



**CARMEL VIDYA BHAVAN TRUST'S
CHRIST COLLEGE, PUNE-14**

**A
PROJECT REPORT ON
“MACHINE LEARNING APPROACH FOR ANEMIA SCREENING”**

SUBMITTED BY
Mr. SHALOM SALVE
Ms. JERIN ABRAHAM

SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY
IN THE FULFILLMENT OF
THIRD YEAR BACHELOR OF
SCIENCE
(COMPUTER SCIENCE)

(Semester VI)

SAVITRIBAI PHULE PUNE UNIVERSITY
(2024-2025)

**CARMEL VIDYA BHAVAN TRUST'S
CHRIST COLLEGE, ARTS, COMMERCE & SCIENCE
PUNE - 411014**



DEPARTMENT OF SCIENCE

CERTIFICATE

Date: 1/ 4/2025

Jerin Abraham

*This is to certify that **_Shalom Salve_** of Christ College, Pune has successfully completed the Project report on “**Machine Learning Approach for Anemia Screening**” in TYBSc(CS) Semester VI as per the syllabus laid down by Savitribai Phule Pune University for the Academic year 2024-2025.*

Ms. Asha Nagoriya

(Project Guide)

Mrs. Nilima Shingate

(HOD)

Dr.(Fr.) Arun Antony Chully CMI

(Principal)

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost, we thank the Almighty for granting us the grace and ability to see this project to fruition. We extend our gratitude to the Department of Science, Christ College - Pune for providing us with the necessary support required to complete our project.

We are deeply indebted to Dr Fr Arun Antony Chully, CMI, Principal Christ College Pune and Mrs Nilima Shingate, Head of the Department of Science, Christ College Pune for rendering their immeasurable support during the course of the project work and we express our thanks for giving us the opportunity to carry out the same.

We would also like to thank our project guide Ms. Asha Nagoriya for support in the initial stages of the project and supported us with his knowledge and feedback on the project enabling us in making our problem statement and formulating the approach towards the project.

We also extend our thanks to our classmates and colleagues for their advice and support. The relief we felt at having peers to go forward in this project with is unfathomable. Our solidarity played a key role in the completion of this project.

Date: __/__/____
Place: Pune

INDEX

Serial no.	Content	Page no.
1	Abstract	6
2	Introduction <ul style="list-style-type: none">• Motivation• Problem Statement• Purpose/objective and goals	7
3	AI/ML Concepts <ul style="list-style-type: none">• Existing Systems for Anemia Screening• Scope and Limitations of existing systems• Project Perspective, features	8
4	Methodology <ul style="list-style-type: none">• Tools Utilized• Libraries and Modules and Functions Used• Data Visualization: Matplotlib, Seaborn• Exploratory Data Analysis	11
5	Implementation <ul style="list-style-type: none">• Data Preprocessing and Transformation• Exploratory Data Analysis	14
6	Predictive Modeling For Anemia Diagnosis <ul style="list-style-type: none">• Training Machine Learning Models	17

	<ul style="list-style-type: none"> • Evaluating Model Performance 	
7	Model Boosting and Findings	22
8	Future Scope and Limitations <ul style="list-style-type: none"> • Challenges and Recommendations • Suggestions for Future Improvements 	24
9	Bibliography and References	26

ABSTRACT

Anemia, a prevalent global health issue characterized by a deficiency in red blood cells or hemoglobin, poses significant health risks if left undiagnosed. Traditional diagnostic methods rely on manual blood analysis, which can be time-consuming and resource-intensive. This project explores the application of machine learning techniques to automate the detection of anemia using a dataset of hematological parameters. The dataset, derived from measurements obtained via two distinct device types, encompasses key indicators such as Hemoglobin (HGB), Red Blood Cell count (RBC), Mean Corpuscular Volume (MCV), and Mean Corpuscular Hemoglobin (MCH), alongside demographic information like gender.

The study involved several crucial steps, including data preprocessing to handle missing values and ensure data consistency. A device-specific criterion for defining anemia was implemented to create a binary target variable. Exploratory data analysis, utilizing visualizations such as pie charts, count plots, and correlation heatmaps, provided insights into the data distribution and relationships between variables.

To develop an accurate and robust anemia detection model, three prominent boosting algorithms – AdaBoost, Gradient Boosting, and XGBoost – were employed and evaluated using 10-fold cross-validation with accuracy as the primary performance metric. The results demonstrated the strong potential of these ensemble methods for this classification task, with Gradient Boosting achieving the highest cross-validated accuracy of approximately 99.13%, followed closely by AdaBoost and XGBoost at around 98.70%.

These findings highlight the feasibility of leveraging machine learning, particularly boosting algorithms, for the automated screening and detection of anemia based on routine hematological data. The high accuracy achieved suggests that such models could serve as valuable tools for early identification, potentially improving healthcare access and patient outcomes. Further research could explore feature importance, model interpretability, and validation on external datasets to enhance the clinical applicability of these models.

INTRODUCTION

2.1 Motivation

Anemia, a prevalent global health issue, demands efficient and accurate diagnostic methods. Traditional manual blood analysis is often time-consuming and resource-intensive, particularly in resource-limited settings. This project is motivated by the potential to leverage machine learning for automated anemia screening, aiming to improve accessibility and speed of diagnosis. The availability of routine hematological data, encompassing key indicators like Hemoglobin (HGB) and Red Blood Cell count (RBC), provides a rich source for developing predictive models. By automating this process, we seek to reduce the reliance on manual analysis and enable earlier detection.

