyl
... 17740 more positive ...	
... 18039 more negative ...	
-0.187	believing
-0.206	eehp22
-0.206	hl7204
-0.228	mangoe
-0.228	jsn104
-0.251	penn

colab.research.google.com/drive/1ZSy_pGYfYJBlqBE2K5WppphulJ1igl9u#scrollTo=txZtFJr-FK3g
+ Code + Text
RAM
Disk
✓ 0s completed at 10:04 AM
Type here to search
10:04 AM
2/1/2023

Figure 1.5

B. LIME Explanation for Logistic Regression

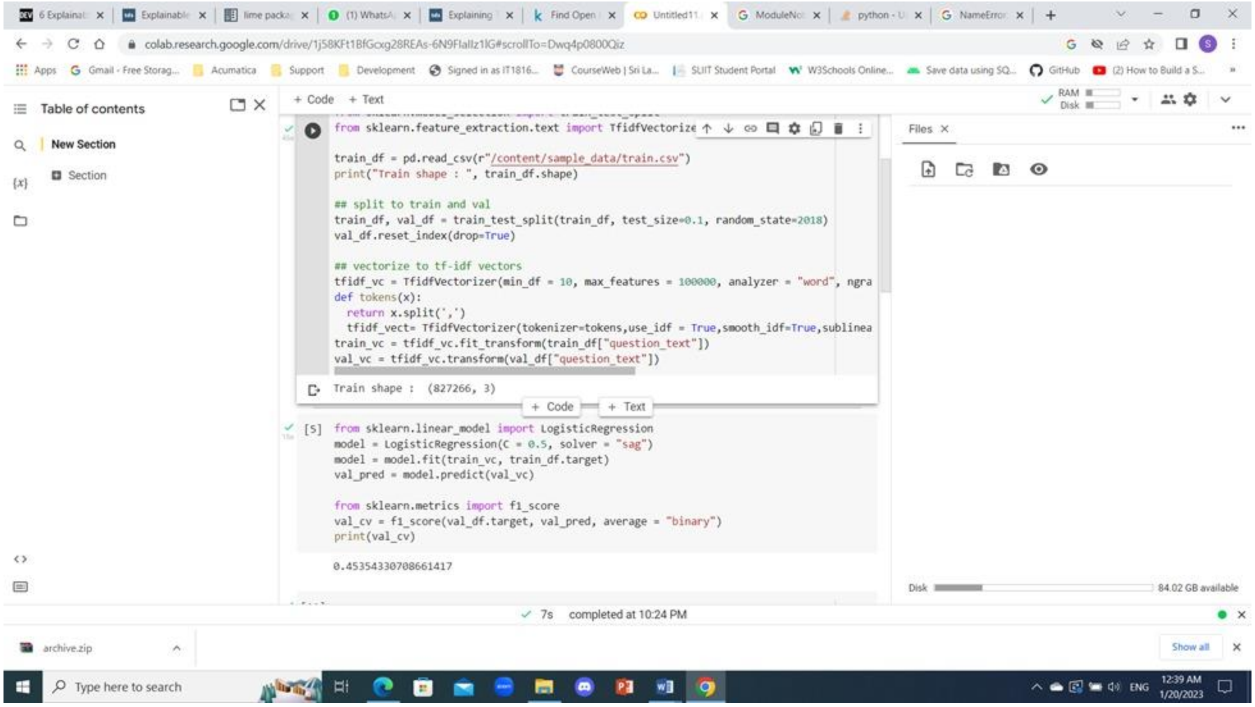


```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

train_df = pd.read_csv(r"/content/sample_data/train.csv")
print("Train shape : ", train_df.shape)

## split to train and val
train_df, val_df = train_test_split(train_df, test_size=0.1, random_state=2018)
val_df.reset_index(drop=True)

## vectorize to tf-idf vectors
tfidf_vc = TfidfVectorizer(min_df = 10, max_features = 100000, analyzer = "word", ngra
def tokens(x):
  return x.split(',')
  tfidf_vect= TfidfVectorizer(tokenizer=tokens,use_idf = True,smooth_idf=True,sublinea
train_vc = tfidf_vc.fit_transform(train_df["question_text"])
val_vc = tfidf_vc.transform(val_df["question_text"])
```

```
Train shape :  (827266, 3)
```

```
[5] from sklearn.linear_model import LogisticRegression
```

