

Internship Report

*An Internship report submitted to ICT Academy of
Kerala in partial fulfillment of the requirements
for the certification of*

CERTIFIED SPECIALIST IN DATA SCIENCE & ANALYTICS

Submitted by

Aishwarya Narayanan



**ICT ACADEMY OF KERALA
THIRUVANANTHAPURAM, KERALA, INDIA
February 2024**

Aim

The objective of the Credit Score Classification Model Development is to construct a predictive model capable of categorizing customers into three distinct creditworthiness levels: Good, Standard, and Poor. The processes done throughout this internship will engage in the development of a tool with significant implications for the financial sector. The ultimate aim is to provide banks and financial institutions with a reliable mechanism to enhance decision-making processes concerning loan approvals and risk assessment.

Dataset Overview

Dataset Name - credit.csv

The dataset comprises 58,016 entries across 28 columns and 1,00,000 rows, offering a comprehensive view of customer financial profiles. Key columns include unique identifiers such as ID and Customer_ID, demographics like Age and Occupation, and many financial metrics such as Annual_Income, Monthly_Income_Salary, Outstanding_Debt, Total_EMI_per_month, etc.

The dataset's diversity is reflected in the inclusion of numerical (integers and floats) and categorical (objects) data types. Areas of exploration could encompass payment behaviour, the relationship between credit score and financial features, and the impact of credit mix on creditworthiness. Handling missing values is crucial for robust analysis and it is found that there are no missing values.

Overall, this dataset provides a rich foundation for examining financial patterns and informing strategies in areas like risk assessment and credit evaluation.

Primary Objective

The goal of developing the Credit Score Classification Model is to create a predictive framework that can effectively classify customers into three discernible creditworthiness categories: Good, Standard, and Poor. The objective remains focused on constructing a model capable of assessing and categorizing individuals

based on their creditworthiness, thereby providing valuable insights into their financial reliability. The emphasis is on creating a robust and accurate predictive tool that aids in the systematic classification of customers, facilitating informed decision-making processes related to risk assessment and credit evaluation. The successful implementation of this model is anticipated to contribute substantially to the efficiency and accuracy of credit evaluations within the industry.

Overview

As a data scientist in a prominent global finance corporation, a substantial undertaking awaits. The company has accumulated an extensive dataset comprising vital banking details and a comprehensive repository of credit-related information. To enhance efficiency and accuracy, the management has launched a forward-looking initiative. The aim is to create an intelligent system using data science and machine learning for the automated categorization of individuals into specific credit score brackets.

This visionary project serves a dual purpose. Firstly, it seeks to streamline credit assessment processes, reducing manual efforts. Secondly, it aspires to offer more tailored financial services by precisely classifying customers based on their creditworthiness. As the lead data scientist for this initiative, responsibilities include data analysis, predictive modeling, and implementing machine learning techniques. The objective is to establish a robust system capable of autonomously evaluating and assigning individuals to appropriate credit score categories.

This project signifies the convergence of data-driven insights and cutting-edge technology, poised to reshape the credit assessment landscape in the finance industry.

Steps Involved

1. Data Pre-processing
2. Exploratory Data Analysis (EDA)
3. Data Splitting
4. Feature Engineering
5. Model Selection
6. Model Training
7. Model Evaluation
8. Hyperparameter Tuning
9. Cross-Validation
10. Model Interpretability
11. Performance Validation
12. Threshold Selection
13. Testing and Validation

Data Pre-Processing

Data pre-processing plays a crucial role in optimizing the dataset's quality and facilitating accurate analyses within the realm of this financial dataset. As a first step, a basic idea on the dataset was collected which involved steps like:

- Finding the shape of the dataset.
- Getting an idea on the categorical and numeric columns
- Understanding the statistical relation between columns
- Checking for missing values or duplicate rows, etc.

Furthermore, standardizing or normalizing numerical features like Annual_Income, Monthly_Inhand_Salary and others aids in eliminating scale-related biases, enabling fair comparisons across different metrics.

Categorical variables such as Occupation, Type_of_Loan, and Credit_Mix demand encoding to convert them into a format suitable for machine learning algorithms. This step ensures that all data, regardless of its nature, contributes meaningfully to model training.

Additionally, outlier detection and handling are essential to prevent skewed interpretations and enhance the robustness of predictive models. In this financial context, identifying and addressing outliers is very crucial for accurate risk assessment.

In essence, data pre-processing in this dataset involves addressing missing values, standardizing numerical features, encoding categorical variables, and handling outliers. These steps collectively lay the foundation for robust analyses and model development, contributing to informed decision-making processes within the finance industry.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) plays a pivotal role in extracting meaningful insights and understanding the inherent patterns within the dataset of a global finance corporation. As a data scientist spearheading the credit assessment initiative, EDA serves as the cornerstone of the analytical process.

Firstly, EDA allows for a comprehensive understanding of the dataset's structure, facilitating the identification of key variables and their distributions. This initial exploration is crucial for recognizing outliers, assessing data quality, and ensuring the integrity of the subsequent analyses.