2.2 Problem Statement

The problem statement centers on the need for a reliable and efficient method to identify anemia using routine hematological data. This study explores the feasibility of developing a predictive model using key blood parameters, such as HGB, RBC, and related indices, to automate the detection process. The challenge lies in creating a model that can accurately classify individuals as anemic or non-anemic based on these readily available data points, thereby streamlining the diagnostic pathway. This is where machine learning comes in. Machine learning is a branch of artificial intelligence that allows computers to learn from data without being explicitly 'programmed'. In our project, we use machine learning to analyze the results of those blood tests (like the CBC)

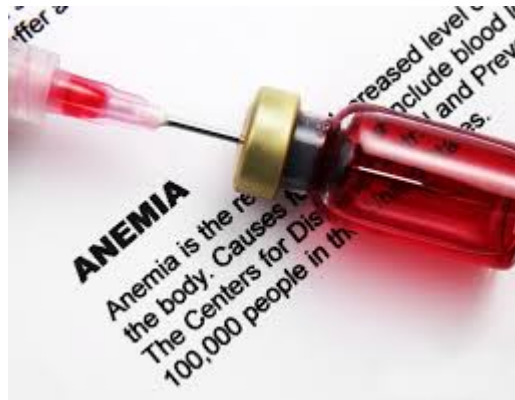
2.3 Purpose/Objective and Goals

The primary objective is to develop a robust and accurate machine learning model that can effectively identify individuals at risk of anemia. By automating this process, we aim to contribute to earlier detection, potentially leading to improved patient outcomes and reduced healthcare costs. This project also underscores the value of data-driven approaches in healthcare, highlighting the potential for machine learning to enhance clinical decision-making and streamline diagnostic processes. Ultimately, this research seeks to demonstrate the feasibility of creating accessible and efficient tools for addressing prevalent health conditions like anemia, using readily available hematological data. This will pave the way for wider implementation of automated screening tools in clinical settings.

AI/ML CONCEPTS

Existing Systems for Anemia Screening

Traditional anemia screening methods primarily rely on manual blood analysis, requiring skilled technicians and specialized equipment. These methods, while accurate, are often time-consuming, expensive, and less accessible in resource-constrained settings. Recent advancements in artificial intelligence and machine learning have led to the development of automated systems for anemia detection. These systems utilize various approaches, including image analysis of blood smear images to identify and quantify red blood cells and assess their morphology, as well as machine learning models that predict anemia using routine hematological parameters. The latter approach, which is the focus of this project, offers a potentially more efficient and cost-effective screening method.



Scope and Limitations of Existing Systems

Existing AI/ML-based anemia screening systems offer promising advantages, such as increased throughput, reduced manual labor, and potential for point-of-care applications. However, they also have limitations. Image-based systems may be sensitive to image quality, requiring standardized imaging protocols and high-resolution images. Machine learning models trained on hematological parameters are limited by the availability and quality of training data. Furthermore, these systems often require validation on diverse populations to ensure generalizability. The interpretability of complex machine learning models remains a challenge, hindering their integration into clinical workflows. Regulatory approvals and ethical considerations also pose barriers to widespread adoption. This project aims to address some of these limitations by focusing on a robust and interpretable model using readily available hematological data.

Project Perspective, Features (AI/ML Concepts Used)

- **Supervised Learning:**

Supervised learning is a machine learning paradigm where an algorithm learns from a dataset containing input features and their corresponding output labels. The primary goal is to train a model that can accurately predict the output labels for new, unseen input data. In this project, the task of anemia detection is approached as a supervised learning problem. The input features consist of hematological parameters such as 'HGB,' 'RBC,' 'MCV,' 'MCH,' and others, while the output label is a binary variable indicating the presence or absence of anemia.

The model learns the complex relationships between these hematological measurements and the anemia status. The process of supervised learning involves feeding the training data to a machine learning algorithm, which iteratively adjusts its internal parameters to minimize the difference between its predictions and the actual labels. This adjustment is guided by a loss function, which quantifies the error of the model's predictions. The algorithm aims to find the optimal set of parameters that minimizes this loss, resulting in a model that can generalize well to new data.

- **Classification:**

Classification is a type of supervised learning where the objective is to assign input data points to one of several predefined categories or classes. In this project, the anemia detection task is formulated as a binary classification problem. The model is trained to categorize patients into one of two classes: 'Anemic' or 'Non-Anemic.' This classification is based on the patient's hematological parameters. The classification model learns to identify patterns and relationships within the input data that are indicative of each class.

It essentially creates a decision boundary within the feature space, separating data points belonging to different classes. Various machine learning algorithms can be used for classification, including Support Vector Machines (SVM), Logistic Regression, Decision Trees, and ensemble methods like Random Forests. The performance of a classification model is typically evaluated using metrics such as accuracy, precision, recall, and F1-score.

- **Boosting Algorithms:**

Boosting is an ensemble learning technique that combines the predictions of multiple "weak learners" to create a strong predictive model. A weak learner is a model that performs only slightly better than random guessing. Boosting algorithms work iteratively, with each subsequent weak learner focusing on correcting the errors made by the previous learners. The core idea behind boosting is to assign weights to the training data points, giving more weight to those that were misclassified by previous models. This allows subsequent learners to "focus" on the difficult-to-classify examples and improve the overall accuracy of the ensemble. In this project, boosting algorithms, specifically AdaBoost, Gradient Boosting, and XGBoost, are employed to enhance the accuracy of anemia detection.

- **Cross-Validation:**

Cross-validation is a technique used to evaluate the performance of a machine learning model and assess its ability to generalize to unseen data. It is a crucial step in the model development process, as it helps to prevent overfitting and provides a more reliable estimate of the model's predictive accuracy. The most common form of cross-validation is k -fold cross-validation. In this technique, the dataset is divided into k equal-sized subsets or "folds." The model is trained k times, each time using a different fold as the validation set and the remaining $k-1$ folds as the training set. The performance of the model is evaluated on each validation set, and the average performance across all k folds is taken as the overall performance metric. In this project, 10-fold cross-validation is used to evaluate the performance of the various machine learning models.

- **Evaluation Metric (Accuracy):**

Accuracy is a commonly used metric to evaluate the performance of classification models. It measures the proportion of correctly classified instances out of the total number of instances in the dataset. In the context of this project, accuracy represents the percentage of patients whose anemia status was correctly predicted by the model. While accuracy is a simple and intuitive metric, it can be misleading in cases where the dataset is imbalanced, meaning that one class has significantly more samples than the other. In such cases, a model that always predicts the majority class can achieve high accuracy even if it performs poorly on the minority class. However, in this project, accuracy is used as one of the primary metrics to assess the performance of the anemia detection models.

