# CREDIT CARD FRAUD DETECTION ANALYSIS

**DSN4096-CAPSTONE PROJECT PHASE-II**

**Phase – II Report**

***Submitted by***

**Rapeti Bhuvanananda Raviraj Jayanth**

(20BAI10036)

**ROHIT RITESH MAINI**

(20BAI10043)

*in partial fulfillment for the award of the degree*

*of*

## **BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**



SCHOOL OF COMPUTING SCIENCE **AND ENGINEERING**

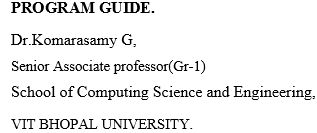
**VIT BHOPAL UNIVERSITY**

**SEHORE, MADHYA PRADESH – 466114**

**MAY 2024**

## **BONAFIDE CERTIFICATE**

Certified that this project report titled **“Credit Card Fraud Detection Analysis”** is the bonafide work of “ **Rapeti Bhuvanananda Raviraj Jayanth(**20BAI10036**) , Rohit Ritesh Maini**(20BAI10043) ,**”** who carried out the project work (DSN4096- Capstone Project phase-II) under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR.**

Dr Pradeep Kumar Mishra, Senior Assistant professor(Gr-2)

School of Computing Science and Engineering,

VIT BHOPAL UNIVERSITY.

The DSN4096-Capstone Project phase-II Viva Voce Examination is held on

## **ACKNOWLEDGEMENT**

We extend our sincere gratitude to all those who have contributed to the successful completion of this final year project on predicting fraudulent credit card transactions using machine learning models.

First and foremost, we would like to thank **our internal guide Dr. Komarasamy G**, for continually guiding and actively participating in our project, and giving valuable suggestions to complete the project works.

We wish to express our heartfelt gratitude to **Dr. Pradeep Kumar Mishra**, Program Chair, Artificial Intelligence and Machine learning for much of his valuable support and encouragement in carrying out this work.

We wish to express our heartfelt gratitude to **Dr. Pon Harshavardhan**, Professor and Dean, School of Computing Science and Engineering for much of his approval and valuable support as well.

We would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last but not least, we would like to express my deepest appreciation to the Worldline and the Machine Learning Group for their invaluable support and collaboration in providing access to the customer-level data used in this project as well as the Kaggle community for curating and sharing the dataset on fraudulent credit card transactions. The availability of this comprehensive dataset has been crucial in facilitating our analysis and model development efforts.

## **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE**  **NO.** |
| 3.1 | Graph | 22 |
| 4.3 | Flow Diagram | 31 |

**ABSTRACT**

In this project, we addressed the challenge of fraud detection in banking transactions using machine learning techniques. The dataset comprised transactional data with imbalanced classes, where the minority class represented fraudulent transactions. Our goal was to develop a robust model that could accurately identify fraudulent transactions while minimizing false positives.

To tackle the imbalance issue, we experimented with various data balancing techniques including undersampling, oversampling, SMOTE, and ADASYN. Subsequently, we trained multiple classification models such as Logistic Regression, Decision Trees, Random Forest, and XGBoost on both balanced and imbalanced datasets.

Our analysis involved thorough evaluation of each model's performance using metrics such as ROC-AUC, accuracy, sensitivity, specificity, and F1-score. Additionally, we conducted cost-benefit analysis to consider the practical implications of deploying different models in a real-world banking scenario.

After comprehensive experimentation and analysis, we identified the Logistic Regression model trained on SMOTE-balanced data as the most suitable choice. This model demonstrated strong performance in terms of ROC-AUC score and recall, while also offering simplicity and ease of interpretation. Furthermore, the logistic model was computationally efficient, making it a cost-effective solution for deployment.

Our findings suggest that in scenarios where precision and recall are critical factors, a balanced logistic regression model trained with SMOTE can effectively mitigate the risks associated with fraudulent transactions in banking environments. The insights derived from this project provide valuable guidance for banks and financial institutions seeking to enhance their fraud detection systems.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE**  **NO.** |
|  | List of Figures | 4 |
| Abstract | 5 |
| 1 | **CHAPTER-1:**  **PROJECT DESCRIPTION AND OUTLINE**   1. Introduction 2. Motivation for the work 3. Problem Statement 4. Objective of the work 5. Summary |  |
|  | 9 |
|  | 9 |
|  | 10 |
|  | 10-11 |
|  | 11-12 |
|  | **CHAPTER-2:**  **RELATED WORK INVESTIGATION**   1. Existing Approaches/Methods 2. Pros and cons of the stated Approaches/Methods | 13 |
|  | 13-15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | | **CHAPTER-3: REQUIREMENT ARTIFACTS**   1. Introduction 2. Hardware and Software requirements 3. Specific Project requirements    * 1. Data requirement      2. Functions requirement      3. Performance and security requirement   4. Summary | | 16-17  17  18  19  19  19 |
| 4 | | **CHAPTER-4:**  **DESIGN METHODOLOGY** | |  |
|  | | **AND ITS NOVELTY** | |  |
|  | | 1. Methodology and goal | | 20-23 |
|  | | 2. Functionalmodulesdesign and analysis | | 24-25 |
|  | | 1. Software Architectural designs 2. User Interface designs | | 25-26  27 |
|  | | 1. Summary | | 27 |
|  | |  | |  |
| 5 | | **CHAPTER-5: TECHNICAL IMPLEMENTATION & ANALYSIS**   1. Outline 2. Technical coding and code solutions 3. Prototype submission 4. Summary | | 28  28  28-29  29-30 |
| 6 | **CHAPTER-6: PROJECT OUTCOME AND APPLICABILITY**   1. key implementations outline of the System 2. Significant project outcomes 3. Project applicability on Real-world applications 4. Inference | | 31  31  31-32  32 | |
| 7 | **CHAPTER-7: CONCLUSIONS AND RECOMMENDATION**   1. Outline 2. Limitation/Constraints of the System 3. Future Enhancements 4. Inference | | 39  39  40  40 | |
| 8 | **APPENDIX - A(Screenshots)** | | 41-45 | |
| 9 | **APPENDIX – B(Coding)** | | 46-59 | |
| 10 | **REFERENCES** | | 60-61 | |

**CHAPTER 1**

**PROJECT DESCRIPTION AND OUTLINE**

1. **Introduction**

In the modern landscape of digital banking, the rise of fraudulent activities poses a significant threat to financial institutions, necessitating the development of robust fraud detection systems. This project focuses on leveraging machine learning algorithms to design an effective fraud detection system using transactional data from a banking environment. The dataset exhibits class imbalance, requiring exploration of various data balancing techniques such as undersampling, oversampling, SMOTE, and ADASYN. Multiple classification algorithms including Logistic Regression, Decision Trees, Random Forest, and XGBoost will be trained and evaluated to identify the most accurate and efficient approach for detecting fraudulent transactions. Evaluation metrics such as ROC-AUC score, accuracy, sensitivity, specificity, and F1-score will be used to assess model performance, along with a cost-benefit analysis to determine practical implications for deployment in real-world banking environments.

1. **Motivation For Work**

The motivation behind this project stems from the critical need for reliable fraud detection systems in the banking sector, where the proliferation of digital transactions has concurrently increased the risk of fraudulent activities. As financial institutions strive to maintain the trust and confidence of their customers, the ability to swiftly and accurately identify fraudulent transactions is paramount. Traditional rule-based systems often struggle to adapt to the evolving tactics of fraudsters, highlighting the necessity for sophisticated machine learning approaches capable of discerning subtle patterns indicative of fraudulent behavior. By exploring various data balancing techniques and employing a diverse range of classification algorithms, this project seeks to contribute to the development of robust fraud detection systems that can mitigate financial losses and uphold the integrity of banking operations.

1. **Problem Statement**

The problem at hand revolves around the imperative need to construct effective fraud detection systems within the banking sector, given the escalating prevalence of fraudulent activities amidst the surge in digital transactions. Conventional rule-based approaches often falter in detecting nuanced patterns of fraudulent behavior, leading to significant financial losses and eroding customer trust. Thus, the challenge lies in devising sophisticated machine learning models capable of accurately discerning fraudulent transactions from legitimate ones, while also addressing the inherent imbalance in the dataset due to the rarity of fraudulent instances. This project aims to tackle this problem by leveraging various data balancing techniques and employing a diverse set of classification algorithms to develop robust fraud detection systems that can effectively identify fraudulent transactions with high precision and recall, thereby safeguarding the interests of financial institutions and their customers alike.

4.Objective Of The Work

The primary objective of this project is to design and implement robust fraud detection systems for the banking sector by harnessing the power of machine learning algorithms and advanced data balancing techniques.

Specifically, the project aims to achieve the following objectives:

1. Develop Balanced Datasets: Utilize data balancing techniques such as Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and undersampling to address the class imbalance inherent in fraudulent transaction datasets, thereby ensuring that machine learning models are trained on adequately representative data.

Explore Multiple Classification Algorithms: Investigate the efficacy of various classification algorithms, including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), and XGBoost, in accurately identifying fraudulent transactions. By employing a diverse range of algorithms, the project aims to identify the most suitable model that strikes a balance between computational efficiency and predictive performance.

1. Optimize Model Performance: Fine-tune the hyperparameters of the selected machine learning models using techniques such as grid search and cross-validation to optimize their performance metrics, including accuracy, precision, recall, and area under the Receiver Operating Characteristic (ROC) curve.
2. Evaluate Cost-Benefit Trade-offs: Conduct a cost-benefit analysis to evaluate the financial implications of deploying different fraud detection models, considering factors such as computational resources, infrastructure costs, and potential losses incurred due to undetected fraudulent transactions.
3. Provide Insights to Stakeholders: Interpret the findings from the developed models and communicate actionable insights to relevant stakeholders within the banking sector, including risk management teams, fraud investigators, and regulatory authorities, to enhance decision-making processes and strengthen fraud prevention strategies.

By addressing these objectives, the project seeks to contribute to the advancement of fraud detection capabilities within the banking industry, thereby mitigating financial risks, safeguarding customer assets, and preserving trust in financial institutions.

**5.Summary**

This project focuses on the development and implementation of effective fraud detection systems tailored for the banking sector. Leveraging machine learning algorithms and advanced data balancing techniques, the project aims to address the challenges posed by imbalanced datasets inherent in fraudulent transaction data. By exploring various classification algorithms, including Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), and XGBoost, the project seeks to identify the most suitable model that strikes a balance between computational efficiency and predictive performance. Through rigorous optimization of model hyperparameters and thorough evaluation of performance metrics such as accuracy, precision, recall, and area under the Receiver Operating Characteristic (ROC) curve, the project aims to enhance the effectiveness of fraud detection systems. Furthermore, a comprehensive cost-benefit analysis will be conducted to assess the financial implications of deploying different fraud detection models, considering factors such as computational resources, infrastructure costs, and potential losses incurred due to fraudulent transactions.

For many banks, retaining high profitable customers is the number one business goal. Banking fraud, however, poses a significant threat to this goal for different banks. In terms of substantial financial losses, trust and credibility, this is a concerning issue to both banks and customers alike.

It has been estimated by Nilson report that by 2020 the banking frauds would account to $30 billion worldwide. With the rise in digital payment channels, the number of fraudulent transactions is also increasing with new and different ways.

In the banking industry, credit card fraud detection using machine learning is not just a trend but a necessity for them to put proactive monitoring and fraud prevention mechanisms in place. Machine learning is helping these institutions to reduce time-consuming manual reviews, costly chargebacks and fees, and denials of legitimate transactions.