Figure 2.1



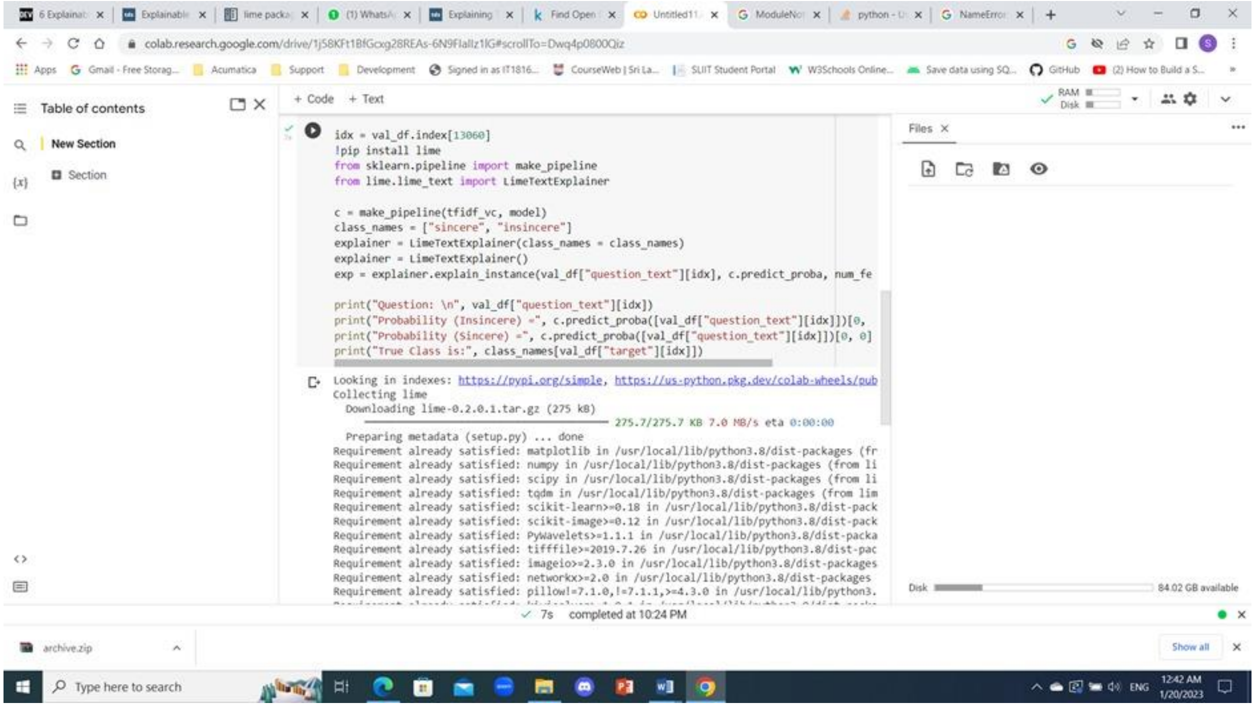
```
from sklearn.feature_extraction.text import TfidfVectorize

train_df = pd.read_csv(r"/content/sample_data/train.csv")
print("Train shape : ", train_df.shape)

## split to train and val
train_df, val_df = train_test_split(train_df, test_size=0.1, random_state=2018)
val_df.reset_index(drop=True)

## vectorize to tf-idf vectors
tfidf_vc = TfidfVectorizer(min_df = 10, max_features = 100000, analyzer = "word", ngra
def tokens(x):
  return x.split(',')
  tfidf_vect= TfidfVectorizer(tokenizer=tokens,use_idf = True,smooth_idf=True,sublinea
train_vc = tfidf_vc.fit_transform(train_df["question_text"])
val_vc = tfidf_vc.transform(val_df["question_text"])
```

```
Train shape :  (827266, 3)
```

```
[5] from sklearn.linear_model import LogisticRegression
    model = LogisticRegression(C = 0.5, solver = "sag")
    model = model.fit(train_vc, train_df.target)
    val_pred = model.predict(val_vc)

    from sklearn.metrics import f1_score
    val_cv = f1_score(val_df.target, val_pred, average = "binary")
    print(val_cv)
```

```
0.45354330708661417
```