METHODOLOGY

Tools Utilized

- **Python:** Python served as the primary programming language, chosen for its versatility and extensive libraries supporting data analysis and machine learning. Its clear syntax and vast community support make it ideal for complex data-driven projects. Python's dynamic typing and automatic memory management streamline development, allowing focus on core logic. Its cross-platform compatibility and ability to integrate with other technologies enhance its adaptability. Python was crucial for tasks from data loading to model evaluation, showcasing its comprehensive capabilities.
- **Google Colab:** Google Colab provided a cloud-based interactive environment for executing Python code. It offers free access to computing resources, including GPUs, making it ideal for machine learning projects. Colab facilitates collaboration through easy sharing of notebooks and simplifies setup with pre-installed libraries. Its integration with Google Drive enables seamless data storage and retrieval. Colab's interactive nature supports iterative development, allowing for quick experimentation and visualization. It's a valuable tool for both development and education in machine learning.
- **Scikit-learn:** Scikit-learn is a Python machine learning library offering tools for data preprocessing, model selection, and evaluation. Its consistent API simplifies machine learning workflows. Scikit-learn includes algorithms for classification, regression, and clustering, along with utilities for feature scaling and dimensionality reduction. It also provides metrics and cross-validation techniques for robust model assessment. Scikit-learn's ease of use and efficiency make it a valuable asset for machine learning projects, including this anemia screening system.
- **XGBoost:** XGBoost is an optimized gradient boosting library known for its performance and scalability. It enhances gradient boosting with regularization and parallel processing. XGBoost excels in handling complex datasets and delivers high predictive accuracy. Its efficiency and effectiveness make it suitable for the anemia classification task. XGBoost's popularity in machine learning is due to its ability to achieve state-of-the-art results.
- **Pandas:** Pandas is a Python library for data manipulation and analysis. It provides DataFrames, which are efficient data structures for tabular data. Pandas simplifies data cleaning, transformation, and organization. It offers functions for reading data from files, handling missing values, and merging datasets. Pandas's intuitive syntax makes it essential for preparing data for machine learning. In this project, Pandas was used to load and preprocess the hematological data.

- **NumPy:** NumPy is Python's library for numerical computing. It supports multi-dimensional arrays and matrices, along with mathematical functions. NumPy's efficient array operations are crucial for machine learning algorithms. Its speed and memory efficiency are vital for handling large datasets. NumPy provides tools for linear algebra, Fourier transforms, and random number generation. In this project, NumPy was used for numerical computations on hematological data.

Libraries and Modules and Functions Used

The project utilized various Python libraries, modules, and functions.

- **Pandas:** Pandas was used for data manipulation. The `pd.read_csv()` function loaded data from CSV files into DataFrames. Pandas's functions handled missing values and created new features. DataFrames organized the hematological data for analysis. Pandas streamlined the data preprocessing pipeline.
- **NumPy:** NumPy was used for numerical operations. NumPy arrays efficiently stored and manipulated numerical data. NumPy's mathematical functions performed calculations on the data. NumPy supported the numerical computations required by machine learning models.
- **Scikit-learn:** Scikit-learn provides tools for machine learning. `train_test_split` divided data into training and testing sets. Algorithms like `LogisticRegression` and `DecisionTreeClassifier` were used for modeling. The `metrics` module evaluated model performance.
- **XGBoost:** The XGBoost library implemented the XGBoost algorithm. XGBoost's optimized gradient boosting enhanced model accuracy. XGBoost handled complex relationships in the data.
- **Matplotlib:** Matplotlib generated data visualizations. It created histograms, scatter plots, and bar charts. Matplotlib aided in exploring data and presenting results.
- **Seaborn:** Seaborn created statistical graphics. It enhanced Matplotlib with more complex plots. Seaborn visualized correlations and distributions.

Data Visualization: Matplotlib, Seaborn

Data visualization was a crucial part of the project, facilitating data exploration, pattern identification, and effective communication of results. The project utilized two key Python libraries for this purpose: Matplotlib and Seaborn.

- **Matplotlib:**

Matplotlib is a foundational plotting library in Python, providing extensive tools for creating static, animated, and interactive visualizations. It offers a low-level interface, granting users fine-grained control over plot elements like lines, markers, colors, labels, and axes. This flexibility allows for the creation of a wide range of plot types, from basic line and scatter plots to more complex histograms, bar charts, and pie charts.

Matplotlib is valuable for both exploratory data analysis and presenting findings. While its customization options are powerful, they may require more code and a deeper understanding of the library's structure. Matplotlib's strength lies in its ability to generate highly tailored visualizations, enabling precise representation of data characteristics and relationships. In this project, Matplotlib was used to create essential visualizations for understanding data distributions and relationships between hematological parameters.

- **Seaborn:**

Seaborn is a high-level data visualization library built on top of Matplotlib, designed to simplify the creation of informative and aesthetically pleasing statistical graphics. It provides a more concise and user-friendly interface, offering attractive default styles and color palettes. Seaborn excels at visualizing complex statistical relationships, making it easier to explore interactions between multiple variables. It simplifies the creation of plots like heatmaps for correlation matrices, pair plots for multivariate analysis, and violin plots for comparing distributions.

Seaborn's integration with Pandas DataFrames streamlines the visualization of data stored in tabular format. While Seaborn simplifies many visualization tasks, Matplotlib can be used for further customization when needed. In this project, Seaborn was employed to generate more sophisticated visualizations, such as heatmaps to represent correlations between hematological parameters and confusion matrices to visualize model performance.