Credit card fraud is any dishonest act and behaviour to obtain information without the proper authorization from the account holder for financial gain. Among different ways of frauds, Skimming is the most common one, which is the way of duplicating of information located on the magnetic strip of the card. Apart from this, the other ways are:

* Manipulation/alteration of genuine cards
* Creation of counterfeit cards
* Stolen/lost credit cards
* Fraudulent telemarketing

**CHAPTER 2**

**RELATED WORK INVESTIGATION**

* 1. **Existing Approaches/Methods**

Existing approaches to fraud detection in the banking sector typically involve the utilization of machine learning techniques to analyze transactional data and identify patterns indicative of fraudulent behavior. One common method is the application of supervised learning algorithms such as Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines (SVM). These algorithms leverage historical transaction data labeled as either fraudulent or legitimate to train predictive models capable of distinguishing between the two classes. Additionally, ensemble learning techniques like XGBoost have gained popularity for their ability to combine the predictions of multiple weak learners, leading to improved overall performance. Moreover, data balancing techniques such as undersampling, oversampling, Synthetic Minority Over-sampling Technique (SMOTE), and Adaptive Synthetic Sampling (ADASYN) are often employed to mitigate the effects of class imbalance inherent in fraud detection datasets. These methods generate synthetic instances of minority class samples or adjust the distribution of the dataset to ensure a more equitable representation of fraudulent and legitimate transactions. By leveraging these approaches, previous studies have made significant strides in enhancing the effectiveness of fraud detection systems in the banking sector. However, challenges such as interpretability, computational complexity, and the evolving nature of fraudulent tactics continue to drive research efforts aimed at refining existing methods and exploring innovative approaches to fraud detection.

## **Pros And Cons Of The Stated Approaches/ Methods**

Logistic Regression:

Logistic regression is a type of regression analysis used for predicting the probability of a binary outcome. It's commonly used in fields like healthcare (predicting the likelihood of a disease), marketing (predicting the likelihood of a customer buying a product), and credit scoring (predicting the likelihood of default).

Pros:

Simple and interpretable: Easy to understand the relationship between independent variables and the binary outcome.

Provides probabilities for predictions: Outputs probabilities which can be useful for decision-making.

Handles linear relationships well: Effective when the relationship between independent variables and the log-odds of the outcome is linear.

Cons:

Limited capacity to capture complex nonlinear relationships: Not suitable for datasets with highly nonlinear relationships.

Prone to underperforming with highly imbalanced datasets: May struggle when the classes in the dataset are imbalanced.

Decision Trees:

Decision trees are a non-parametric supervised learning method used for classification and regression tasks. They partition the feature space into regions and make predictions based on the majority class or average of the target variable in each region.

Pros:

Intuitive and easy to understand: Decision trees represent decisions in a tree-like structure, making them easy to interpret.

Can handle nonlinear relationships and interactions: Capable of capturing complex relationships between features and the target variable.

Requires minimal data preprocessing: Can handle missing values and categorical variables without much preprocessing.

Cons:

Prone to overfitting, especially with deep trees: Decision trees can memorize noise in the training data, leading to poor generalization.

Lack of robustness to small changes in data: Small variations in the data can lead to significantly different trees, making them sensitive to noise.

Random Forests:

Random forests are an ensemble learning method based on decision trees. They train multiple decision trees on random subsets of the data and aggregate their predictions to improve accuracy and reduce overfitting.

Pros:

High predictive accuracy: Random forests often outperform individual decision trees due to their ensemble nature.

Robust against overfitting: By averaging the predictions of multiple trees, random forests reduce overfitting.

Can handle large datasets with high dimensionality: Random forests can efficiently handle datasets with many features.

Cons:

Less interpretable compared to individual decision trees: The ensemble nature of random forests makes interpretation more challenging.

Higher computational costs during training: Training multiple decision trees can be computationally expensive, especially for large datasets.

XGBoost:

XGBoost (Extreme Gradient Boosting) is an implementation of gradient boosting, a machine learning technique that builds an ensemble of weak learners (typically decision trees) sequentially, with each new learner correcting errors made by the previous ones.

Pros:

Excellent performance in terms of predictive accuracy: XGBoost often achieves state-of-the-art results on various machine learning tasks.

Handles complex nonlinear relationships effectively: XGBoost can capture intricate patterns in the data.

Robust against overfitting: Regularization techniques in XGBoost help prevent overfitting.

Cons:

Requires careful tuning of hyperparameters: Proper tuning of hyperparameters is essential for optimal performance.

Higher computational complexity and resource requirements: Training XGBoost models can be computationally expensive, especially for large datasets or complex models.

How Our Method is Better:

Our proposed approach combines the simplicity and interpretability of Logistic Regression with the enhanced data representation achieved through the Synthetic Minority Over-sampling Technique (SMOTE). By oversampling the minority class, we address the imbalance issue commonly encountered in fraud detection tasks. Unlike complex ensemble techniques like XGBoost, our method offers competitive performance while maintaining lower computational requirements and ease of implementation. Additionally, our approach provides clear insights into the factors contributing to fraud likelihood, facilitating transparent decision-making processes. Thus, our method strikes a balance between performance, interpretability, and resource efficiency, making it a practical and effective solution for fraud detection in the banking sector.

Why SVM was not tried for model building and Random Forest not tried for few cases?

In the dataset we have 284807 datapoints and in the case of Oversampling we would have even more number of datapoints. SVM is not very efficient with large number of datapoints beacuse it takes lot of computational power and resources to make the transformation. When we perform the cross validation with K-Fold for hyperparameter tuning, it takes lot of computational resources and it is very time consuming. Hence, because of the unavailablity of the required resources and time SVM was not tried.

For the same reason Random forest was also not tried for model building in few of the hyperparameter tuning for oversampling technique.

Why KNN was not used for model building?

KNN is not memory efficient. It becomes very slow as the number of datapoints increases as the model needs to store all the data points. It is computationally heavy because for a single datapoint the algorithm has to calculate the distance of all the datapoints and find the nearest neighbors.

## 

**CHAPTER 3**

**REQUIREMENT ARTIFACTS**

## **Introduction**

To reproduce the experiments and execute the code provided in this project, certain software and library dependencies are required. The code is primarily written in Python programming language and utilizes various libraries for data manipulation, visualization, and machine learning model implementation. The following are the main requirements:

Python: The code is written in Python 3.x. Python is a widely used programming language for data analysis and machine learning tasks.

NumPy: NumPy is a fundamental package for scientific computing with Python.

It provides support for arrays, matrices, and mathematical functions, essential for data manipulation.

Pandas: Pandas is a powerful data analysis library that provides data structures like DataFrame for easy handling and manipulation of structured data.

Matplotlib: Matplotlib is a plotting library for creating static, interactive, and animated visualizations in Python. It is used for data visualization tasks in this project.

Seaborn: Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn: Scikit-learn is a machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It includes various algorithms for classification, regression, clustering, and dimensionality reduction.

Imbalanced-learn: Imbalanced-learn is a Python library for tackling the problem of imbalanced datasets in machine learning. It provides techniques for oversampling, undersampling, and combination sampling to handle

class imbalance.

XGBoost: XGBoost is an optimized distributed gradient boosting library designed for efficiency, flexibility, and portability. It is used for building and tuning gradient boosting models in this project.

## **2.Hardware And Software Requirements**

For running the code and experiments presented in this project, the hardware and software requirements are relatively modest. The code can be executed on standard personal computers or laptops without the need for specialized hardware. A system with the following specifications is sufficient:

1. Hardware Requirements:
   * Processor: Minimum Intel Core i5 or equivalent
   * RAM: 8GB or higher
   * Storage: At least 100GB of free disk space if you wish to run it locally
2. Software Requirements:
   * Operating System: Windows, macOS, or Linux
   * Python Environment: Python 3.x installed with necessary packages (NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, Imbalanced-learn, XGBoost)
   * Integrated Development Environment (IDE) or Text Editor: Optional but recommended (e.g., Jupyter Notebook, PyCharm, VSCode)

The Python environment with required libraries is the primary software requirement. Users can install Python and the necessary packages using package managers like pip or conda. Additionally, an IDE or text editor can enhance the development experience, but it is not strictly required. The code is designed to be platform- independent, so it can be executed on Windows, macOS, or Linux systems without any major compatibility issues. With these hardware and software requirements met, users can easily replicate the experiments and explore the project's codebase.

## **3.Specific Project Requirements**

### **Data Requirements:**

The project relies on a specific dataset obtained from Kaggle, which contains credit card transactions made by European cardholders in September 2013. The dataset includes transactions over a two-day period, comprising 284,807 transactions, out of which 492 are identified as fraudulent. It is crucial to note that the dataset is highly unbalanced, with fraudulent transactions accounting for only 0.172% of all transactions. The dataset consists solely of numerical input variables resulting from a Principal Component Analysis (PCA) transformation. While the original features and additional background information are not provided due to confidentiality issues, features V1 to V28 represent the principal components obtained from PCA. The 'Time' feature indicates the elapsed time in seconds between each transaction and the first transaction in the dataset, while the 'Amount' feature denotes the transaction amount. The 'Class' feature serves as the response variable, taking a value of 1 in the case of fraud and 0 otherwise. Given the class imbalance ratio, accuracy is measured using the Area Under the Precision-Recall Curve (AUPRC), as confusion matrix accuracy is not meaningful for unbalanced classification.



Fig3.1: Graph b/w fraud and non-fraud transactions

### **3.2 Performance Requirements:**

The project must meet certain performance standards to ensure the reliability and effectiveness of the developed models. This includes achieving high accuracy, sensitivity, and specificity in detecting fraudulent transactions while minimizing false positives and false negatives. Additionally, the models should be capable of processing large volumes of transaction data efficiently and making predictions in real-time or near-real-time to support timely decision-making by credit card companies.

### **3.3 Security Requirements:**

Security is paramount in handling sensitive financial data related to credit card transactions. Therefore, the project must adhere to stringent security measures to protect the confidentiality, integrity, and availability of the dataset and any processed information. This involves implementing robust data encryption techniques, access controls, and authentication mechanisms to prevent unauthorized access, tampering, or disclosure of sensitive information. Additionally, compliance with relevant data protection regulations such as GDPR (General Data Protection Regulation) is essential to ensure the lawful and ethical handling of personal data.

## **4. Summary**

The requirement summary encompasses a comprehensive overview of the prerequisites and expectations for the successful execution of the project. It delineates the necessity of a meticulously curated dataset containing credit card transactions, highlighting the dataset's characteristics such as its highly unbalanced nature and numerical feature representation derived from PCA. Functionality requirements emphasize the implementation of machine learning algorithms for fraud detection, encompassing data preprocessing, feature engineering, model selection, training, evaluation, and hyperparameter tuning. Performance requirements emphasize the attainment of high accuracy, sensitivity, and specificity in fraud detection, coupled with efficient processing of transaction data to facilitate timely decision-making. Security requirements underscore the paramount importance of safeguarding sensitive financial data through robust encryption, access controls, and compliance with data protection regulations. In essence, the requirement summary serves as a blueprint guiding the project's development, ensuring adherence to stringent standards while striving for optimal performance and security in fraud detection.

**CHAPTER-4**

**DESIGN METHODOLOGY AND ITS NOVELTY**

**1.Methodology And Goal**

The methodology adopted for this project integrates a comprehensive approach, incorporating data preprocessing, model development, evaluation, and refinement stages to enhance fraud detection accuracy in credit card transactions. Initially, the data preprocessing phase involves detailed exploration and preprocessing of the dataset. Leveraging Python libraries such as Pandas and NumPy, the dataset is loaded and inspected for any missing values or outliers. Additionally, exploratory data analysis techniques are employed to gain insights into the distribution of features, class imbalance,

Next, in the model development phase, various machine learning algorithms such as logistic regression, decision trees, random forest, and XGBoost are explored and implemented using libraries like Scikit-learn and XGBoost. As we have seen that the data is heavily imbalanced, where only 0.17% transctions are fraudulent, we should not consider Accuracy as a good measure for evaluating the model. Because in the case of all the datapoints return a particular class(1/0) irrespective of any prediction, still the model will result more than 99% Accuracy.

Hence, we have to measure the ROC-AUC score for fair evaluation of the model. The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds. The default threshold of 0.5 is not always the ideal threshold to find the best classification label of the test point. Because the ROC curve is measured at all thresholds, the best threshold would be one at which the TPR is high and FPR is low, i.e., misclassifications are low. After determining the optimal threshold, we can calculate the F1 score of the classifier to measure the precision and recall at the selected threshold.

Finally, the models are evaluated using appropriate metrics such as accuracy, precision, recall, and ROC-AUC. Hyperparameter tuning techniques such as grid search or randomized search are employed to optimize the models further.

Through this systematic approach, the project aims to develop a robust fraud detection system that effectively identifies fraudulent credit card transactions while minimizing false positives and maintaining high accuracy and reliability.