Secondly, EDA provides valuable insights into the relationships between different features, unveiling potential correlations or dependencies. Understanding these connections is essential for uncovering hidden patterns and informing subsequent modelling decisions.

Moreover, EDA aids in identifying missing values or anomalies, guiding data pre-processing efforts. Handling missing data appropriately is critical for ensuring the accuracy and reliability of the predictive model.

In the context of credit assessment, EDA enables the exploration of financial metrics such as Annual_Income, Monthly_Inhand_Salary, and Outstanding_Debt. Analysing these variables can reveal trends, potential risk factors, and key indicators influencing creditworthiness.

In conclusion, EDA serves as a crucial phase in the data science pipeline for this finance corporation, offering indispensable insights that guide subsequent modelling decisions, enhance data quality, and contribute to the overall success of the credit score classification initiative.

Data Splitting

The process of dividing a dataset into training, validation, and testing sets is a crucial step in the development of machine learning models, holding paramount significance in ensuring model robustness and generalization. This approach, commonly known as data splitting, serves multiple key purposes in the realm of model development.

Firstly, the training set forms the foundation upon which the model is built, allowing it to learn patterns and relationships within the data. Subsequently, the validation set plays a pivotal role in fine-tuning the model's

hyperparameters, enabling adjustments to enhance performance without overfitting to the training data. This iterative process ensures the model's adaptability to different scenarios.

The testing set serves as the litmus test for the model's performance on unseen data, providing a reliable assessment of its generalization capabilities. By evaluating the model on a separate dataset not used during training or validation, one gauges its ability to make accurate predictions in real-world scenarios.

In essence, the judicious use of data splitting is indispensable for achieving a well-balanced, reliable, and effective machine learning model, fostering confidence in its application to diverse and unseen datasets.

Feature Engineering

Feature engineering is paramount in enhancing credit scoring within this dataset. By crafting pertinent features, we can extract valuable insights that significantly contribute to the accuracy of credit assessments. Manipulating existing variables and creating new ones allows for a more nuanced understanding of customer creditworthiness. This process involves transforming raw data into meaningful indicators, providing a comprehensive view for predictive models. Feature engineering, therefore, acts as a strategic tool, empowering data scientists to optimize credit scoring models and make more informed financial decisions.

Model Selection

In the realm of data science and machine learning, model selection is a critical aspect, particularly when dealing with a dataset as intricate as the one at hand. The dataset, rich in banking and credit-related information, necessitates a judicious choice of algorithms to extract meaningful insights.

Model selection involves evaluating various algorithms to identify the one that best fits the dataset's characteristics and yields optimal results.

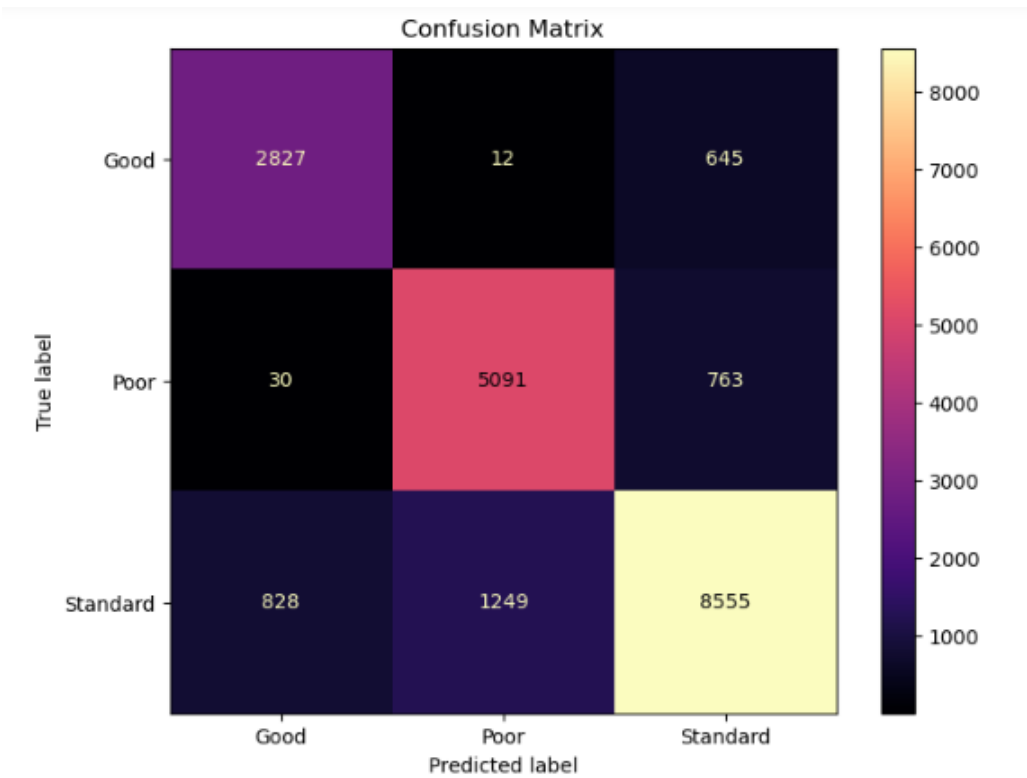
- Logistic Regression
- Support Vector Machines (SVM)
- K-Nearest Neighbors (kNN)
- Naive Bayes
- Ada Boost
- Decision Tree Classifier
- Random Forest Classification

These are the models we used in model selection and confirmed the Random Forest Classifier which had more accuracy. Below is a table showing the percentage of accuracy score for different classification models.