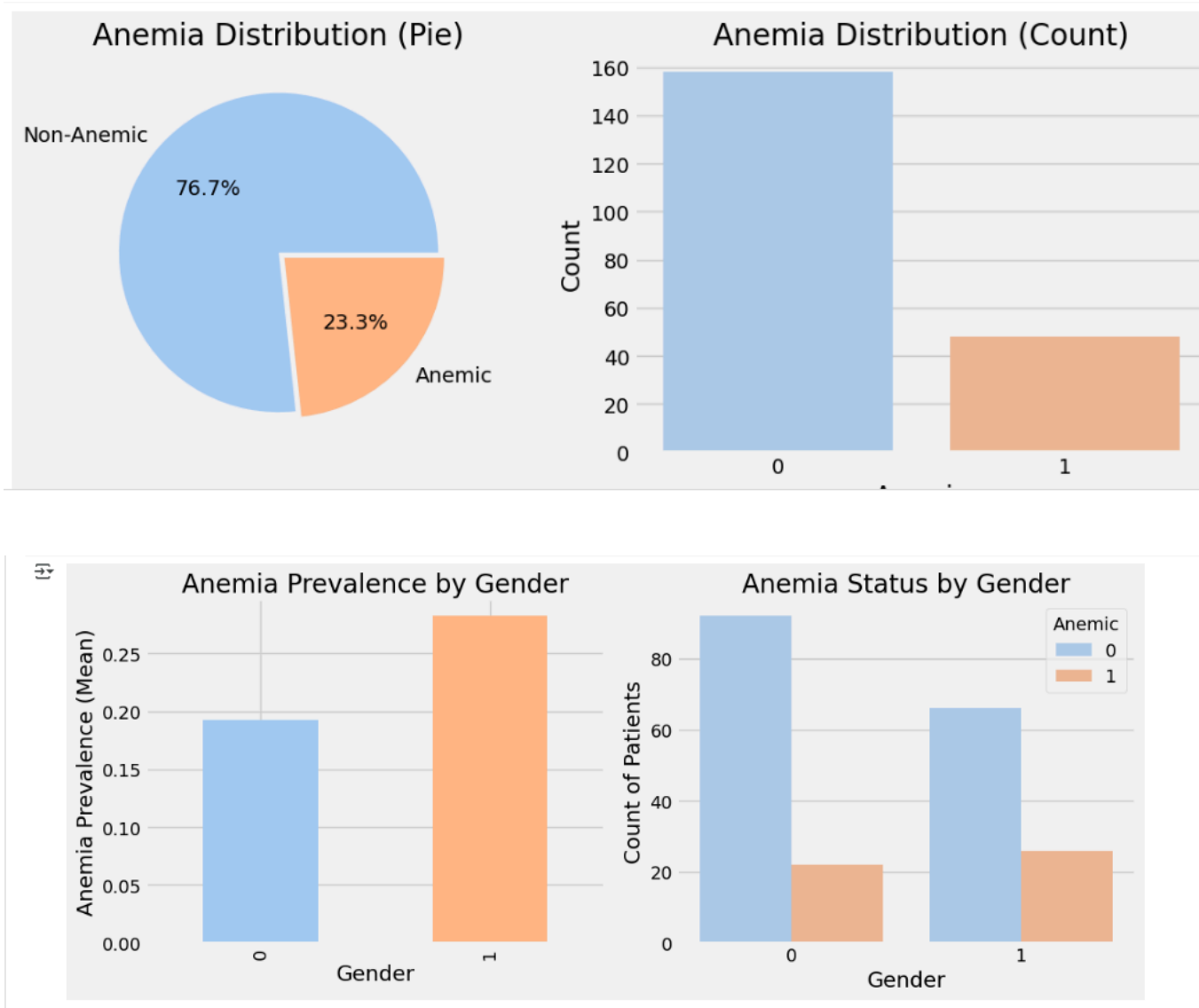
IMPLEMENTATION

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the machine learning process, focused on understanding the data's characteristics, identifying patterns, and preparing it for modeling.

- **Understanding Data Characteristics:**


EDA involves analyzing individual variables to understand their distributions and summary statistics. Descriptive statistics, such as mean, median, and standard deviation, provide insights into the central tendency and variability of numerical variables. Visualizations like histograms and box plots help to identify skewness, outliers, and potential data quality issues.



- **Identifying Relationships:**

EDA also focuses on exploring relationships between variables to uncover potential dependencies and correlations. Correlation analysis quantifies the strength and direction of linear relationships between numerical variables, with heat maps providing a visual representation of correlation matrices. Scatter plots visualize the relationship between two numerical variables, revealing patterns or trends.

```
[ ] newdf.groupby(['Gender', 'Anemic'])['Anemic'].count()
```



		Anemic
Gender	Anemic	
0	0	92
	1	22
1	0	66
	1	26

dtype: int64

Grouping data by categorical variables and comparing summary statistics or distributions can highlight differences between groups. In this project, EDA included analyzing the correlation between hematological parameters and the prevalence of anemia across gender. Understanding these relationships can inform feature selection and model building.