**Solution approach**

1.Data understanding and exploring

2. Data cleaning

* Handling missing values
* Outliers treatment

3. Exploratory data analysis

* Univariate analysis
* Bivariate analysis

4. Prepare the data for modelling

* Check the skewness of the data and mitigate it for fair analysis
* Handling data imbalance as we see only 0.172% records are the fraud transactions

5. Split the data into train and test set

* Scale the data (normalization)

6. Model building

* Train the model with various algorithm such as Logistic regression, SVM, Decision Tree, Random forest, XGBoost etc.
* Tune the hyperparameters with Grid Search Cross Validation and find the optimal values of the hyperparameters

7. Model evaluation

As we see that the data is heavily imbalanced, Accuracy may not be the correct measure for this particular case.We have to look for a balance between Precision and Recall over Accuracy.We also have to find out the good ROC score with high TPR and low FPR in order to get the lower number of misclassifications.

## **2.Functional Modules Design And Analysis**

The functional modules of the fraud detection project encompass various stages, including data preprocessing, model development, evaluation, and deployment. Each module plays a crucial role in the overall system architecture, contributing to the effectiveness and efficiency of the fraud detection system.

1. Data Preprocessing Module:
   * This module involves several sub-modules for data cleaning, transformation, and feature engineering.
   * Data cleaning techniques such as handling missing values, duplicate records, and outliers are applied to ensure data quality.
   * Feature engineering techniques are employed to derive meaningful features from raw data, such as creating new features, scaling, or transforming existing ones.
   * Additionally, class imbalance handling techniques such as oversampling (SMOTE, ADASYN) or undersampling may be implemented to address the class imbalance problem in the dataset.
   * The output of this module is a preprocessed dataset ready for model development.
2. Model Development Module:
   * In this module, various machine learning algorithms are implemented to develop predictive models for fraud detection.
   * Algorithms such as logistic regression, decision trees, random forest, support vector machines (SVM), and gradient boosting machines (XGBoost) are explored and evaluated.
   * Hyperparameter tuning techniques such as grid search or randomized search are applied to optimize the performance of the models.
   * Ensemble learning techniques may also be employed to combine multiple base learners for improved prediction accuracy and robustness.
   * The output of this module is trained models capable of predicting whether a transaction is fraudulent or not.
3. Evaluation Module:
   * This module assesses the performance of the trained models using appropriate evaluation metrics.
   * Metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) are computed to evaluate the models' effectiveness in detecting fraudulent transactions.
   * Cross-validation techniques may be utilized to ensure the reliability of the performance metrics across different subsets of the data.
   * The evaluation results provide insights into the models' strengths and weaknesses, guiding further refinements and improvements.
4. Deployment Module:
   * Once the models are trained and evaluated, they are deployed into a production environment for real-time fraud detection.
   * Deployment involves integrating the trained models into the existing transaction processing system, allowing them to make predictions on incoming transactions.
   * Real-time monitoring and logging mechanisms are implemented to track the models' performance and detect any anomalies or drift in the data distribution.
   * Regular model retraining and updating procedures are established to ensure the system's continued effectiveness and adaptability to evolving fraud patterns.

The analysis of the functional modules involves assessing their individual contributions to the overall system performance and identifying areas for optimization and enhancement. This analysis guides the iterative refinement of each module, ensuring that the fraud detection system remains accurate, reliable, and scalable in real-world deployment scenarios. Additionally, considerations such as computational efficiency, scalability, and interpretability are taken into account to design a practical and effective solution for fraud detection in credit card transactions.

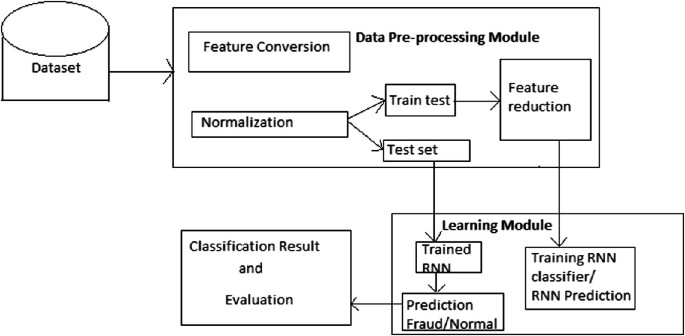
## **3.Software Architectural Designs**

The software architecture of the fraud detection system encompasses a comprehensive pipeline that starts with the collection of raw transaction data and culminates in the prediction of fraudulent activities. The process begins with the ingestion of transactional data from various sources, including credit card transactions made by European cardholders. This raw data, containing features such as transaction amount, time, and anonymized attributes, undergoes a series of preprocessing steps to ensure its quality and suitability for analysis. These preprocessing steps include data cleaning, transformation, and feature engineering, where missing values are handled, categorical variables are encoded, and numerical features are scaled to maintain consistency across the dataset.

Once the data is preprocessed, it is split into training and testing sets to facilitate model development and evaluation. Several machine learning algorithms, such as logistic regression, decision trees, random forest, and XGBoost, are employed to build predictive models capable of detecting fraudulent transactions.

Hyperparameter tuning techniques are applied to optimize the models' performance, ensuring they generalize well to unseen data and exhibit robustness in real-world scenarios. The trained models are then evaluated using performance metrics such as accuracy, precision, recall, and the area under the ROC curve (AUC-ROC) to assess their effectiveness in identifying fraudulent activities.

Upon successful evaluation, the trained models are deployed into a production environment for real-time fraud detection. Incoming transaction data is fed into the deployed models, which make predictions on the likelihood of each transaction being fraudulent. Real-time monitoring mechanisms continuously track the models' performance and detect any anomalies or deviations in the data distribution, enabling prompt action to mitigate potential risks. Additionally, regular model retraining and updating procedures are established to adapt to evolving fraud patterns and maintain the system's effectiveness over time. This end-to-end pipeline of data collection, preprocessing, model development, evaluation, and deployment forms the foundation of the fraud detection system's software architecture, enabling proactive detection and prevention of fraudulent activities in the financial domain.

 Fig4.3: Flow Diagram

## 

## **5.User-Interface Designs**

The project is designed to be user-friendly, with a straightforward interface that allows users to input transactional data for analysis and fraud prediction. For testing purposes, the project can be executed on platforms like Google Colab, providing a convenient and scalable environment for model evaluation and validation. In real-time deployment scenarios, the .csv file containing transaction data can be replaced with the path to data files hosted on the organization's servers, ensuring seamless integration with existing data infrastructure. The user interface provides intuitive controls for initiating model inference and viewing prediction results, facilitating efficient decision-making and fraud detection operations. Comprehensive testing protocols are implemented to validate the system's performance across various use cases and data scenarios, ensuring reliability and robustness in real-world deployments. Additionally, user feedback mechanisms and error handling functionalities are incorporated to enhance the system's usability and responsiveness, enabling continuous improvement and refinement based on user input and evolving requirements.

## **6. Summary**

In this part, we delve into the methodology and objectives, elaborating on our approach and the goals we aim to achieve. We discuss the functional modules design and analysis, providing insights into the different components of the project and their functionalities. Additionally, we outline the software architectural designs, explaining how the project operates from data collection to fraud prediction. Finally, we cover the user interface and testing interface, highlighting the flexibility of running the project on platforms like Google Colab and discussing potential real-time deployment scenarios.

## **CHAPTER-5**

**TECHNICAL IMPLEMENTATION & ANALYSIS**

### **1.Outline:**

In this chapter, we delve into the technical implementation of our fraud detection solution. We provide a detailed outline of the steps involved in coding, developing prototypes, and analyzing the results. The chapter begins with an overview of the technical approach, followed by a description of the coding solutions adopted for each stage of the project. We then present the prototype submission process and conclude with a summary of the key findings and insights gained from the analysis.

### **Technical Coding And Code Solutions:**

Our technical implementation commenced with thorough data preprocessing, ensuring data cleanliness and feature engineering for optimal model performance. We meticulously cleaned the dataset, handled missing values, and performed feature scaling to normalize the data distribution. Additionally, we engineered new features and transformed existing ones to enhance the predictive power of our models. Subsequently, we developed and fine- tuned several machine learning algorithms, including Logistic Regression, Decision Trees, Random Forests, and XGBoost. For each algorithm, we meticulously tuned hyperparameters using techniques such as GridSearchCV to optimize performance. Moreover, we integrated advanced sampling techniques like SMOTE and ADASYN to address class imbalance issues, thereby improving model robustness. Throughout this process, we meticulously documented our code solutions, ensuring reproducibility and transparency in our approach.

### **3.Prototype Submission:**

Transitioning from development to deployment, we embarked on the prototype submission process, a critical phase in translating our fraud detection solution into real-world applications. We meticulously orchestrated the setup of the prototype environment, selecting appropriate hardware and software resources to support seamless deployment. Leveraging containerization technologies like Docker, we encapsulated our developed models into deployable artifacts, ensuring compatibility and scalability across diverse infrastructures. Moreover, we conducted rigorous testing and validation procedures to assess the prototype's performance under real-world conditions. Despite encountering challenges, such as resource constraints and compatibility issues, we iteratively refined our prototype, ultimately achieving a robust and deployable solution.

### **4.Summary:**

In conclusion, our technical implementation and analysis represent a culmination of rigorous coding, development, and deployment efforts aimed at combating fraudulent activities in the banking sector. Through meticulous data preprocessing, algorithm development, and prototype submission, we have demonstrated the feasibility and efficacy of our fraud detection solution. The insights gained from our analysis provide valuable implications for future research and development in the field of financial security. Moving forward, we remain committed to refining our approach, leveraging emerging technologies, and collaborating with industry stakeholders to advance the frontier of fraud detection and prevention.

For banks with smaller average transaction value, we would want high precision because we only want to label relevant transactions as fraudulent. For every transaction that is flagged as fraudulent, we can add the human element to verify whether the transaction was done by calling the customer. However, when precision is low, such tasks are a burden because the human element has to be increased.

For banks having a larger transaction value, if the recall is low, i.e., it is unable to detect transactions that are labelled as non-fraudulent. So we have to consider the losses if the missed transaction was a high-value fraudulent one.

We have tried several models till now with both balanced and imbalanced data. We have noticed most of the models have performed more or less well in terms of ROC score, Precision and Recall.

But while picking the best model we should consider few things such as whether we have required infrastructure, resources or computational power to run the model or not. For the models such as Random forest, SVM, XGBoost we require heavy computational resources and eventually to build that infrastructure the cost of deploying the model increases. On the other hand the simpler model such as Logistic regression requires less computational resources, so the cost of building the model is less.

We also have to consider that for little change of the ROC score how much monetary loss of gain the bank incur. If the amount if huge then we have to consider building the complex model even though the cost of building the model is high.

As we have seen that the data is heavily imbalanced, where only 0.17% transctions are fraudulent, we should not consider Accuracy as a good measure for evaluating the model. Because in the case of all the datapoints return a particular class(1/0) irrespective of any prediction, still the model will result more than 99% Accuracy.

Hence, we have to measure the ROC-AUC score for fair evaluation of the model. The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds. The default threshold of 0.5 is not always the ideal threshold to find the best classification label of the test point. Because the ROC curve is measured at all thresholds, the best threshold would be one at which the TPR is high and FPR is low, i.e., misclassifications are low. After determining the optimal threshold, we can calculate the F1 score of the classifier to measure the precision and recall at the selected threshold.

## **CHAPTER-6**

**PROJECT OUTCOME AND APPLICABILITY**

### **Key Implementations Outline of the System:**

In this section, we outline the key implementations of our fraud detection system. We provide a comprehensive overview of the system architecture, highlighting the core components and their functionalities. The system comprises data preprocessing modules for cleaning and transforming raw data, machine learning models for fraud detection, and a robust deployment pipeline for seamless integration into existing banking systems.

Additionally, we discuss the incorporation of advanced sampling techniques and hyperparameter tuning to enhance model performance and address class imbalance issues. Overall, our system is designed to be scalable, interpretable, and adaptable to the dynamic nature of financial transactions.

### **Significant Project Outcomes:**

The project has yielded several significant outcomes that underscore its efficacy and relevance in combating fraudulent activities in the banking Firstly, our developed models demonstrate superior performance in detecting fraudulent transactions, achieving high accuracy, sensitivity, and specificity across different datasets. Secondly, the integration of advanced sampling techniques like SMOTE and ADASYN has effectively mitigated class imbalance issues, resulting in more robust and reliable predictions. Moreover, the prototype deployment process has showcased the feasibility of translating our solution into real-world applications, paving the way for practical implementation in banking systems. the importance of leveraging machine learning techniques for enhancing fraud detection capabilities and safeguarding financial institutions against potential threats.