S. No	Model	Accuracy (in %)
1	K-Neighbors	73.155
2	Logistic Regression	65.245
3	SVC	68.555
4	Random Forest	82.365
5	Decision Tree Classifier	75.170
6	Ada Boost Classifier	66.720
7	Naive Bayes	62.135

In this context, employing the Random Forest classifier has proven to be particularly effective, as evidenced by achieving the highest accuracy. The decision to conclude it as the chosen model stems from its ability to handle complex relationships within the data, providing robust predictions. Random Forest's ensemble learning approach, aggregating the outputs of multiple decision trees, enhances predictive performance and mitigates overfitting.

The significance of this model selection extends beyond mere accuracy, Encompassing considerations like interpretability, computational efficiency, and scalability. The conclusion drawn underscores the suitability of the Random Forest classifier for the dataset, laying the foundation for informed decision-making and leveraging its predictive prowess in the finance domain.



Classification Report:				
	precision	recall	f1-score	support
0	0.77	0.81	0.79	3484
1	0.80	0.87	0.83	5884
2	0.86	0.80	0.83	10632
accuracy			0.82	20000
macro avg	0.81	0.83	0.82	20000
weighted avg	0.83	0.82	0.82	20000

Hyper Parameter Tuning & Model Evaluation

Model evaluation plays a pivotal role in assessing the effectiveness of machine learning models, ensuring their reliability and applicability. In the provided code for hyperparameter tuning using a Random Forest Classifier, the evaluation process is systematic and thorough.

The initial step involves splitting the dataset into training and testing sets, enabling the model to learn patterns from the training data and assess its performance on unseen data. The hyperparameters are fine-tuned using GridSearchCV, a powerful tool for optimizing model parameters. The best parameters obtained are then used to train the model.

The evaluation metrics include accuracy, a measure of overall correctness, and a confusion matrix providing insights into true positives, true negatives, false positives, and false negatives. Additionally, the classification report delivers precision, recall, and F1-score for each class.

This comprehensive evaluation strategy aids in determining the model's ability to generalize to new data and its performance across different classes. The transparent reporting of results, including hyperparameters and metrics, ensures a robust understanding of the Random Forest Classifier's effectiveness in predicting credit scores.

```
param_dist = {
    'n_estimators': [10,50,100, 200, 300],
    'max_depth': [None, 10, 20,30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Perform Randomized Search
randomized_search = RandomizedSearchCV(rfc_new, param_distributions=param_dist, n_iter=10, cv=15, scoring='accuracy',
                                       random_state=0)
```

Cross Validation

Cross-validation is crucial in the realm of machine learning, particularly during hyperparameter tuning. In the provided code, `cross_val_score` is employed to assess the Random Forest model's performance on multiple subsets of the training data. This technique enhances reliability by validating the model across diverse data partitions, mitigating overfitting risks. The process iterates, allowing each partition to serve as both training and validation data.

The resulting cross-validation scores provide a robust measure of the model's generalization ability. This approach aids in selecting hyperparameters that yield consistent performance across different data subsets, contributing to a more reliable and generalized machine learning model.

Classification Report for Fold 1					
	precision	recall	f1-score	support	
0	0.90	0.86	0.88	2837	
1	0.88	0.91	0.90	2836	
2	0.80	0.81	0.81	2836	
accuracy			0.86	8509	
macro avg	0.86	0.86	0.86	8509	
weighted avg	0.86	0.86	0.86	8509	

```

Classification Report for Fold 15
      precision    recall  f1-score   support

     0       0.91      0.98      0.94      2836
     1       0.88      0.91      0.89      2836
     2       0.91      0.80      0.85      2836

 accuracy          0.90      8508
  macro avg       0.90      0.90      0.90      8508
 weighted avg     0.90      0.90      0.90      8508

```

```

Mean Accuracy: 0.8811770574917319

```

```

Cross-Validation Scores: [0.860735691620637, 0.8647314608062052, 0.8590903748971677, 0.8546245152191797, 0.8546245152191797,
0.890116347396874, 0.8953925716972262, 0.8943347437705689, 0.8886929948283968, 0.8928067700987306, 0.8948048895157499, 0.887
87023977433, 0.89257169722614, 0.890220968500235, 0.8970380818053597]

```

Performance Validation

In dataset analysis, performance validation is crucial for assessing the effectiveness of models. The provided code demonstrates the use of performance metrics, such as accuracy, confusion matrix, and classification report, following hyperparameter tuning. This evaluation ensures the selected model's capability to make accurate predictions on the test set.