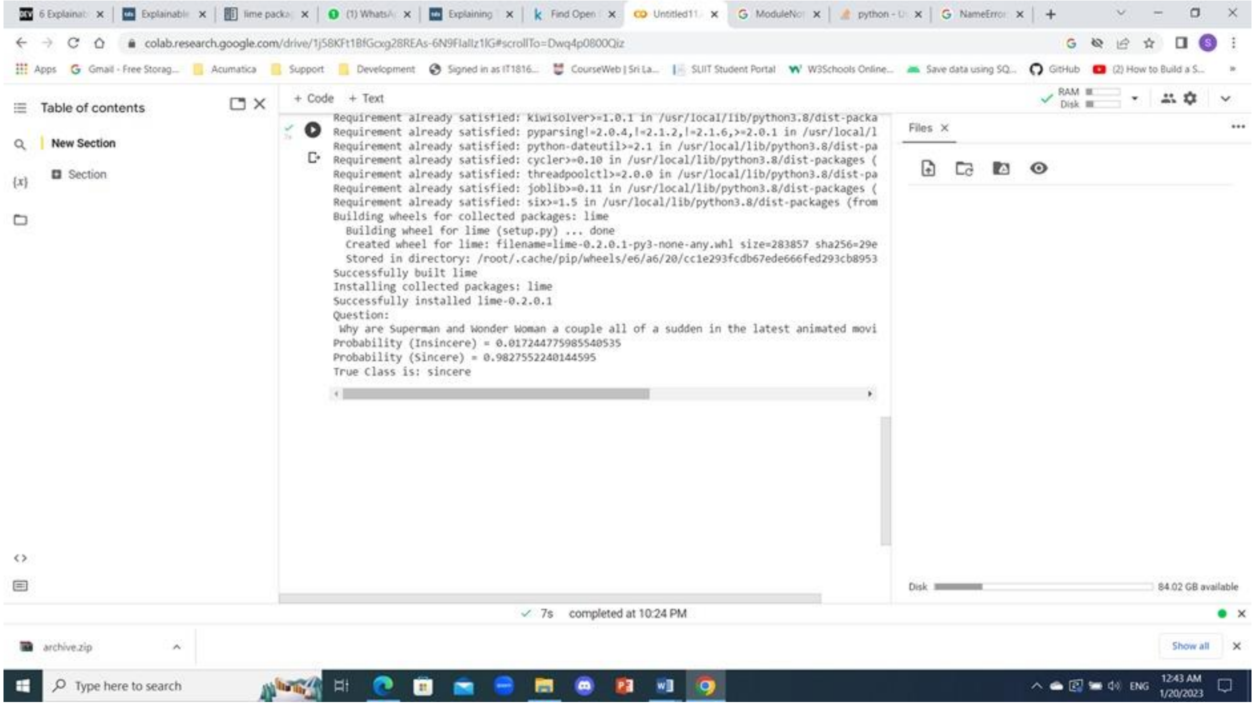
Figure 2.2



```
idx = val_df.index[13060]
!pip install lime
from sklearn.pipeline import make_pipeline
from lime.lime_text import LimeTextExplainer

c = make_pipeline(tfidf_vc, model)
class_names = ["sincere", "insincere"]
explainer = LimeTextExplainer(class_names = class_names)
explainer = LimeTextExplainer()
exp = explainer.explain_instance(val_df["question_text"][idx], c.predict_proba, num_fe

print("Question: \n", val_df["question_text"][idx])
print("Probability (Insincere) =", c.predict_proba([val_df["question_text"][idx]])[0,
print("Probability (Sincere) =", c.predict_proba([val_df["question_text"][idx]])[0, 0]
print("True Class is:", class_names[val_df["target"][idx]])
```