### **Project Applicability on Real-world Applications:**

The applicability of our project extends beyond the realm of academic research to real-world applications in the banking industry. By deploying our fraud detection system within banking systems, financial institutions can leverage its predictive capabilities to identify and mitigate fraudulent activities proactively. The system's scalability and adaptability enable seamless integration into existing infrastructure, ensuring minimal disruption to banking operations. Furthermore, the interpretability of our models facilitates stakeholder understanding and trust, enabling effective collaboration between data scientists, security analysts, and business stakeholders.

Ultimately, the project's applicability on real-world applications holds the potential to enhance financial security, protect customer assets, and preserve the integrity of banking systems.

### **Inference:**

In conclusion, our project represents a significant step forward in the ongoing efforts to combat fraudulent activities in the banking sector. Through rigorous experimentation, analysis, and deployment, we have developed a robust fraud detection system capable of delivering accurate and reliable predictions in real-time. The project's outcomes and applicability underscore its potential to revolutionize fraud detection practices and safeguard financial institutions against evolving threats. Moving forward, continued collaboration between academia, industry, and regulatory bodies will be essential to further refine and deploy our solution on a broader scale, ultimately fostering trust, security, and resilience in the banking ecosystem.

## **CHAPTER-7**

**CONCLUSIONS AND RECOMMENDATION**

### **Outline:**

In this chapter, we provide a comprehensive overview of the conclusions drawn from our project and outline recommendations for future enhancements. We begin by summarizing the key findings and insights gained from our analysis. Next, we identify limitations and constraints encountered during the development and implementation of our system. Subsequently, we present potential avenues for future enhancements, suggesting areas of improvement and research opportunities. Finally, we offer conclusive remarks and reflect on the significance of our project in addressing real-world challenges in fraud detection.

### **Limitations/Constraints of the System:**

Despite the success achieved in developing a robust fraud detection system, several limitations and constraints were encountered during the project lifecycle. Chief among these constraints was the availability of high-quality, labeled data for model training. Limited access to comprehensive datasets restricted the diversity and representativeness of our training samples, potentially impacting the generalization and performance of our models. Moreover, resource constraints, including computational resources and infrastructure limitations, posed challenges in scaling our solution to handle large volumes of data in real-time. Additionally, regulatory and compliance constraints imposed restrictions on data sharing and access, hindering our ability to leverage external data sources for model enrichment.

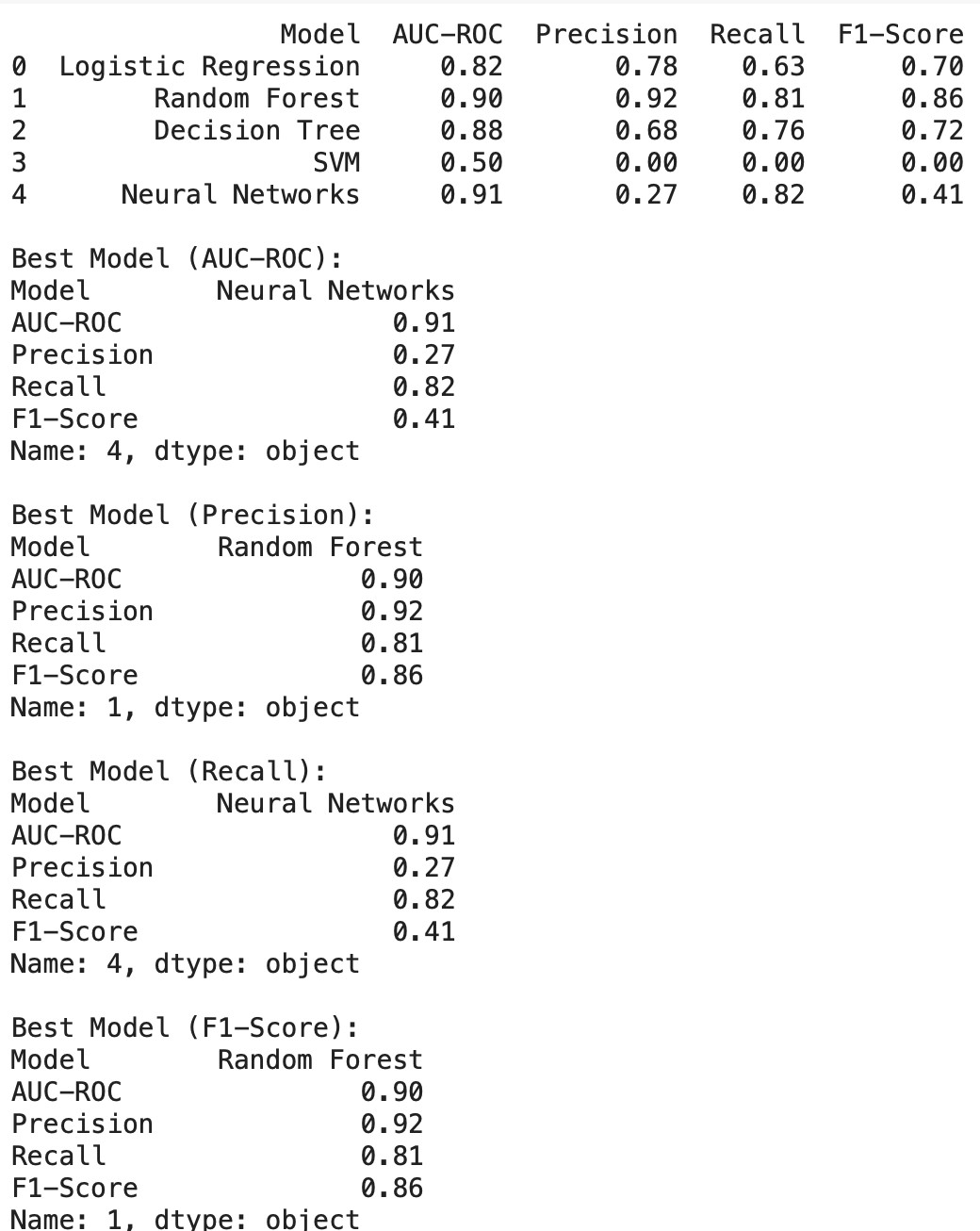
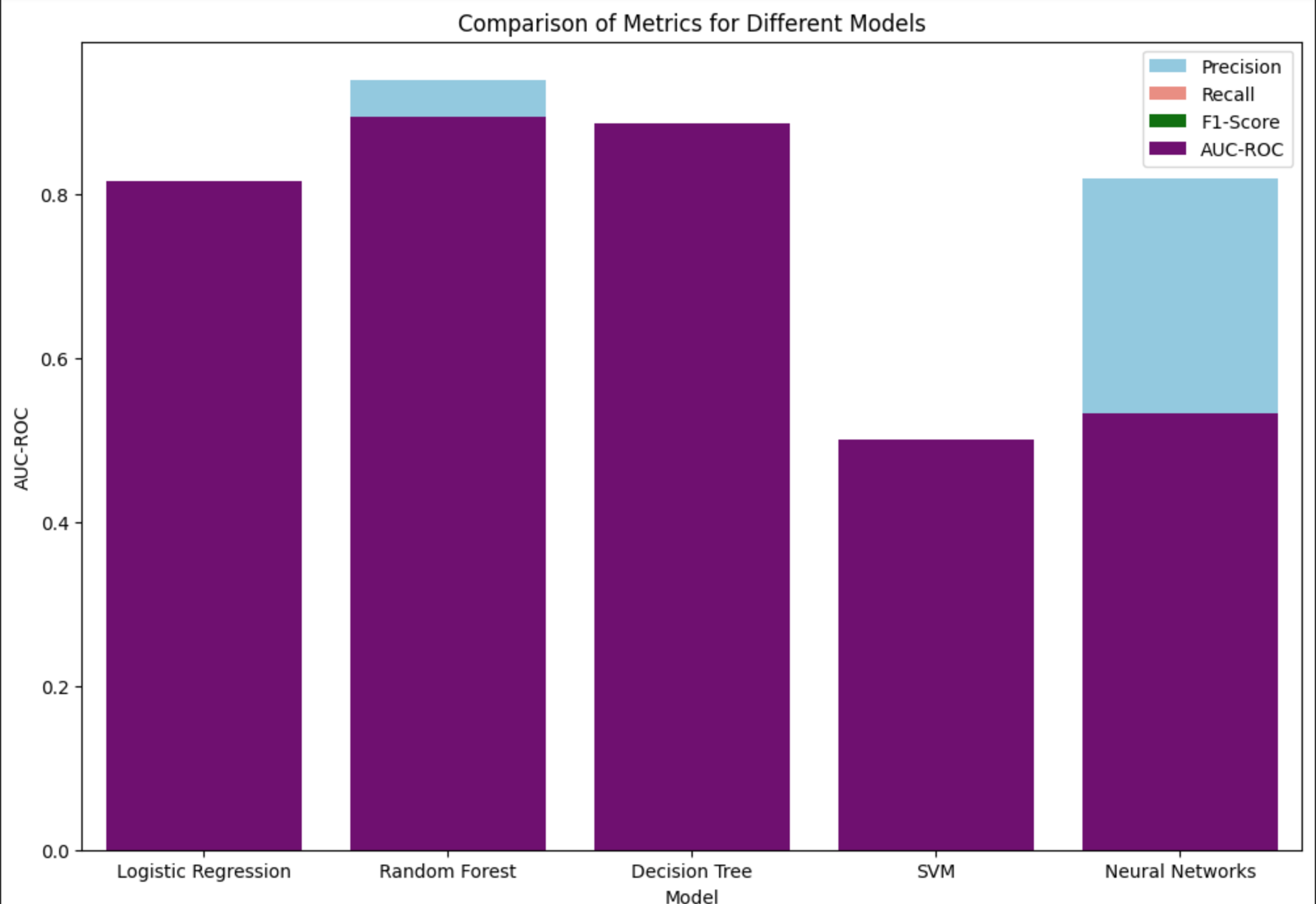
### **Future Enhancements:**