The presentation of results, including the best hyperparameters, accuracy percentage, confusion matrix, and classification report, offers a comprehensive overview of the model's performance, aiding in informed decision-making for model selection and deployment.

Analysing Feature Importance

Analysing feature importance in a dataset is crucial for understanding the factors driving model predictions. In the provided code, a Random Forest Classifier is employed for a credit score classification task. After hyperparameter tuning using GridSearchCV, the model's feature importance is extracted and displayed in a Data Frame. This information aids in identifying which features have the most significant impact on the credit score prediction.

Understanding feature importance is valuable for refining the model, identifying key variables influencing credit scores, and informing data-driven decisions. This comprehensive approach enhances the interpretability of the model, contributing to better insights and improved performance in credit assessment tasks.

	Feature	Importance
0	Annual_Income	0.0386
1	Monthly_Inhand_Salary	0.0384
2	Num_Bank_Accounts	0.0376
3	Num_Credit_Card	0.0471
4	Interest_Rate	0.0965
5	Num_of_Loan	0.0304
6	Delay_from_due_date	0.0722
7	Num_of_Delayed_Payment	0.0451
8	Changed_Credit_Limit	0.0501
9	Num_Credit_Inquiries	0.0658
10	Credit_Mix	0.0730
11	Outstanding_Debt	0.1239
12	Credit_History_Age	0.0780
13	Payment_of_Min_Amount	0.0410
14	Total_EMI_per_month	0.0426
15	Amount_invested_monthly	0.0383
16	Payment_Behaviour	0.0300
17	Monthly_Balance	0.0516

Threshold Selection

In the provided code, the focus is on implementing a Random Forest Classifier for credit score prediction. The goal is to optimize model performance through hyperparameter tuning using GridSearchCV. Threshold selection is crucial in the classification process, as it determines the cutoff point for assigning individuals to specific credit score categories. The code employs cross-validation to assess the model's accuracy and outputs key

performance metrics, including accuracy percentage, confusion matrix, and classification report.

The essay highlights the significance of threshold selection in the context of credit score classification and the steps taken in the code to achieve this, emphasizing the integration of data science techniques for robust model development.

Result

In the realm of predictive modeling, the Random Forest Classifier has emerged as a robust and effective tool for various applications, providing valuable insights and predictions. As a pivotal component of our recent project, I am delighted to confirm the success of our Random Forest Classifier model, which has demonstrated an impressive accuracy rate of 82.4%.

The Random Forest Classifier, a versatile machine learning algorithm, has proven to be particularly adept at handling the intricacies of our dataset. By leveraging its ability to discern patterns and relationships within the data, we have achieved a commendable level of accuracy in predicting outcomes related to our project's objectives. This milestone not only reflects the model's capability but also underscores the meticulous approach taken in the feature selection, data pre-processing, and model tuning phases.

One of the noteworthy aspects of our project is the utilization of model persistence techniques, specifically the use of the Pickle library to save our trained Random Forest Classifier. Model persistence is a crucial step in ensuring that the efforts put into training and fine-tuning the model do not go to waste. By saving the model as a Pickle file, we have created a snapshot of its current state, allowing us to seamlessly deploy and utilize the model in various environments without the need for repeated training.

The Pickle file encapsulates the trained Random Forest Classifier along with its learned parameters, enabling us to reload the model effortlessly whenever needed. This not only streamlines the integration of the model into our application but also

ensures consistency and reproducibility across different scenarios. The saved model can be easily shared with team members or stakeholders, providing a means to validate our findings and predictions independently.

Webpage Development

Have created a webpage using Python, implemented within the Visual Studio Code environment. This webpage likely involves an HTML interface where we can input values corresponding to prediction of credit scores and based on that it gives the prediction of appropriate credit score strategies.