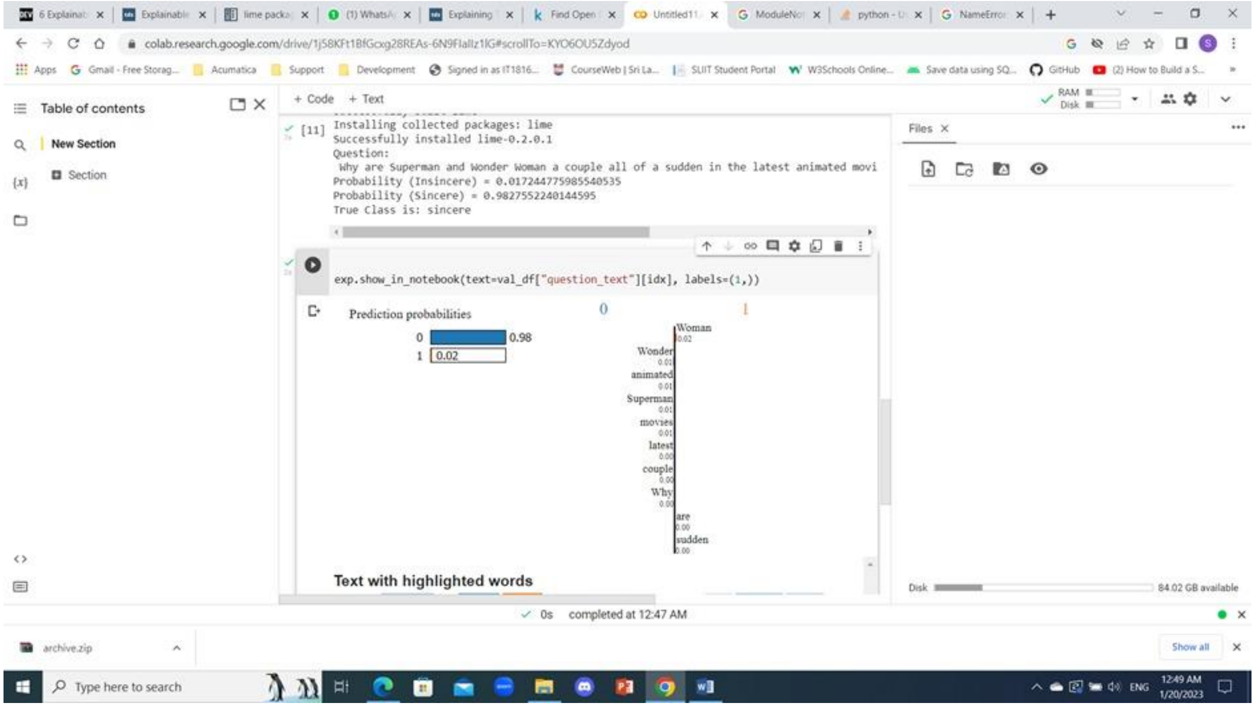
```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 275.7/275.7 KB 7.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-packages (fr
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from li
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from li
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages (from lim
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.8/dist-pack
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.8/dist-pack
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.8/dist-packa
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.8/dist-pac
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,>=4.3.0 in /usr/local/lib/python3.
```

Figure 2.3



```
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packa
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-pa
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from
Building wheels for collected packages: lime
  Building wheel for lime (setup.py) ... done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283857 sha256=29e
  Stored in directory: /root/.cache/pip/wheels/e6/a6/20/cc1e293fcdb67ede666fed293cb8953
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
Question:
 Why are Superman and Wonder Woman a couple all of a sudden in the latest animated movi
Probability (Insincere) = 0.017244775985540535
Probability (Sincere) = 0.9827552240144595
True Class is: sincere
```