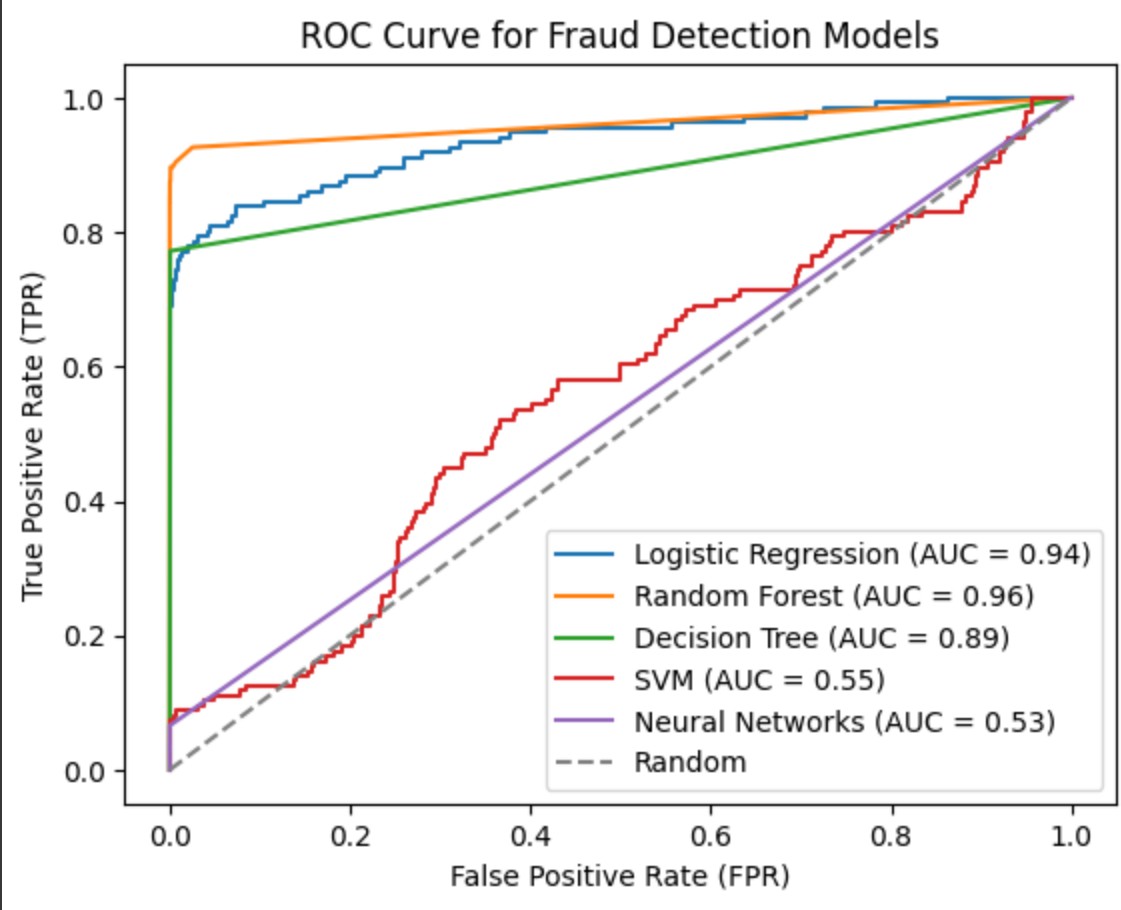
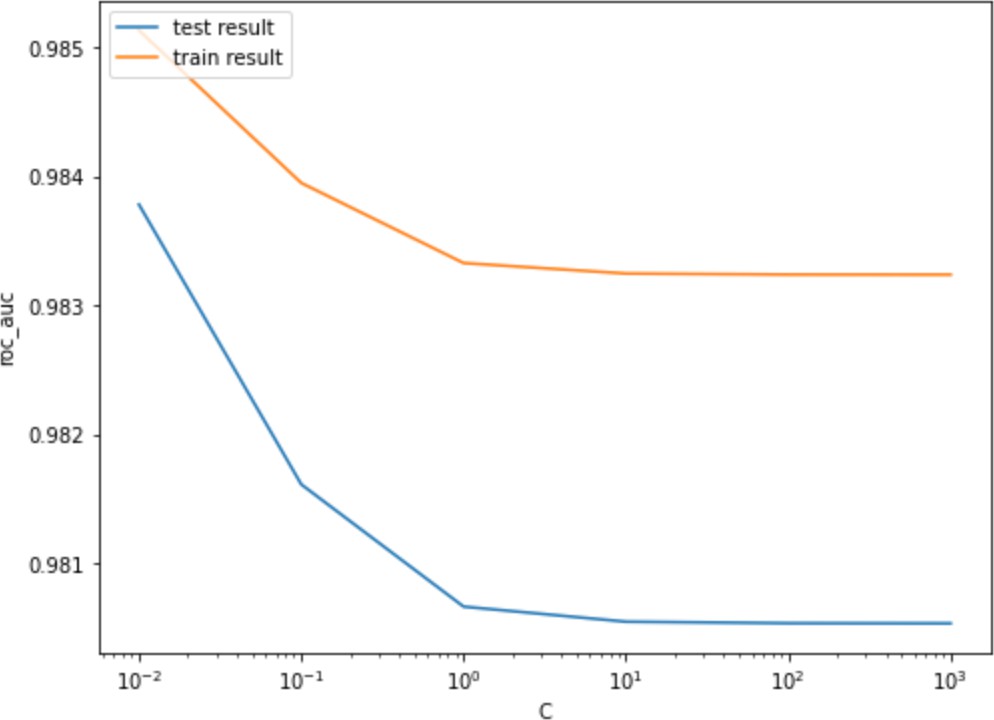
Moving forward, there exist several opportunities for enhancing the effectiveness and efficiency of our fraud detection system. Firstly, investing in data acquisition and enrichment strategies to access more comprehensive and diverse datasets can significantly enhance model performance and robustness. Leveraging advanced data analytics techniques, such as deep learning and anomaly detection, holds promise for detecting sophisticated fraudulent patterns and minimizing false positives. Furthermore, integrating real-time monitoring and alerting mechanisms into our system can enable proactive fraud prevention and mitigation strategies, thereby reducing financial losses and preserving customer trust. Collaboration with industry partners and regulatory bodies to establish data-sharing frameworks and standards can facilitate access to valuable data sources while ensuring compliance with privacy and security regulations.

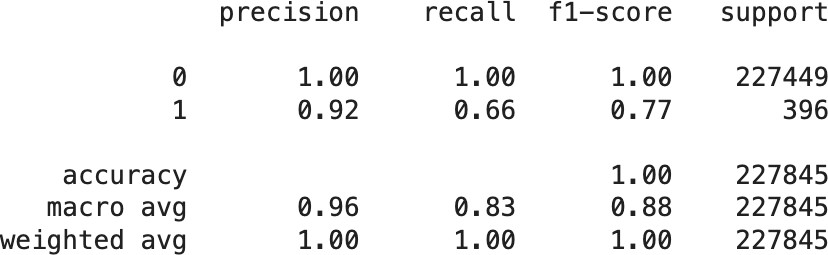
### **Inference:**

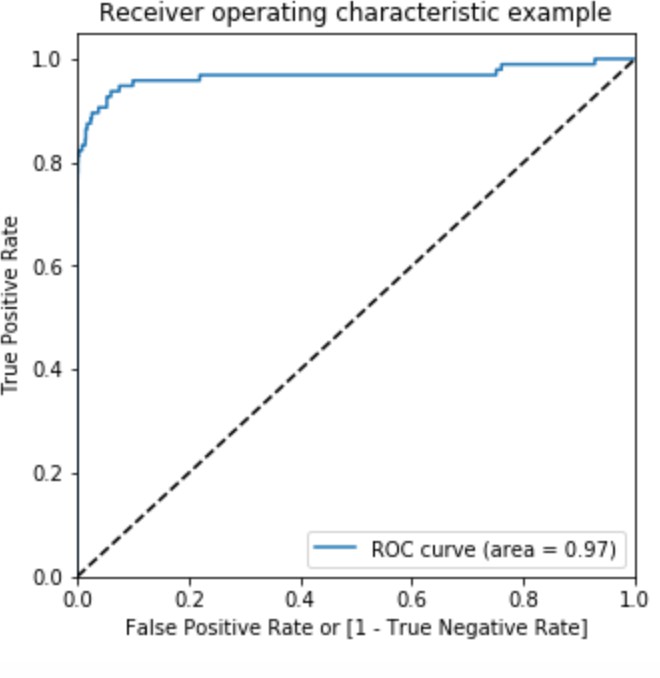
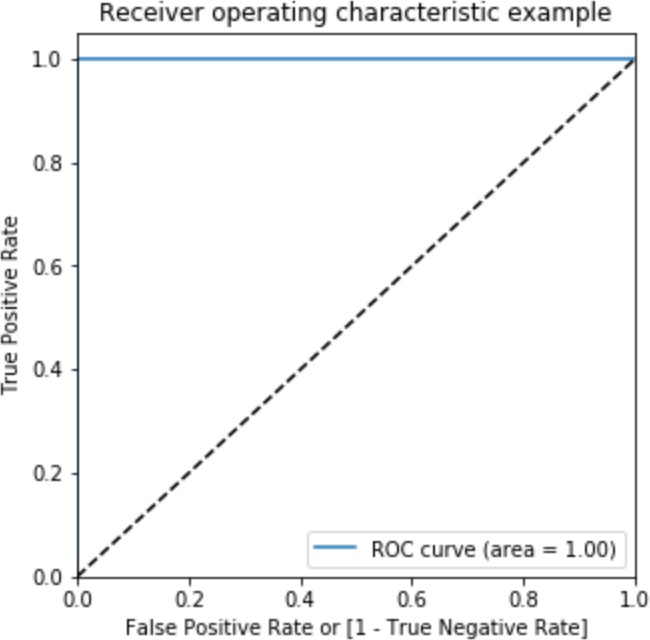
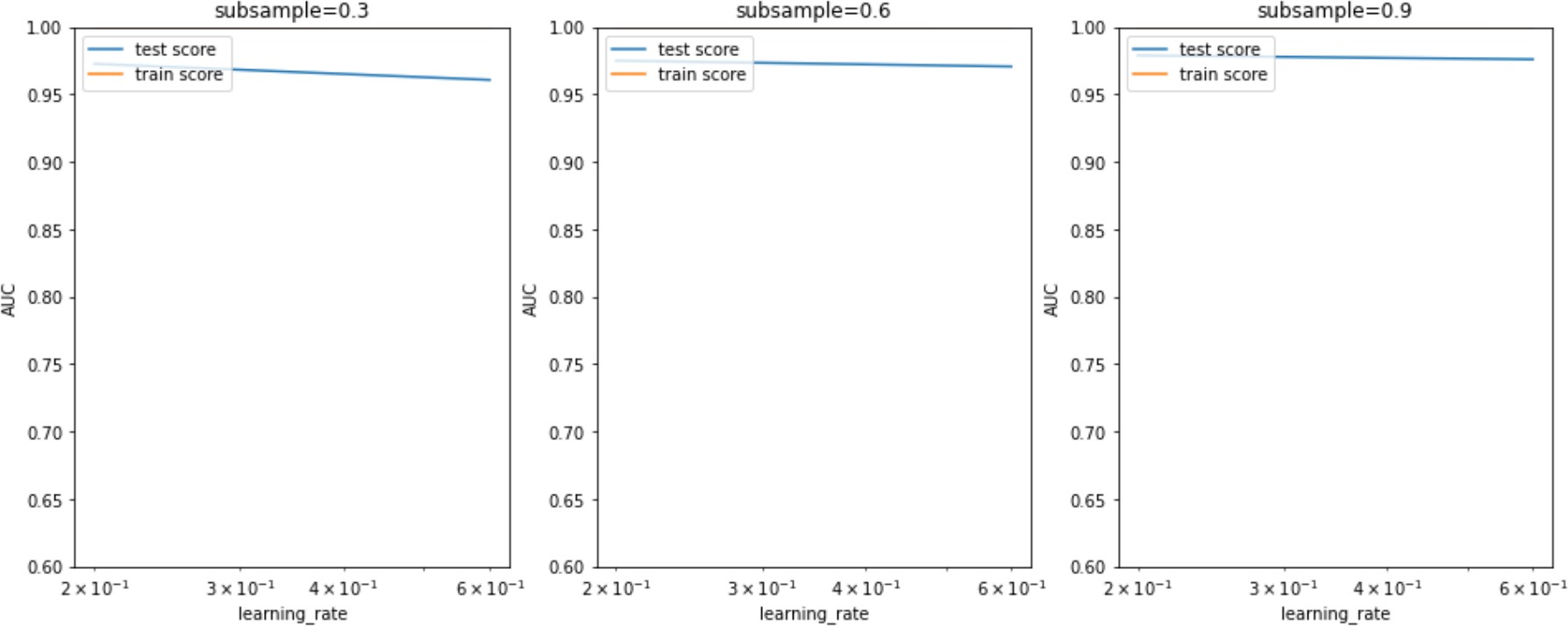
In conclusion, our project represents a significant step forward in addressing the pervasive challenge of fraud detection in the banking sector. Despite encountering limitations and constraints, our efforts have yielded valuable insights and laid the groundwork for future research and development in this domain. By embracing a collaborative and innovative approach, we can continue to refine and enhance our fraud detection system, ultimately bolstering financial security and safeguarding the interests of stakeholders. As we navigate the evolving landscape of financial crime, it is imperative to remain vigilant, adaptable, and proactive in our efforts to combat fraud and uphold the integrity of the financial ecosystem.