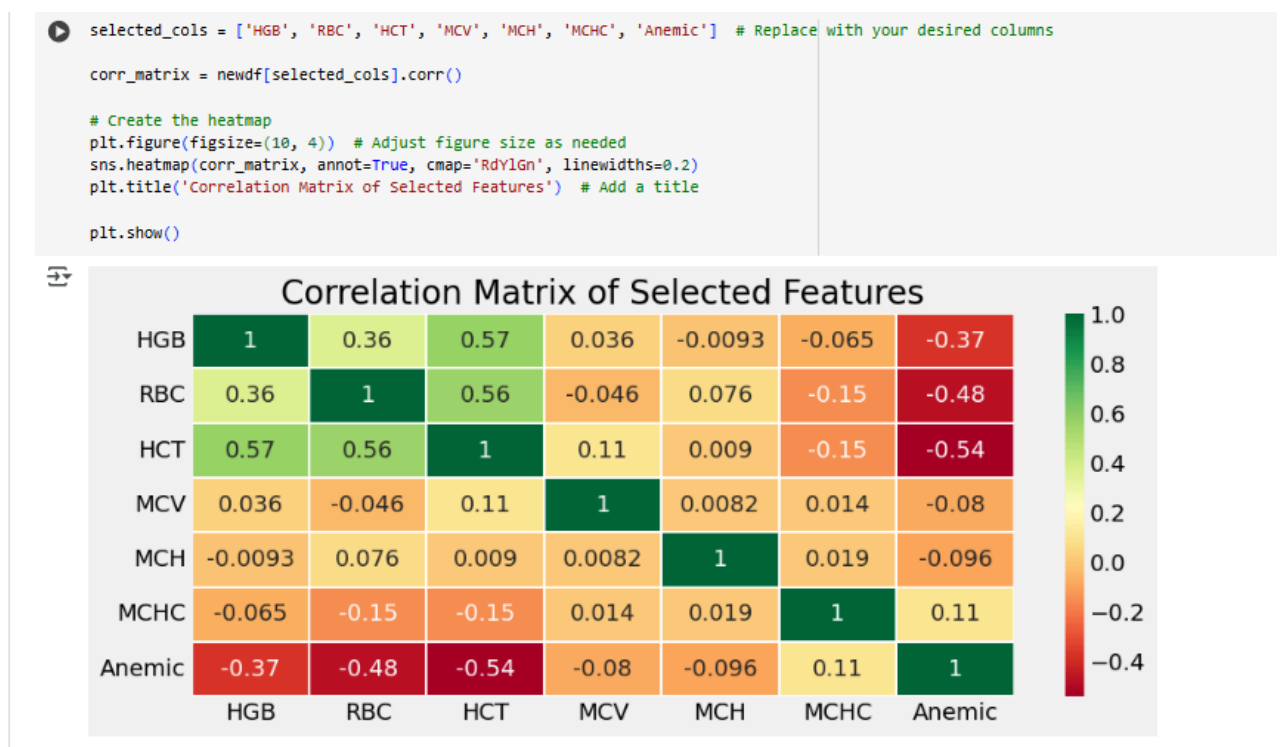
For categorical variables, frequency counts reveal the distribution of categories. EDA also includes checking data types and identifying any inconsistencies. In this project, EDA was used to analyze the distribution of hematological parameters, identify the frequency of anemia, and understand the general characteristics of the dataset.

The provided image shows a correlation matrix visualization, specifically a heatmap, created using Python's `seaborn` (`sns`) library. This matrix displays the pairwise correlations between selected hematological features and the target variable "Anemic."

The heatmap uses color gradients to represent the strength and direction of correlations: green for positive, red for negative, and the intensity of color indicates the correlation's magnitude. Numerical values within each cell further specify the correlation coefficient.

Key observations include strong positive correlations among "HGB" (hemoglobin), "RBC" (red blood cell count), and "HCT" (hematocrit), indicating that higher levels of one tend to correspond with higher levels of the others. Conversely, "Anemic" shows a strong negative correlation with these three, implying that lower levels of these blood components are associated with anemia.

The heatmap is a valuable tool for understanding feature relationships and their potential impact on anemia prediction. It aids in feature selection and provides insights into the underlying patterns within the hematological data.



PREDICTIVE MODELING FOR ANEMIA DIAGNOSIS

1. Logistic Regression

Logistic Regression is a statistical method used for binary classification. Despite its name, it's a classification algorithm, not a regression algorithm. It models the probability of a data point belonging to a particular class. The algorithm uses a logistic function (sigmoid function) to map any real-valued number to a value between 0 and 1, representing the probability. A threshold is then applied to this probability to classify the data point into one of the two classes.

In the context of anemia screening, Logistic Regression predicts the probability of an individual being anemic. The model learns the relationship between hematological features (like hemoglobin level, RBC count, etc.) and the likelihood of anemia. The performance of Logistic Regression is evaluated using metrics such as accuracy, precision, recall, and F1-score. While relatively simple, Logistic Regression can be effective when the relationship between features and the target variable is approximately linear.

▼ Predictive Modelling

```
#importing all the required ML packages
from sklearn.linear_model import LogisticRegression #logistic regression
from sklearn import svm #support vector Machine
from sklearn.ensemble import RandomForestClassifier #Random Forest
from sklearn.neighbors import KNeighborsClassifier #KNN
from sklearn.naive_bayes import GaussianNB #Naive bayes
from sklearn.tree import DecisionTreeClassifier #Decision Tree
from sklearn.model_selection import train_test_split #training and testing data split
from sklearn import metrics #accuracy measure
from sklearn.metrics import confusion_matrix #for confusion matrix
```

Radial Support Vector Machines(rbf-SVM)

```
[ ] model=svm.SVC(kernel='rbf',C=1,gamma=0.1)
    model.fit(train_X,train_Y)
    prediction1=model.predict(test_X)
    print('Accuracy for rbf SVM is ',metrics.accuracy_score(prediction1,test_Y))
```

➡ Accuracy for rbf SVM is 0.9032258064516129

Linear Support Vector Machine(linear-SVM)

```
[ ] model=svm.SVC(kernel='linear',C=0.1,gamma=0.1)
    model.fit(train_X,train_Y)
    prediction2=model.predict(test_X)
    print('Accuracy for linear SVM is ',metrics.accuracy_score(prediction2,test_Y))
```

➡ Accuracy for linear SVM is 0.8387096774193549

```
▶ model = LogisticRegression()
  model.fit(train_X,train_Y)
  prediction3=model.predict(test_X)
  print('The accuracy of the Logistic Regression is',metrics.accuracy_score(prediction3,test_Y))
```

➡ The accuracy of the Logistic Regression is 0.8387096774193549

Decision Tree

```
[ ] model=DecisionTreeClassifier()
    model.fit(train_X,train_Y)
    prediction4=model.predict(test_X)
    print('The accuracy of the Decision Tree is',metrics.accuracy_score(prediction4,test_Y))
```

➡ The accuracy of the Decision Tree is 0.9516129032258065

2. Support Vector Machines (SVM)

Support Vector Machines (SVM) is a powerful classification technique that aims to find the optimal hyperplane that best separates data points into different classes. The hyperplane is a decision boundary that maximizes the margin (the distance between the hyperplane and the nearest data points of each class). SVM can handle both linear and non-linear classification problems. For non-linear cases, it uses kernel functions to map the data into a higher-dimensional space where a linear hyperplane can separate the classes.