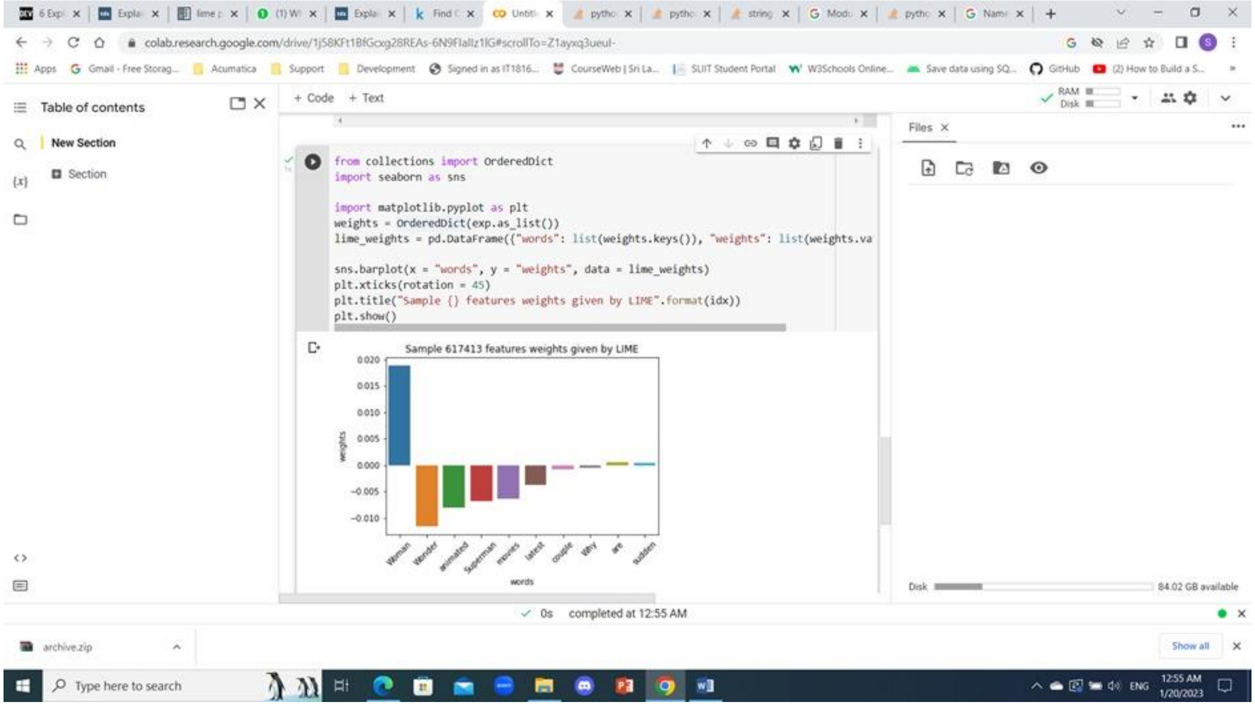
Figure 2.4



```
[11] Installing collected packages: lime
     Successfully installed lime-0.2.0.1
     Question:
      Why are Superman and Wonder Woman a couple all of a sudden in the latest animated movi
     Probability (Insincere) = 0.017244775985540535
     Probability (Sincere) = 0.9827552240144595
     True Class is: sincere
```

```
exp.show_in_notebook(text=val_df["question_text"][idx], labels=(1,))
```