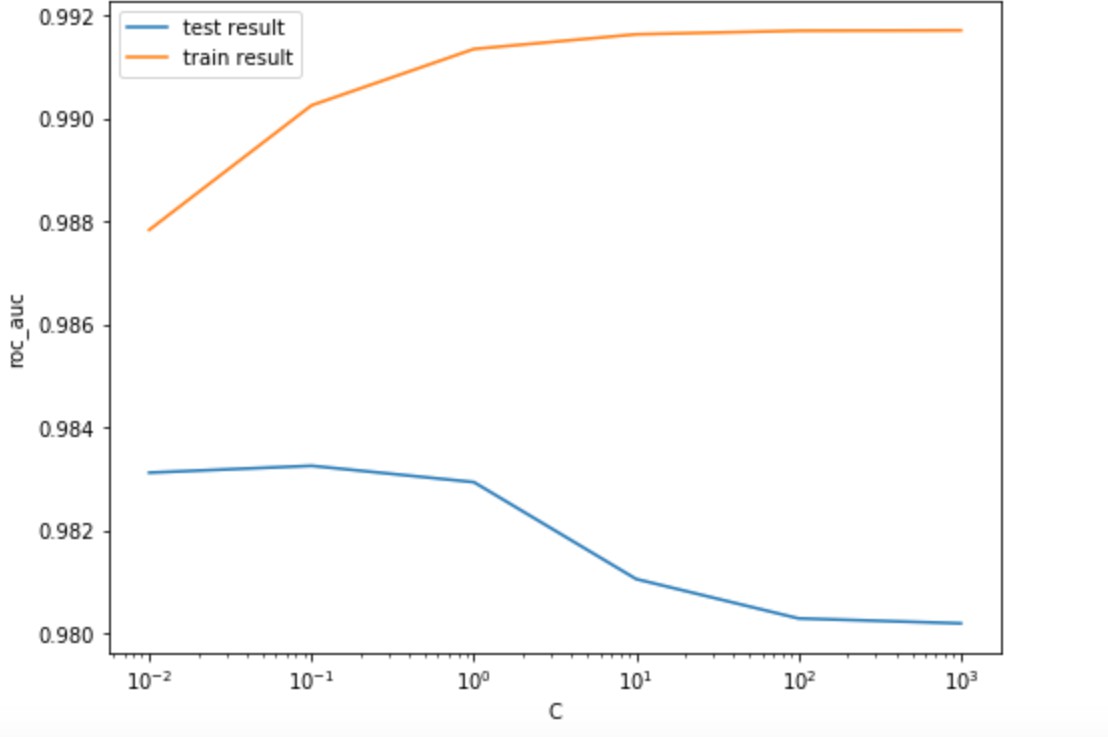
**APPENDIX A(Screenshots)**

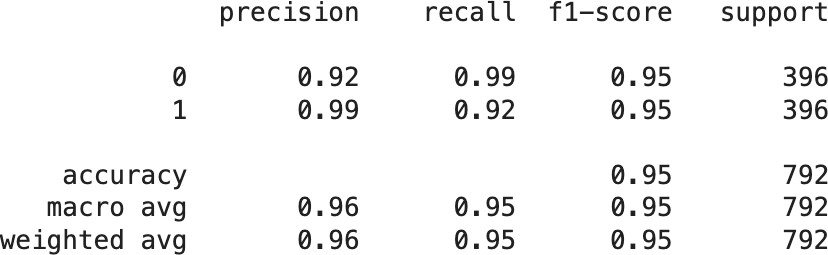
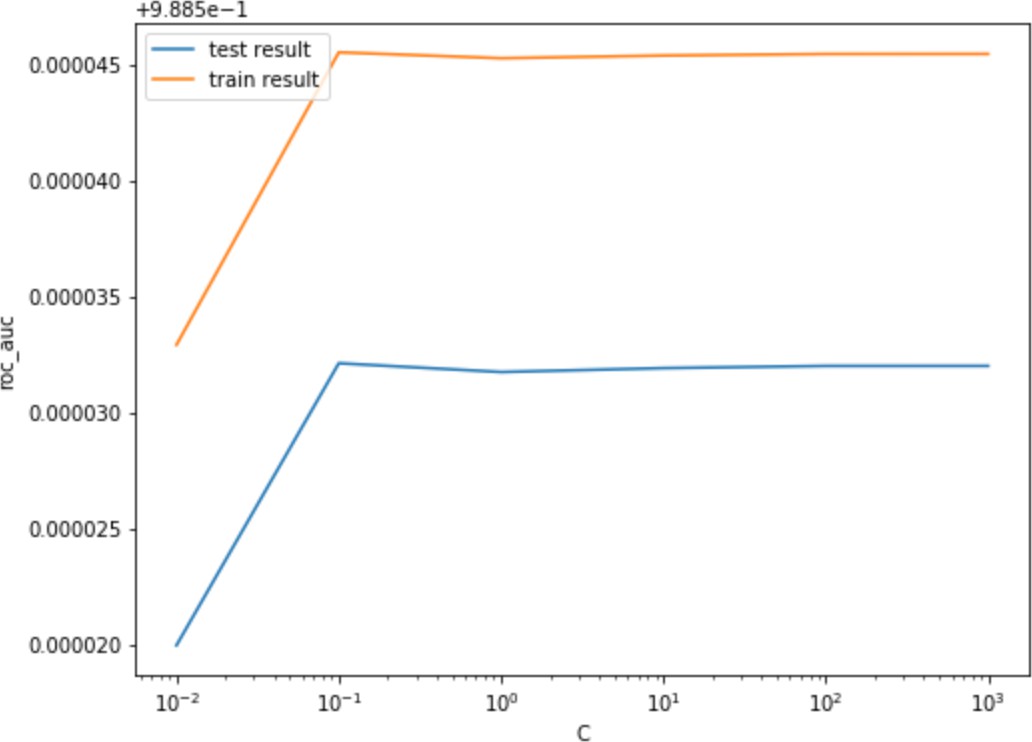
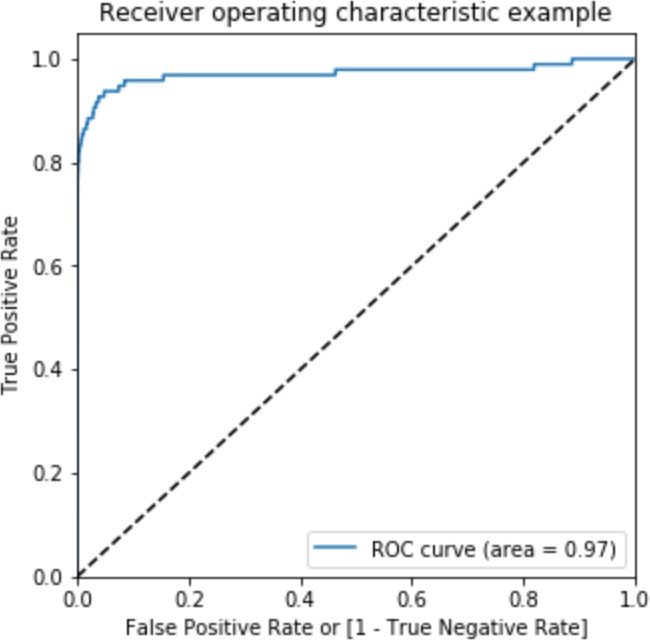


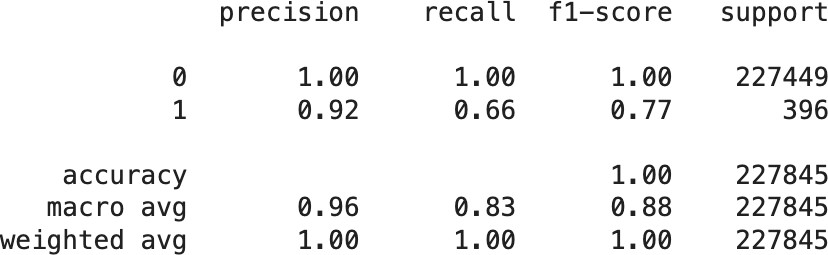
 

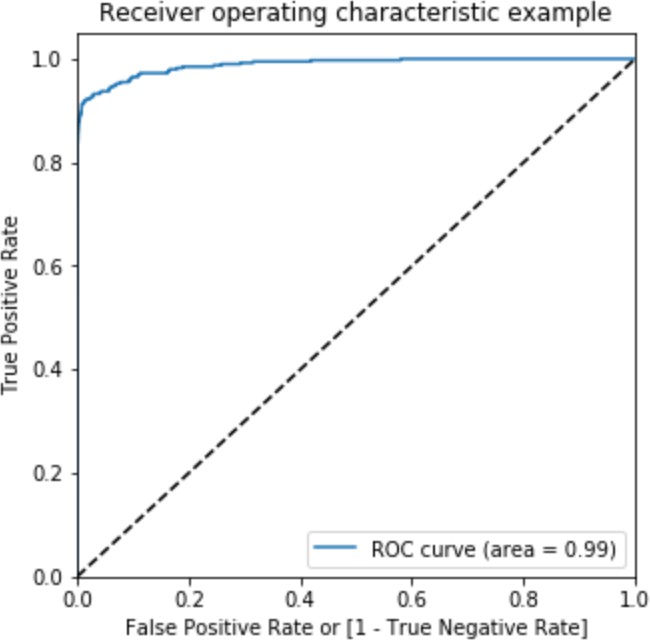
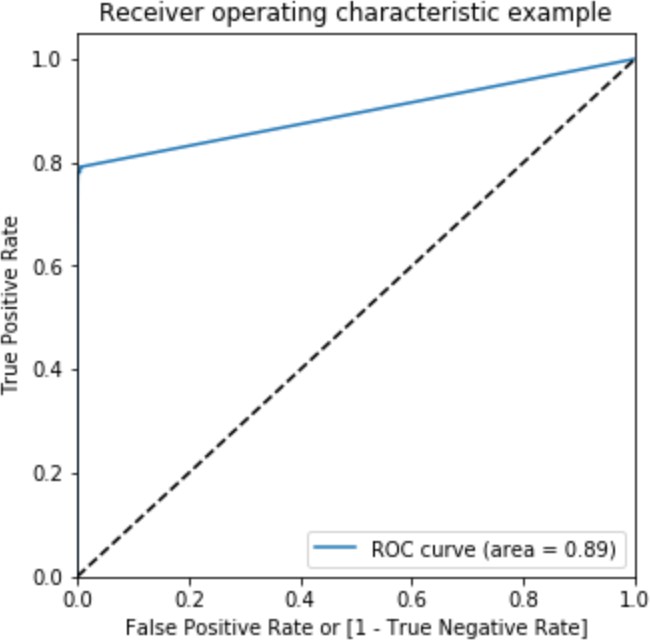