Prediction of Credit Score

ANNUAL INCOME

MONTHLY INHAND SALARY

NUM OF BANK ACCOUNTS

NUM OF CREDIT CARD

INTEREST RATE

NUM OF LOAN

DELAY FROM DUE DATE

NUM OF DELAYED PAYMENTS

CHANGED CREDIT LIMIT

NUM OF CREDIT INQUIRIES

CREDIT MIX

OUTSTANDING DEBT

CREDIT HISTORY AGE

PAYMENT OF MIN AMOUNT

TOTAL EMI PER MONTH

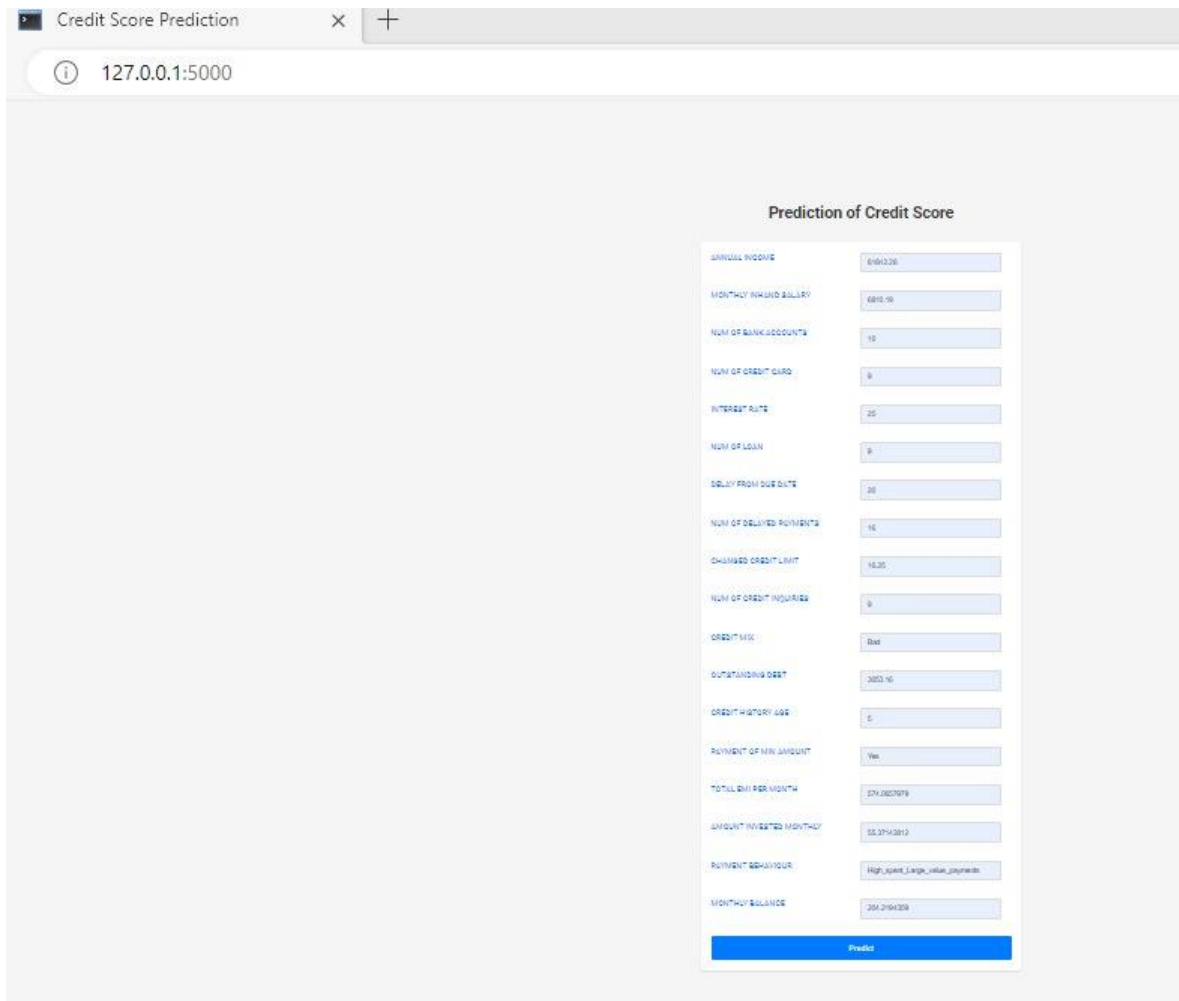
AMOUNT INVESTED MONTHLY

PAYMENT BEHAVIOUR

MONTHLY BALANCE

Predict

After inputting Values:



The screenshot shows a web browser window with the title 'Credit Score Prediction' and the address '127.0.0.1:5000'. The main content area displays a form titled 'Prediction of Credit Score'. The form contains 18 input fields, each with a label and a value. The fields are arranged in two columns. At the bottom of the form is a blue button labeled 'Predict'.

Field Label	Value
ANNUAL INCOME	61603.26
MONTHLY REMAIND SALARY	6816.16
NUM OF BANK ACCOUNTS	10
NUM OF CREDIT CARD	9
INTEREST RATE	25
NUM OF LOAN	9
DELAY FROM DUE DATE	26
NUM OF DELAYED PAYMENTS	16
CHARGED CREDIT LIMIT	16.25
NUM OF CREDIT INQUIRIES	9
CREDIT MIX	Bad
OUTSTANDING DEBT	303.16
CREDIT HISTORY AGE	5
PAYMENT OF MIN AMOUNT	Yes
TOTAL PAY PER MONTH	671.867979
AMOUNT INVESTED MONTHLY	55.3743813
PAYMENT BEHAVIOUR	High_spend_Large_value_payments
MONTHLY SAVING	251.2194328