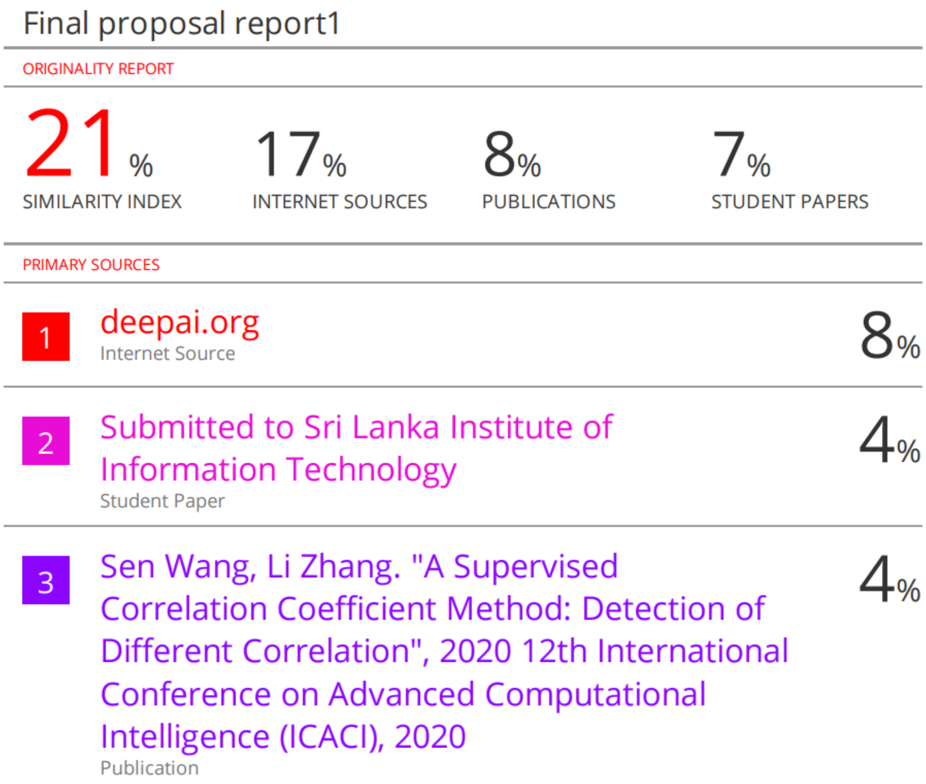
Figure 2.5



```
from collections import OrderedDict
import seaborn as sns

import matplotlib.pyplot as plt
weights = OrderedDict(exp.as_list())
lime_weights = pd.DataFrame({"words": list(weights.keys()), "weights": list(weights.va

sns.barplot(x = "words", y = "weights", data = lime_weights)
plt.xticks(rotation = 45)
plt.title("Sample {} features weights given by LIME".format(idx))
plt.show()
```