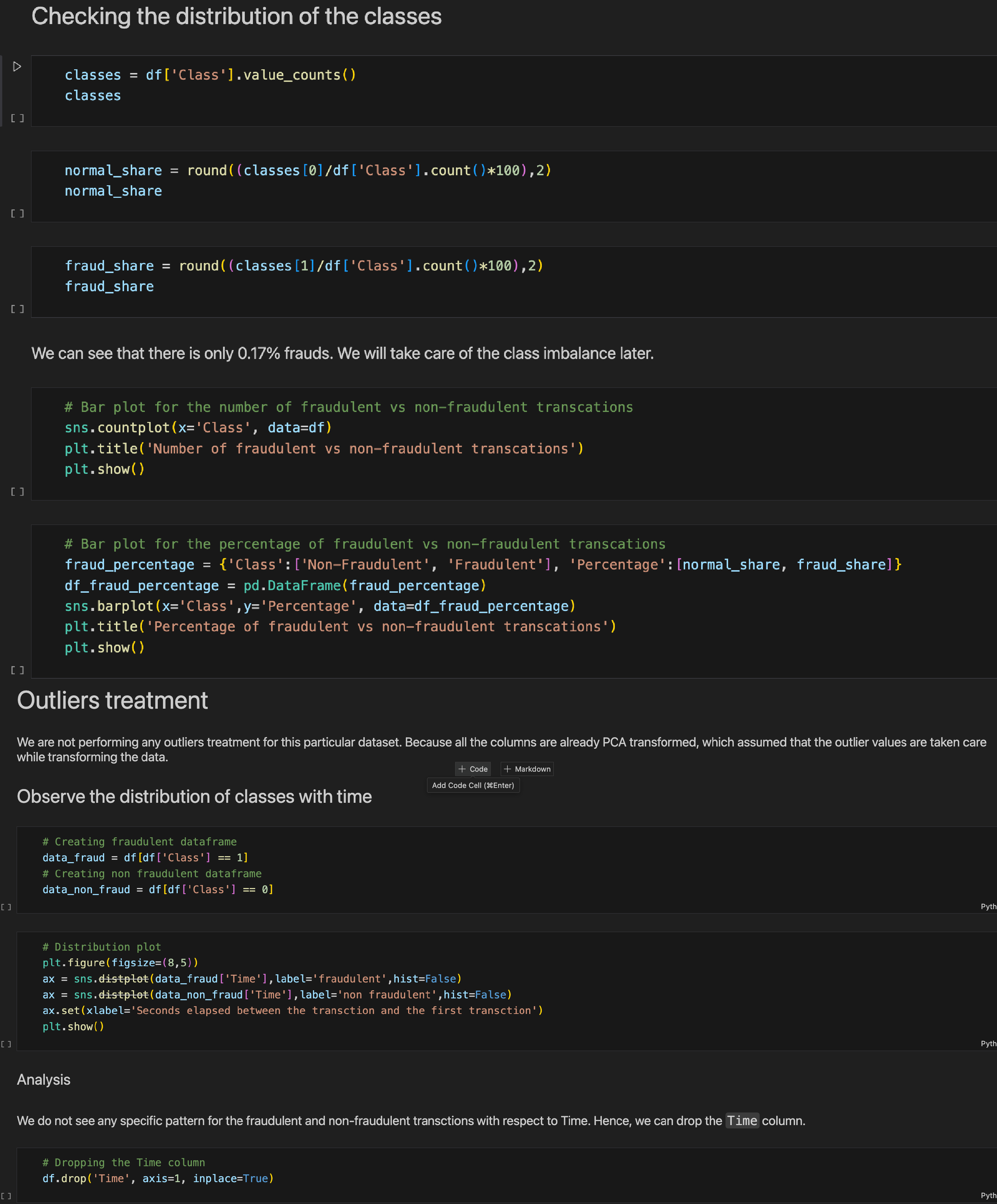


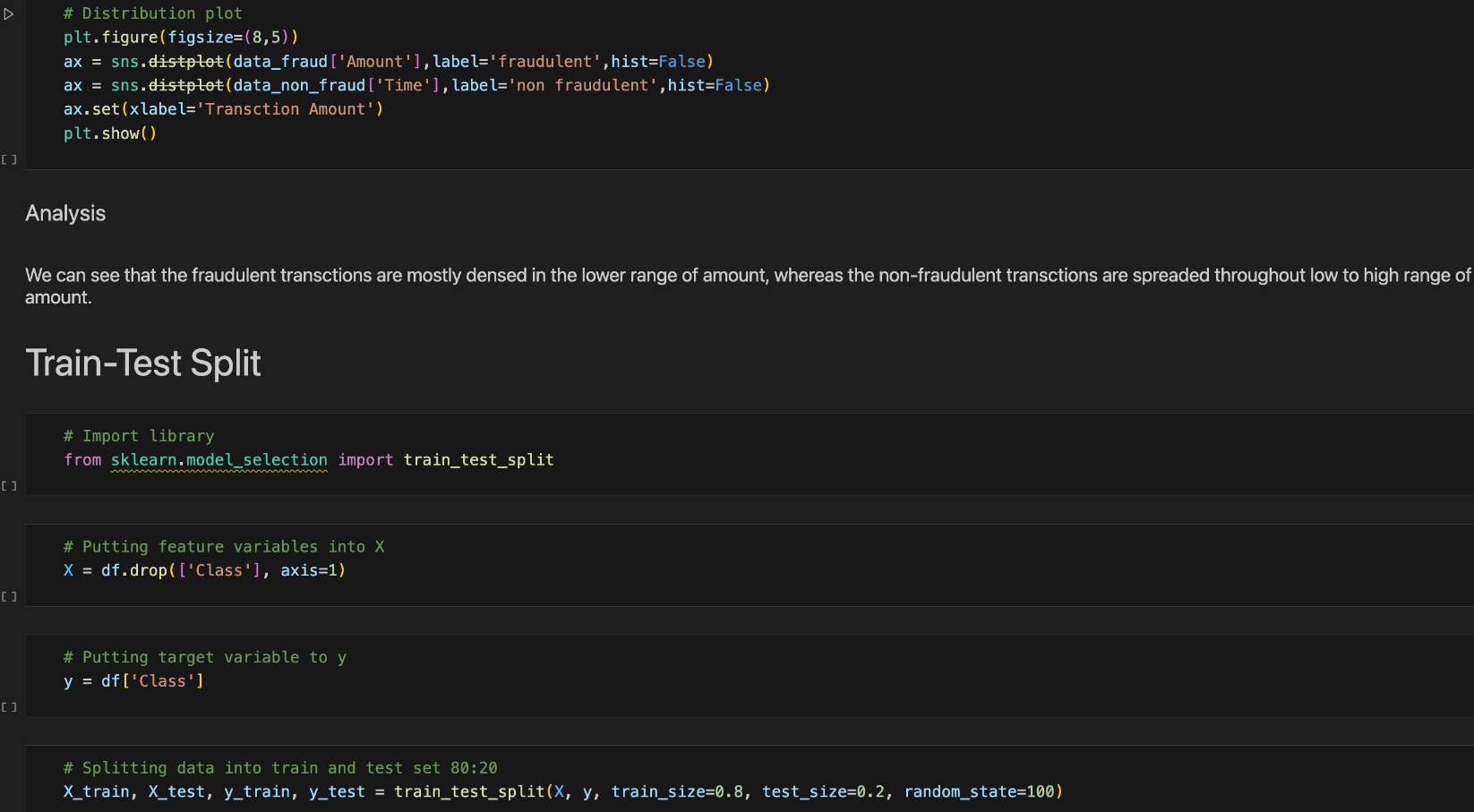
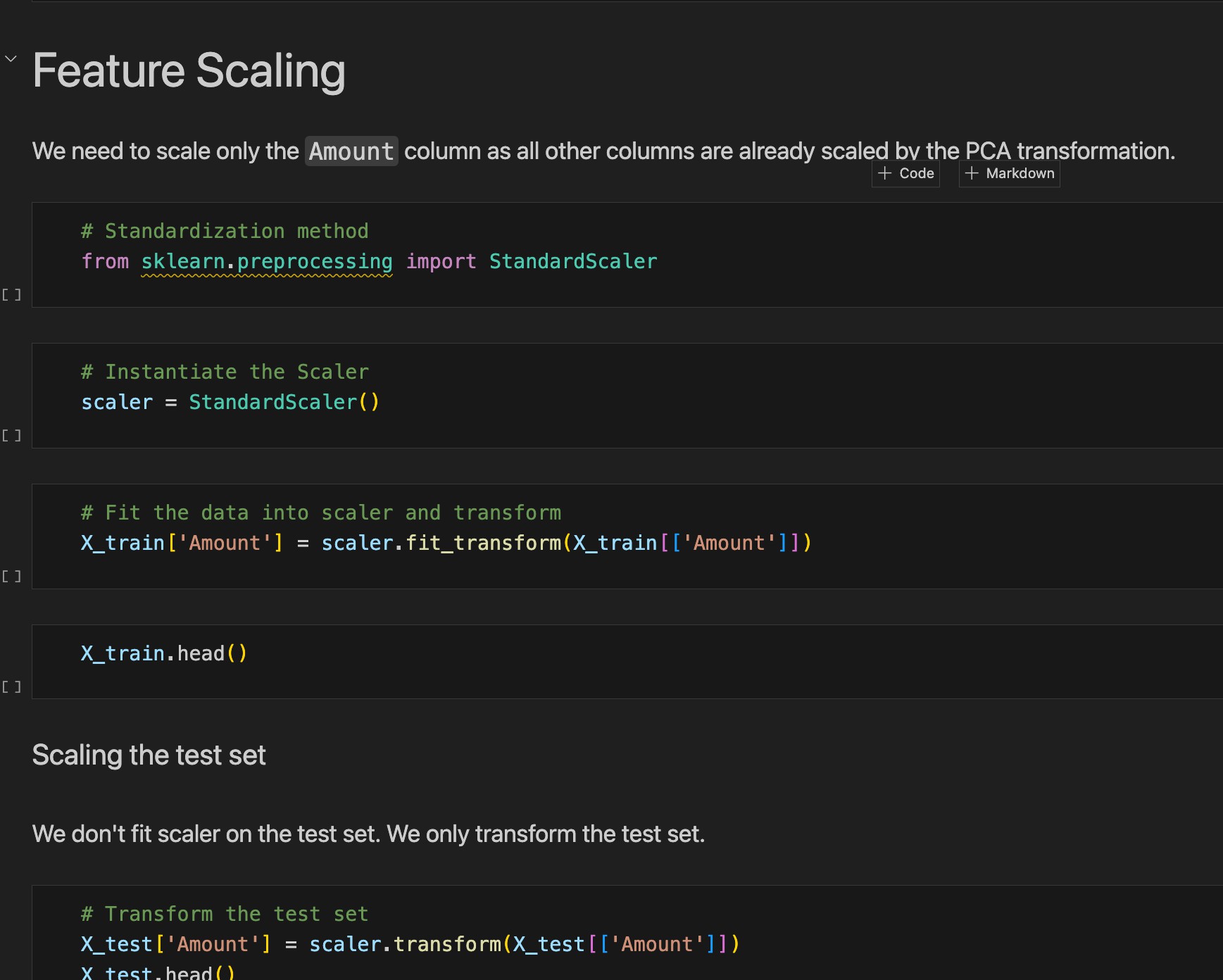