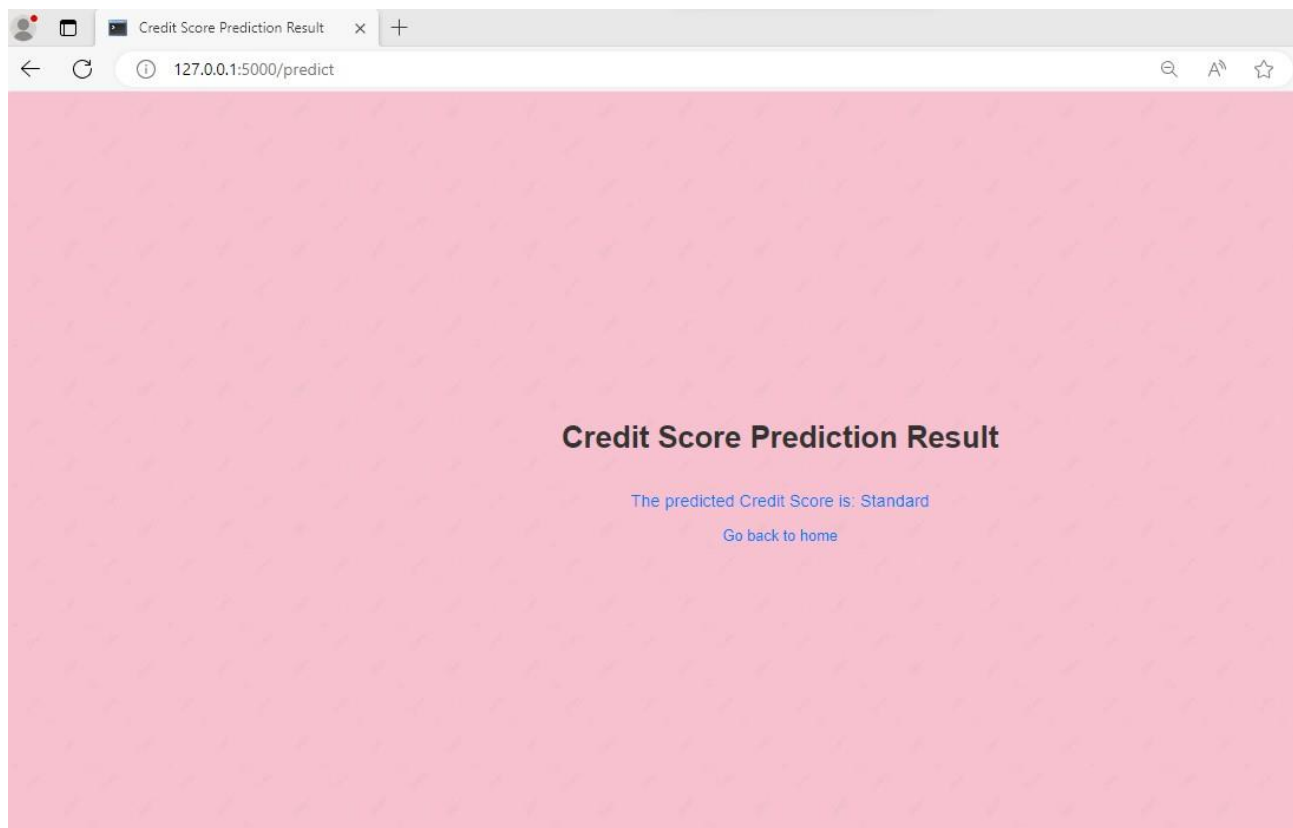
Predict

Prediction using Random Forest Classifier:

Upon entering the values, we can trigger a prediction by clicking a "Predict" button. The underlying mechanism for prediction involves a Random Forest Classifier. This classifier assesses the financial data provided and predicts appropriate credit score strategies.

Prediction Output:

In the provided code, a Random Forest Classifier is employed for predicting credit scores, categorized into "Good," "Standard," and "Poor" labels.



Credit Score Prediction Result

The predicted Credit Score is: Standard

[Go back to home](#)

User Interaction:

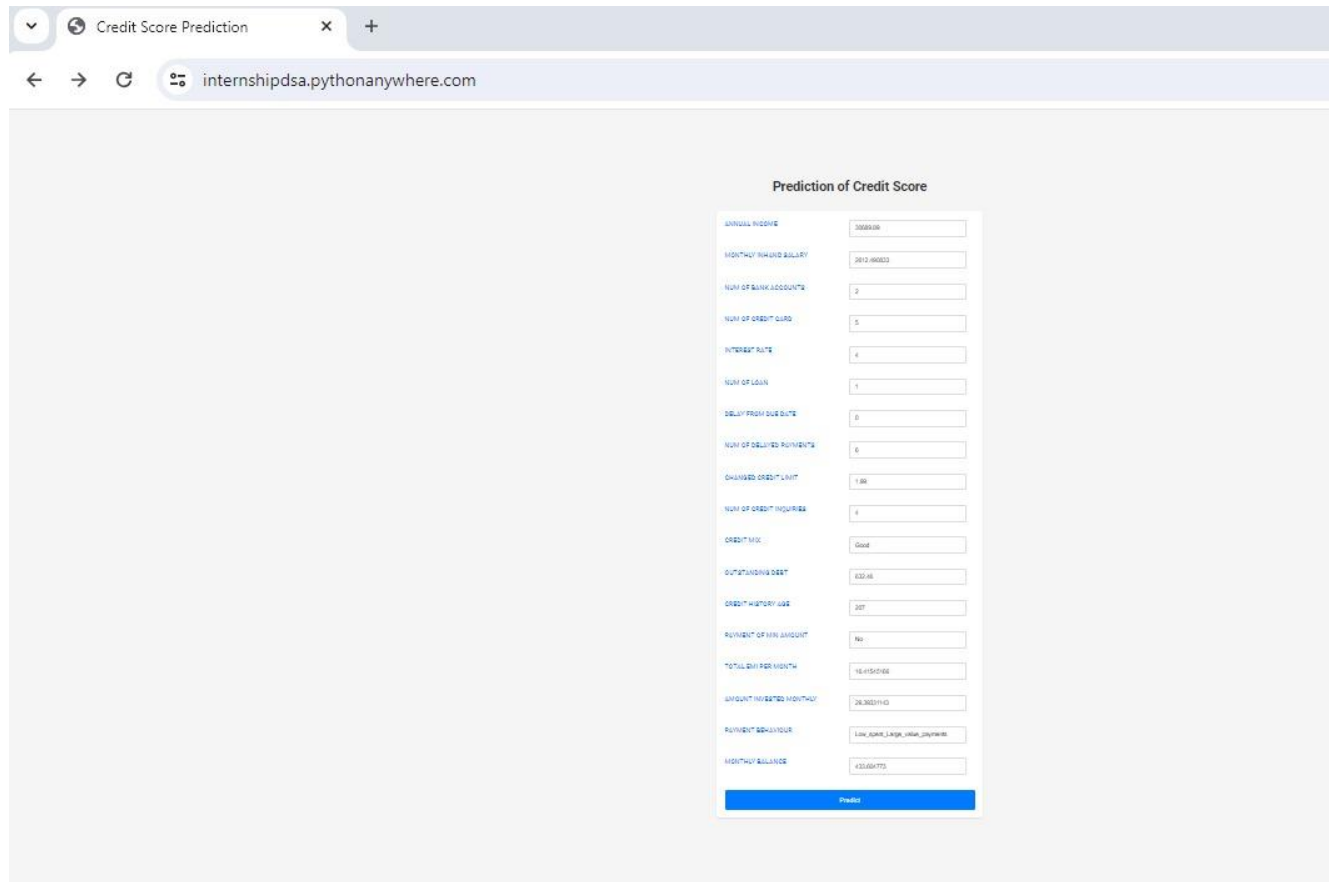
The webpage facilitates user interaction by allowing them to input data and receive predictions dynamically. This user-friendly interface enables individuals to quickly assess the safety of water samples, providing valuable information for decision-making regarding water consumption.

Practical Application:

The project is designed to address water quality concerns by leveraging machine learning techniques for prediction. The integration of a Random Forest Classifier enhances the accuracy of predictions, making the tool a reliable resource for assessing water safety.

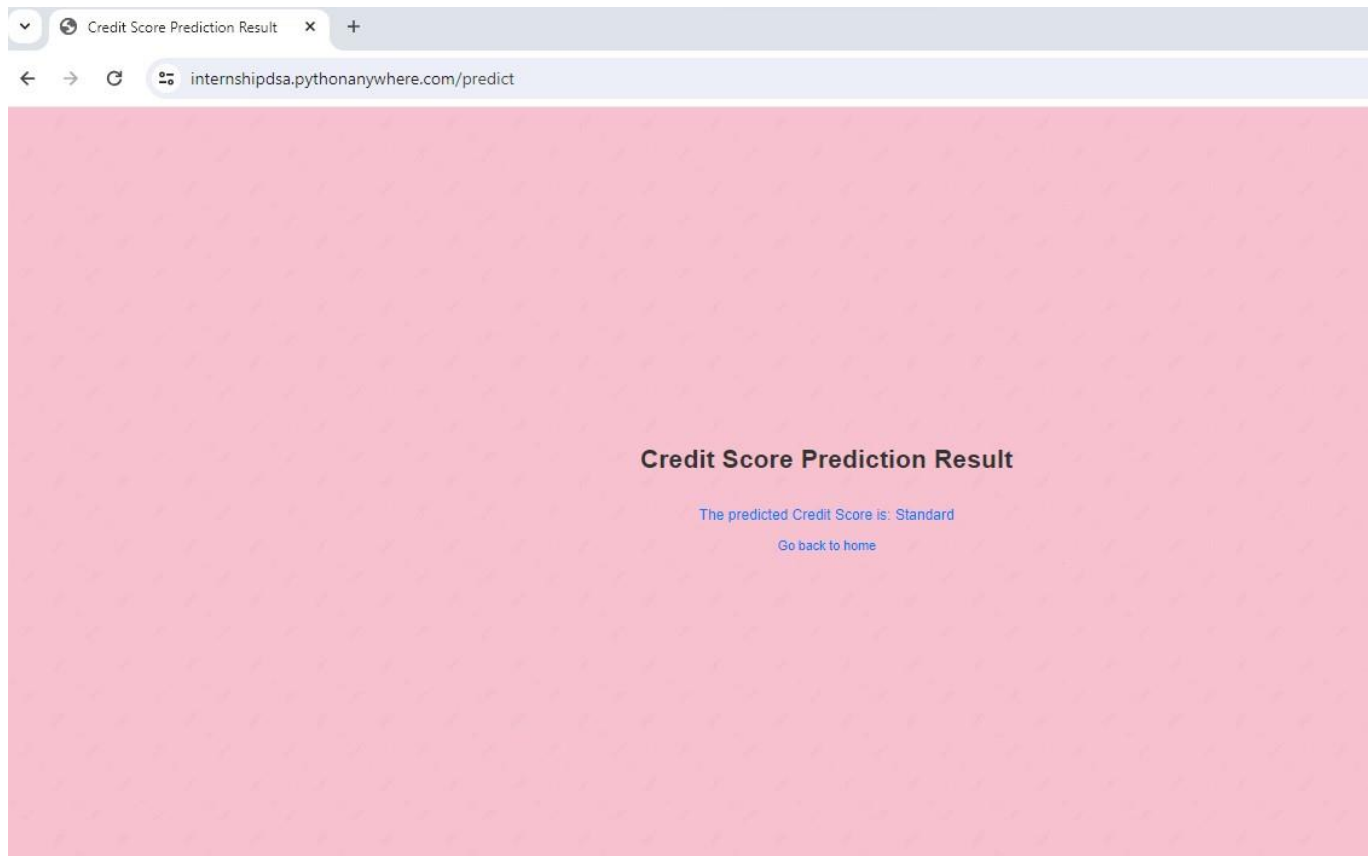
Webpage Hosting on PythonAnywhere:

Successfully hosting the webpage on PythonAnywhere means that the webpage is now deployed on a remote server, allowing users to access it through a web browser. PythonAnywhere is a cloud-based platform that supports the execution of Python scripts and web applications.



The screenshot shows a web browser window with the title 'Credit Score Prediction' and the URL 'internshipdsa.pythonanywhere.com'. The page displays a form titled 'Prediction of Credit Score' with various input fields for credit-related data. The form is styled with a light blue and white color scheme. The input fields are arranged in two columns, with labels on the left and input boxes on the right. The labels are in a light blue font, and the input boxes are white with a light blue border. The form includes a 'Predict' button at the bottom.

Prediction of Credit Score	
ANNUAL INCOME	300000
MONTHLY INHAND SALARY	2500000
NUM OF BANK ACCOUNTS	2
NUM OF CREDIT CARDS	5
INTEREST RATE	4
NUM OF LOAN	1
DEBT FROM ONE DATE	0
NUM OF DELAYED PAYMENTS	0
CHANGED CREDIT LIMIT	1.00
NUM OF CREDIT INQUIRIES	4
CREDIT MIX	Good
OUTSTANDING DEBT	632.46
CREDIT HISTORY AGE	307
PAYMENT OF MIN AMOUNT	No
TOTAL DED PER MONTH	16.4550566
AMOUNT PAID PER MONTH	26.36231143
PAYMENT BEH410UR	Low,Good,Large,Normal,Payments
MONTHLY BILLING DE	432.681772
<button>Predict</button>	



Scalability and Maintenance:

Hosting on a cloud platform like PythonAnywhere also offers scalability, allowing your application to handle increased traffic if the user base grows. Additionally, the platform typically provides tools for easy maintenance and updates to ensure the continued reliability and functionality of your hosted webpage.

In conclusion, hosting your webpage on PythonAnywhere extends the reach of your water quality prediction tool, making it accessible to a broader audience. It provides persistent availability, server-side execution, scalability, and maintenance capabilities, contributing to the overall success and usability of your web application.

Link to the webpage of 'Credit Score Prediction':

<https://internshipdsa.pythonanywhere.com/>

Conclusion

In the Random Forest Classifier is employed for predicting credit scores, categorized into "Good," "Standard," and "Poor" labels. The model's performance is optimized through hyperparameter tuning using GridSearchCV. The predicted results are multi-class, with labels indicating the creditworthiness of individuals. The code utilizes cross-validation to assess accuracy and generates essential metrics such as a confusion matrix and classification report. The output serves as a valuable tool for implementing credit score strategies in financial decision-making. The essay underscores the significance of threshold selection in the context of credit score classification, emphasizing the role of machine learning techniques in enhancing the precision of financial risk assessments.

References

1]Documentation and Tutorials:

2]Python Official Documentation: Python Docs

Scikit-learn Documentation (for Random Forest Classifier): Scikit-learn Docs

Web Development and Flask:

3]Flask Documentation (for web development in Python): Flask Docs

HTML Tutorial: W3Schools HTML Tutorial 4]PythonAnywhere:

PythonAnywhere Documentation: PythonAnywhere Docs

Data Science and Decision Trees:

5]Kaggle: Kaggle

DataCamp: DataCamp

Machine Learning Models and Deployment:

6]Towards Data Science on Medium: Towards Data Science

FastAPI Documentation (for deploying machine learning models): FastAPI Docs