Figure 2.6

Similarity report



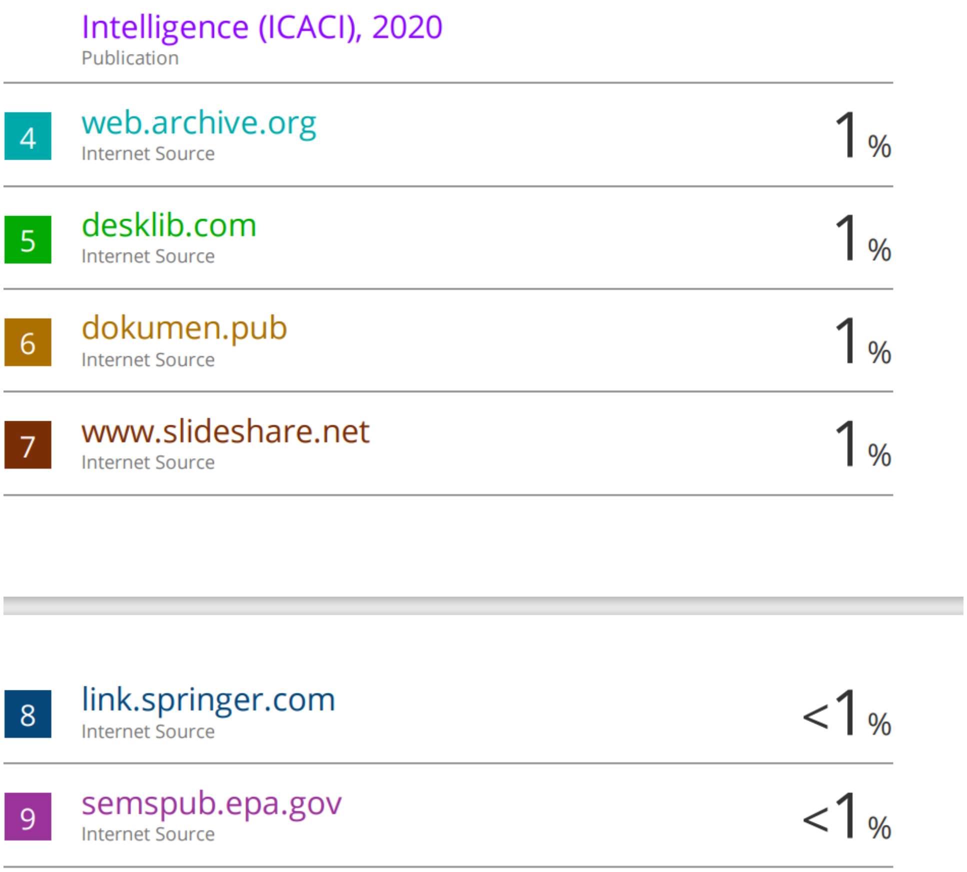
Final proposal report1

ORIGINALITY REPORT

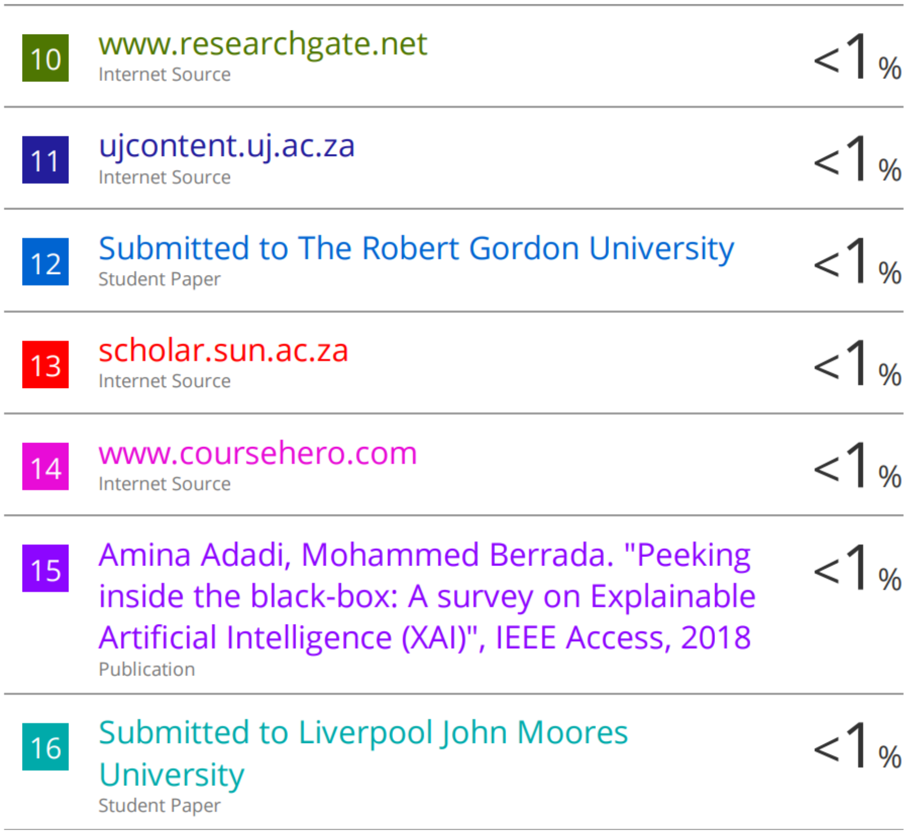
21%	17%	8%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	deepai.org Internet Source	8%
2	Submitted to Sri Lanka Institute of Information Technology Student Paper	4%
3	Sen Wang, Li Zhang. "A Supervised Correlation Coefficient Method: Detection of Different Correlation", 2020 12th International Conference on Advanced Computational Intelligence (ICACI), 2020 Publication	4%



	Intelligence (ICACI), 2020 Publication	
4	web.archive.org Internet Source	1 %
5	desklib.com Internet Source	1 %
6	dokumen.pub Internet Source	1 %
7	www.slideshare.net Internet Source	1 %
8	link.springer.com Internet Source	<1 %
9	semspub.epa.gov Internet Source	<1 %



10	www.researchgate.net Internet Source	<1 %
11	ujcontent.uj.ac.za Internet Source	<1 %
12	Submitted to The Robert Gordon University Student Paper	<1 %
13	scholar.sun.ac.za Internet Source	<1 %
14	www.coursehero.com Internet Source	<1 %
15	Amina Adadi, Mohammed Berrada. "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)", IEEE Access, 2018 Publication	<1 %
16	Submitted to Liverpool John Moores University Student Paper	<1 %

17	arxiv.org Internet Source	<1 %
18	webthesis.biblio.polito.it Internet Source	<1 %

19	www.science.gov Internet Source	<1 %
20	"Explainable Artificial Intelligence for Cyber Security", Springer Science and Business Media LLC, 2022 Publication	<1 %