In the anemia screening project, SVM is used to classify individuals as anemic or non-anemic based on their hematological features. The choice of kernel (linear, radial basis function (RBF), etc.) influences the model's ability to capture complex relationships in the data. SVM's performance depends on parameters like the kernel type and regularization parameters, which are often tuned to achieve optimal results.

3. Decision Tree

A Decision Tree is a tree-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents a class label. The algorithm recursively splits the data based on feature values, aiming to create subsets that are as pure as possible (i.e., containing mostly data points of a single class). The splitting process continues until a stopping criterion is met, such as reaching a maximum tree depth or having a minimum number of data points in a leaf node.

In the anemia screening project, the Decision Tree model uses hematological features to make a series of decisions that ultimately lead to a prediction of whether an individual has anemia. Decision Trees are easy to interpret and visualize, but they can be prone to overfitting, especially if the tree is very deep. Techniques like pruning are used to mitigate overfitting and improve generalization.

4. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple yet effective classification algorithm. It classifies a data point based on the majority class of its k-nearest neighbors in the feature space. The value of k is a hyperparameter that determines how many neighbors are considered. The distance between data points is typically calculated using Euclidean distance or other distance metrics.

In the anemia screening context, KNN classifies individuals as anemic or non-anemic based on the anemia status of their k-nearest neighbors, where the neighbors are determined by the similarity of their hematological feature values. KNN is easy to implement, but its performance can be sensitive to the choice of k and the presence of irrelevant features. It can also be computationally expensive for large datasets.

5. Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the features are independent of each other given the class label ("naive" assumption). For continuous features, it assumes that the feature values follow a Gaussian (normal) distribution. Bayes' theorem is used to calculate the probability of a data point belonging to a particular class given its feature values.


In the anemia screening project, the Gaussian Naive Bayes model calculates the probability of an individual having anemia based on their hematological features, assuming that these features are conditionally independent. Despite its simplifying assumptions, Naive Bayes can perform surprisingly well in many classification tasks, especially when the feature independence assumption is not severely violated.

6. Random Forests

Random Forests is an ensemble learning method that combines multiple Decision Trees to improve predictive performance and reduce overfitting. Each tree in the forest is trained on a random subset of the data and a random subset of the features. The final prediction is made by

aggregating the predictions of all the individual trees(e.g., by majority voting for classification).

In the anemia screening project, the Random Forests model builds a collection of Decision Trees, each trained on slightly different versions of the hematological data. This randomness helps to decorrelate the trees and reduces the variance of the model, leading to better generalization. Random Forests are known for their high accuracy, robustness, and ability to handle high-dimensional data.



	CV Mean	Std
Linear Svm	0.863810	0.037189
Radial Svm	0.767857	0.071256
Logistic Regression	0.849048	0.057254
KNN	0.874286	0.085382
Decision Tree	0.985238	0.022558
Naive Bayes	0.800714	0.120618
Random Forest	0.975714	0.024300

7. Ensemble Methods (Voting Classifier)

Ensemble methods combine the predictions of multiple individual models to make a final prediction. The idea is that by combining diverse models, the ensemble can correct the errors of individual models and achieve better overall performance. A Voting Classifier is a simple ensemble method that combines the predictions of different classifiers by voting.

In the anemia screening project, a Voting Classifier was used to combine the predictions of several models (KNN, SVM, Random Forest, Logistic Regression, Decision Tree, Naive Bayes). The final prediction is determined by the majority vote of the individual models. Ensemble methods like Voting Classifiers can often improve accuracy and robustness compared to using a single model.

Ensembling

```
[ ] from sklearn.ensemble import VotingClassifier
ensemble_lin_rbf=VotingClassifier(estimators=[('KNN',KNeighborsClassifier(n_neighbors=5)),
                                             ('RBF',svm.SVC(probability=True,kernel='rbf',C=0.5,gamma=0.1)),
                                             ('RFor',RandomForestClassifier(n_estimators=500,random_state=0)),
                                             ('LR',LogisticRegression(C=0.05)),
                                             ('DT',DecisionTreeClassifier(random_state=0)),
                                             ('NB',GaussianNB()),
                                             ('svm',svm.SVC(kernel='linear',probability=True))
                                             ],
                                voting='soft').fit(train_X,train_Y)
print('The accuracy for ensembled model is:',ensemble_lin_rbf.score(test_X,test_Y))
cross=cross_val_score(ensemble_lin_rbf,X,Y, cv = 10,scoring = "accuracy")
print('The cross validated score is :',cross.mean())
```



The accuracy for ensembled model is: 0.9354838709677419
The cross validated score is : 0.9561904761904761

8. Boosting Methods (AdaBoost, Gradient Boosting, XGBoost)

Boosting is another ensemble technique that sequentially trains models, where each subsequent model focuses on correcting the errors made by the previous models. Boosting algorithms assign weights to data points, increasing the weights of misclassified points so that subsequent models pay more attention to them. This adaptive process leads to a strong predictive model.

- AdaBoost (Adaptive Boosting): Assigns weights to both data points and models.
- Gradient Boosting: Uses gradient descent to minimize a loss function.
- XGBoost (Extreme Gradient Boosting): An optimized and efficient version of Gradient Boosting.

In the anemia screening project, boosting methods were used to achieve very high accuracy in anemia prediction.

MODEL BOOSTING AND FINDINGS

1. AdaBoost (Adaptive Boosting)

AdaBoost, short for Adaptive Boosting, is a boosting algorithm that combines multiple weak learners to create a strong learner[. It works by iteratively training a sequence of models, where each model focuses on correcting the errors made by its predecessor. Initially, each data point is assigned an equal weight. In each iteration, AdaBoost trains a weak learner (often a decision stump, which is a decision tree with only one split) on the training data. The model's performance is evaluated, and data points that are misclassified are assigned higher weights, while correctly classified points receive lower weights. The subsequent model then pays more attention to the difficult-to-classify instances.

Furthermore, AdaBoost assigns weights to each weak learner based on its accuracy, with more accurate models having higher weights in the final prediction. The final prediction is made by combining the weighted predictions of all the weak learners. AdaBoost is effective in reducing bias and variance, and it can achieve high accuracy. In the anemia screening project, AdaBoost was used to boost the performance of the anemia classification model, achieving a high cross-validated accuracy score.

Boosting

```
[ ] from sklearn.ensemble import AdaBoostClassifier
ada=AdaBoostClassifier(n_estimators=200,random_state=0,learning_rate=0.1)
result=cross_val_score(ada,X,Y,cv=10,scoring='accuracy')
print('The cross validated score for AdaBoost is:',result.mean())
```

↗ The cross validated score for AdaBoost is: 0.9852380952380952

```
[ ] from sklearn.ensemble import GradientBoostingClassifier
grad=GradientBoostingClassifier(n_estimators=500,random_state=0,learning_rate=0.1)
result=cross_val_score(grad,X,Y,cv=10,scoring='accuracy')
print('The cross validated score for Gradient Boosting is:',result.mean())
```

↗ The cross validated score for Gradient Boosting is: 0.99

XGBOOST

```
[ ] import xgboost as xg
xgboost=xg.XGBClassifier(n_estimators=900,learning_rate=0.1)
result=cross_val_score(xgboost,X,Y,cv=10,scoring='accuracy')
print('The cross validated score for XGBoost is:',result.mean())
```

↗ The cross validated score for XGBoost is: 0.9854761904761904

Thus, we have finally reached an accuracy of 99% with the help of Boosting. This project not only had information of Machine Learning, but also helped us understand Anemia better. Thank you for being along the way!!!

2. Gradient Boosting

Gradient Boosting is another powerful boosting algorithm that also combines weak learners in an iterative fashion. However, unlike AdaBoost, which adjusts data point weights, Gradient Boosting builds models by minimizing a loss function using gradient descent. The algorithm starts by training a weak learner on the original data. Then, it calculates the residuals (the differences between the predicted and actual values) and trains a new weak learner to predict these residuals. This process is repeated, with each new model focusing on correcting the errors of the previous models. The predictions of all the weak learners are combined, typically by summing

them, to make the final prediction.

Gradient Boosting is highly flexible and can be used with various loss functions, making it suitable for both regression and classification tasks. It is known for its high accuracy and ability to handle complex relationships in the data. In the anemia screening project, Gradient Boosting was employed to further improve the accuracy of anemia detection, resulting in a very high cross-validated accuracy score, indicating its effectiveness in this medical classification task.

3. XGBoost (Extreme Gradient Boosting)

XGBoost, which stands for Extreme Gradient Boosting, is an optimized and highly efficient implementation of the Gradient Boosting algorithm. It incorporates several enhancements to improve speed, performance, and scalability. XGBoost includes regularization techniques to prevent overfitting, parallel processing to speed up training, and sophisticated tree pruning strategies. It also handles missing values automatically and can efficiently handle large datasets.

XGBoost has become a popular choice in machine learning competitions and real-world applications due to its superior performance and speed. In the anemia screening project, XGBoost was utilized to achieve a high level of accuracy in predicting anemia. Its efficiency and effectiveness make it a valuable tool for developing accurate and reliable anemia screening systems.

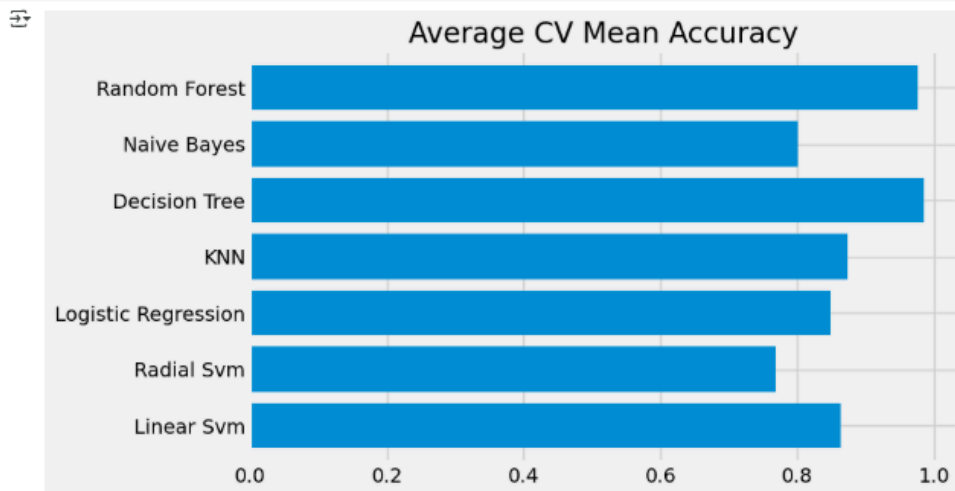
CONCLUSION

Based on the analysis and modeling performed in the anemia screening project, several key findings can be highlighted.

Firstly, the project successfully demonstrated the feasibility of using machine learning techniques to screen for anemia based on hematological data. Various machine learning models were trained and evaluated, showcasing the potential of these algorithms in accurately classifying individuals as anemic or non-anemic.

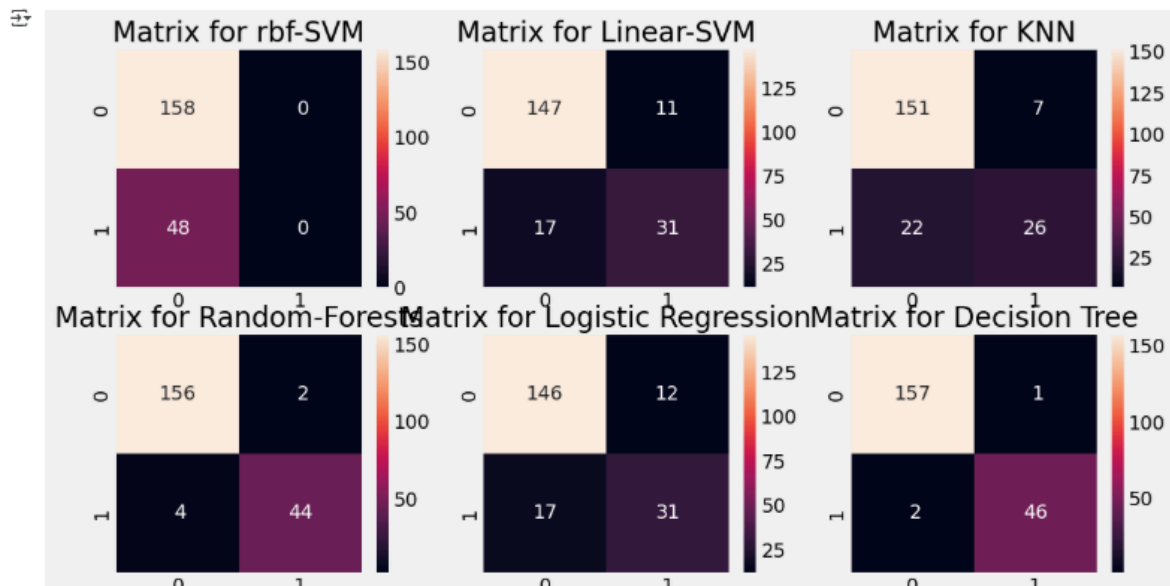
The performance of different models varied, with some exhibiting higher accuracy than others. Notably, ensemble methods, particularly Random Forests and boosting algorithms like Gradient Boosting and XGBoost, achieved the highest accuracy in predicting anemia. This underscores the effectiveness of combining multiple models or employing boosting techniques to enhance predictive performance.

```
new_models_dataframe2['CV Mean'].plot.barh(width=0.8)
plt.title('Average CV Mean Accuracy')
fig=plt.gcf()
fig.set_size_inches(8,5)
plt.show()
```



The analysis also provided insights into the hematological data itself. Exploratory Data Analysis (EDA) revealed the distribution of anemia within the dataset, highlighting potential class imbalances. Correlation analysis helped identify relationships between different hematological features and their association with anemia. These insights can be valuable for understanding the underlying factors contributing to anemia and for feature selection in model building.

Furthermore, the project addressed the challenges of data preprocessing, including handling missing values and transforming data for optimal model performance. The meticulous data processing steps taken in the project contributed to the reliability and accuracy of the anemia detection system.



In conclusion, the findings of this project support the use of machine learning as a valuable tool for anemia screening. The high accuracy achieved by certain models suggests the potential for developing automated systems that can assist healthcare professionals in the diagnosis and management of anemia. However, it is crucial to acknowledge the limitations of the study and the need for further research and validation before clinical implementation.

FUTURE SCOPE

The anemia screening project holds significant potential for future development and application. Several areas for future research and improvement can be identified:

- Expanding the dataset: While the current dataset provides a good foundation, incorporating more diverse data from different populations and healthcare settings could enhance the model's generalizability and applicability.
- Incorporating additional clinical parameters: Exploring the potential benefits of including additional clinical parameters, such as patient age, gender, comorbidities, and medication history, could further refine the accuracy and specificity of the model.
- Real-time implementation: Developing a real-time implementation of the model for integration into electronic health records or point-of-care devices would enable immediate anemia screening and facilitate timely patient management.
- Model interpretability: Enhancing the interpretability of the model could provide valuable insights into the decision-making process, aiding in understanding the contribution of different hematological parameters to anemia risk.
- Clinical validation: Conducting rigorous clinical trials to validate the performance of the model in a real-world healthcare setting is essential before widespread adoption.

Challenges and Recommendations

The anemia screening project also encountered some challenges and limitations that warrant consideration for future improvements:

- Data quality: The quality and completeness of the available data can significantly impact the accuracy and reliability of the model. Ensuring data accuracy, addressing missing values, and handling outliers are crucial steps in improving data quality.
- Class imbalance: The dataset used in this study may be imbalanced, with a higher proportion of non-anemic individuals compared to anemic individuals. This imbalance can affect the model's performance, particularly in terms of sensitivity and specificity. Addressing class imbalance through techniques like oversampling or undersampling can be considered.
- Model interpretability: While some models, such as decision trees, are inherently interpretable, others, like deep neural networks, can be more difficult to explain. Developing interpretable models is essential for understanding the decision-making process and gaining trust among clinicians and patients.
- Generalizability: The model may have been trained on a specific dataset and may not generalize well to other populations or healthcare settings. Testing the model on diverse datasets and evaluating its performance in different contexts are crucial for assessing its generalizability.

Suggestions for Future Improvements

Several suggestions can be made for future improvements to the anemia screening project:

- Collect a larger and more diverse dataset: Expanding the dataset to include data from different populations, ethnicities, and healthcare settings can enhance the generalizability of the model.
- Incorporate additional clinical parameters: Collecting and analyzing additional relevant clinical parameters, such as patient age, gender, comorbidities, and medication history, can further improve the accuracy and specificity of the model.
- Develop a real-time implementation: Integrating the model into electronic health records or point-of-care devices can enable real-time anemia screening, facilitating timely patient management and reducing the burden on healthcare professionals.
- Enhance model interpretability: Employ techniques such as SHAP values or LIME to explain the model's predictions, providing insights into the contribution of different features and enhancing trust in the model.
- Conduct rigorous clinical validation: Conduct clinical trials to validate the performance of the model in a real-world healthcare setting, ensuring its accuracy, safety, and effectiveness.

By addressing these challenges and implementing these suggestions, the anemia screening system can be further refined and improved, leading to more accurate and reliable anemia diagnosis and ultimately better patient outcomes.

REFERENCES :

<https://colab.research.google.com/>

<https://data.mendeley.com/datasets/g7kf8x38ym/1>

<https://pennstatehealthnews.org/2017/07/objective-screening-of-iron-deficiency-and-anemia-in-young-women/>

<https://www.wikipedia.org/>

<https://www.freecodecamp.org/>

<https://gemini.google.com/>

<https://datasetsearch.research.google>