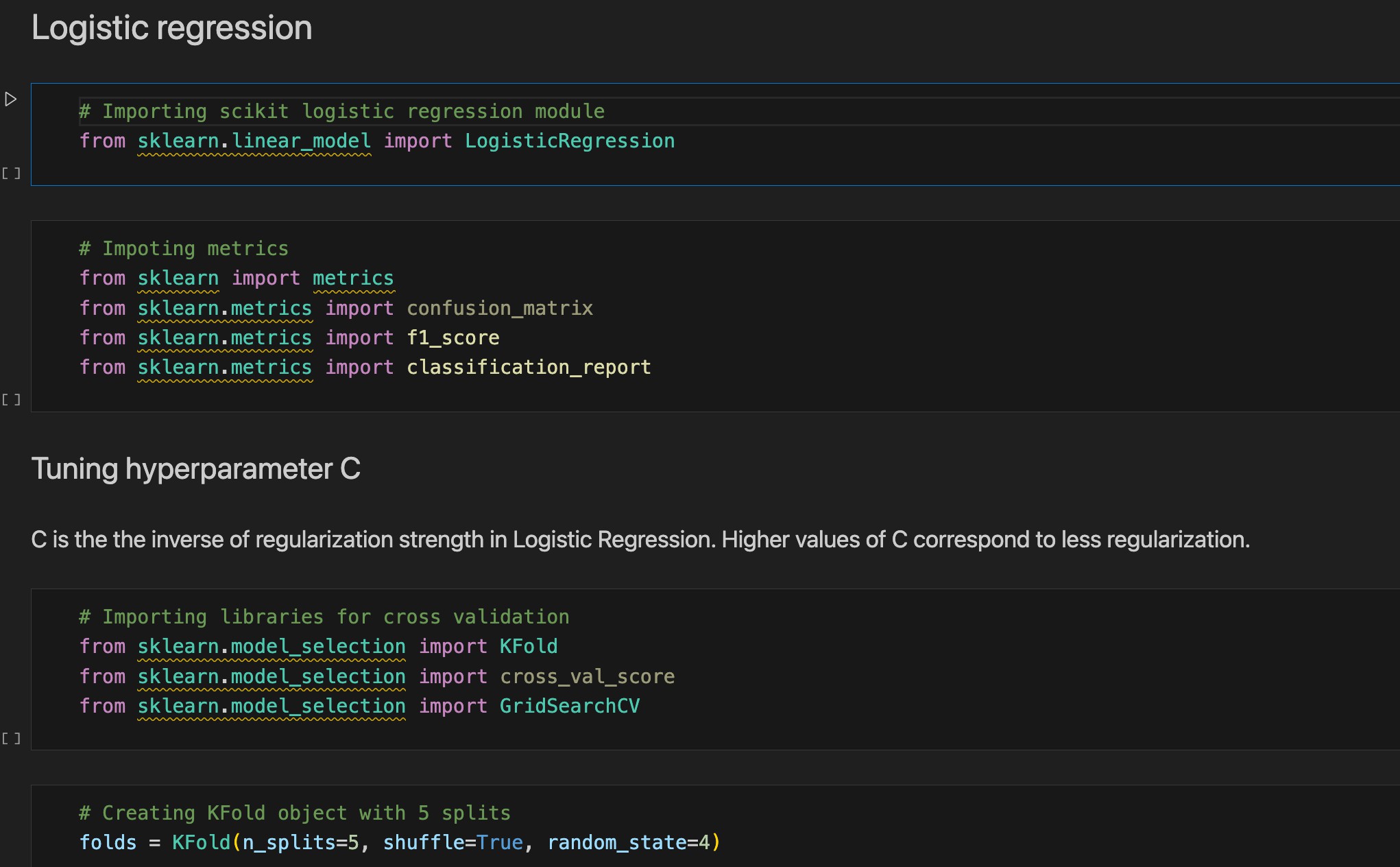


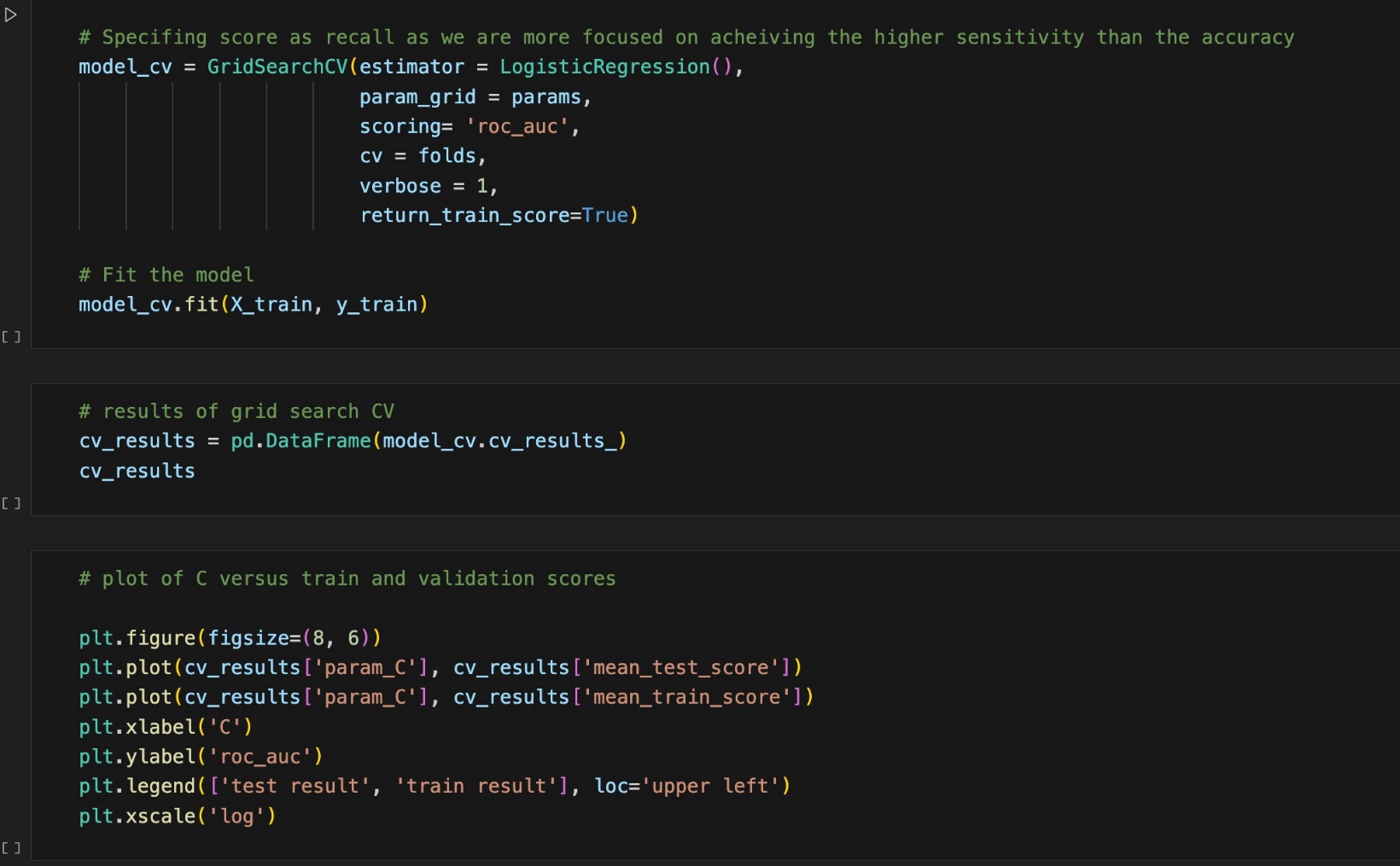


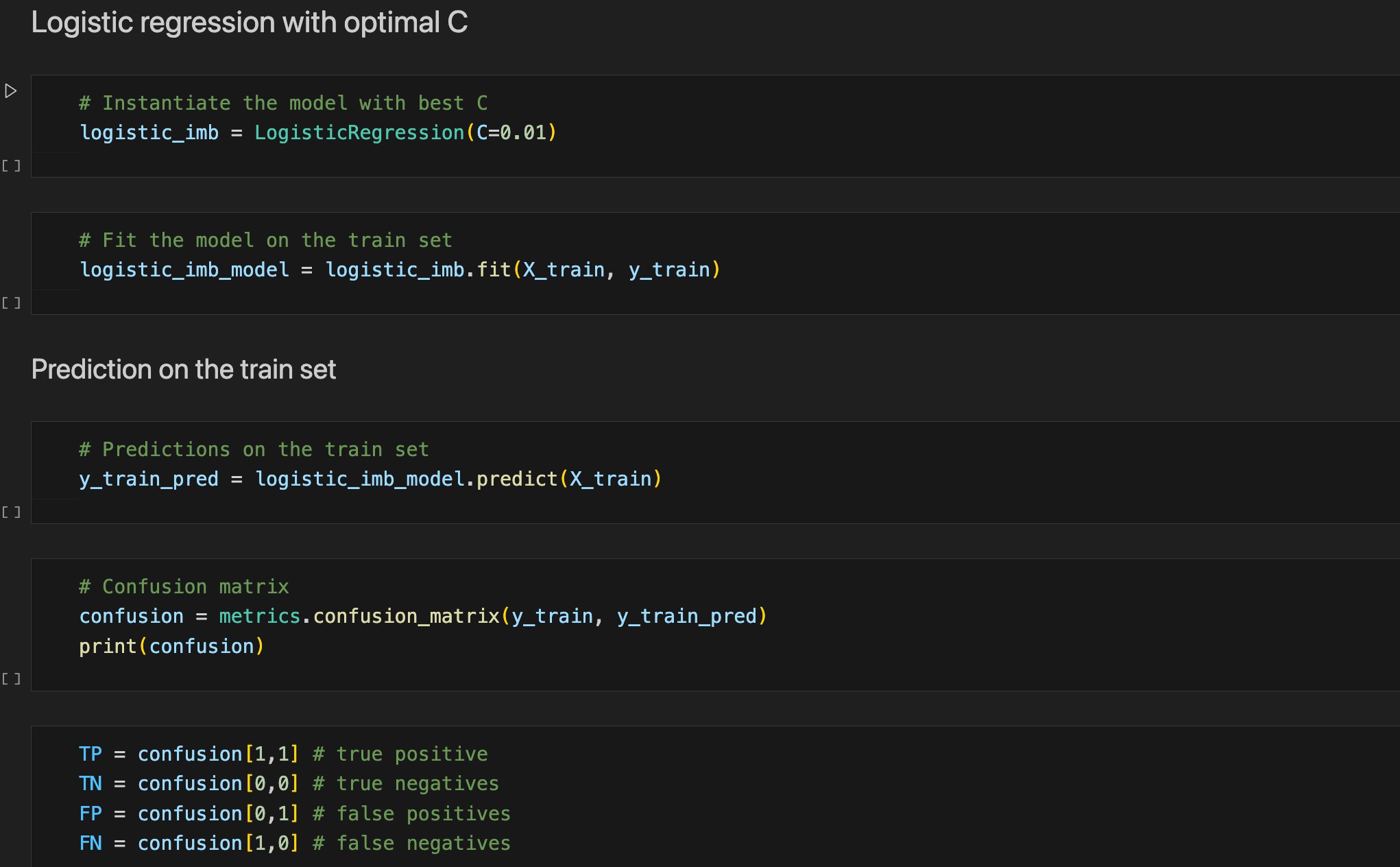


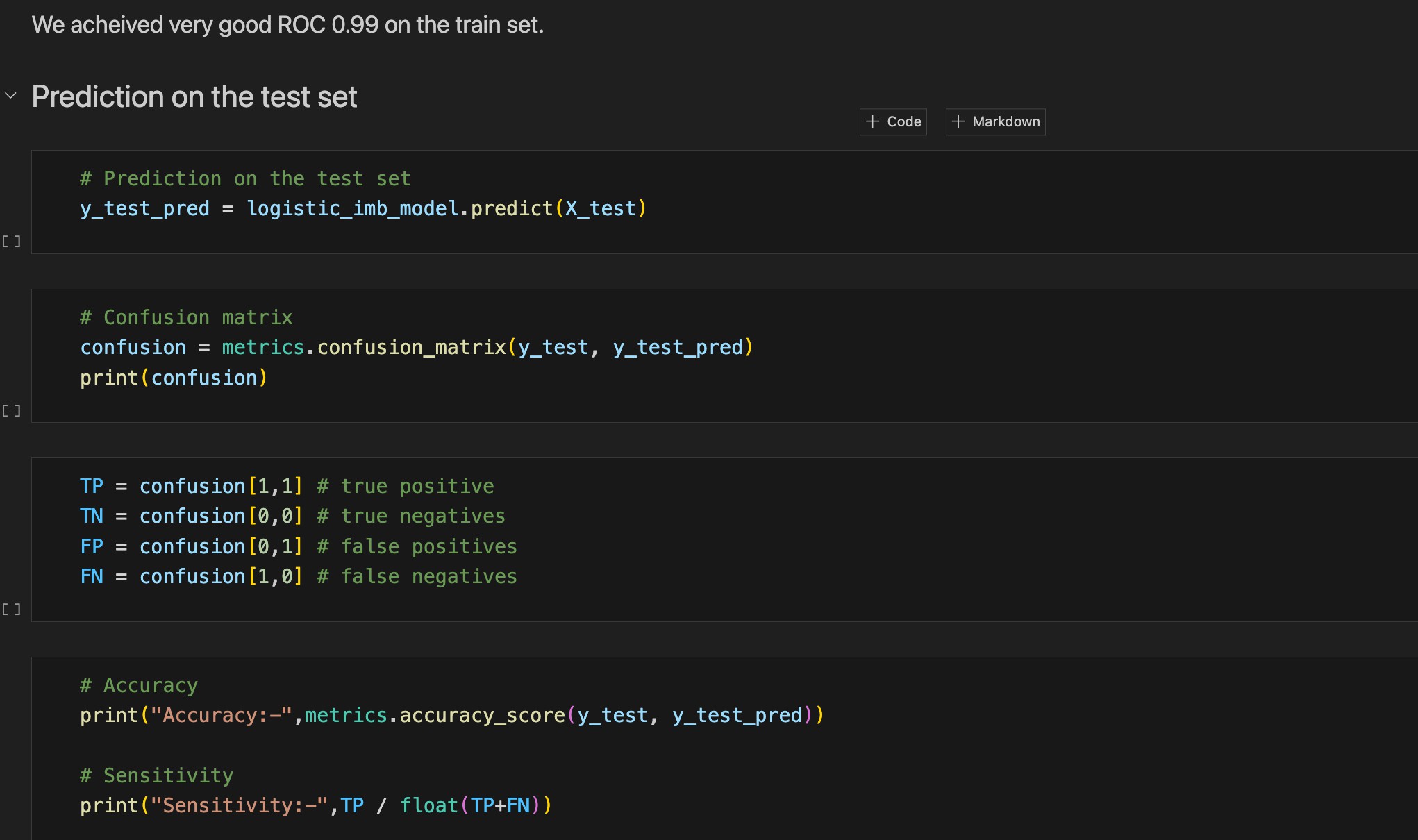
**Appendix-B(Coding)**

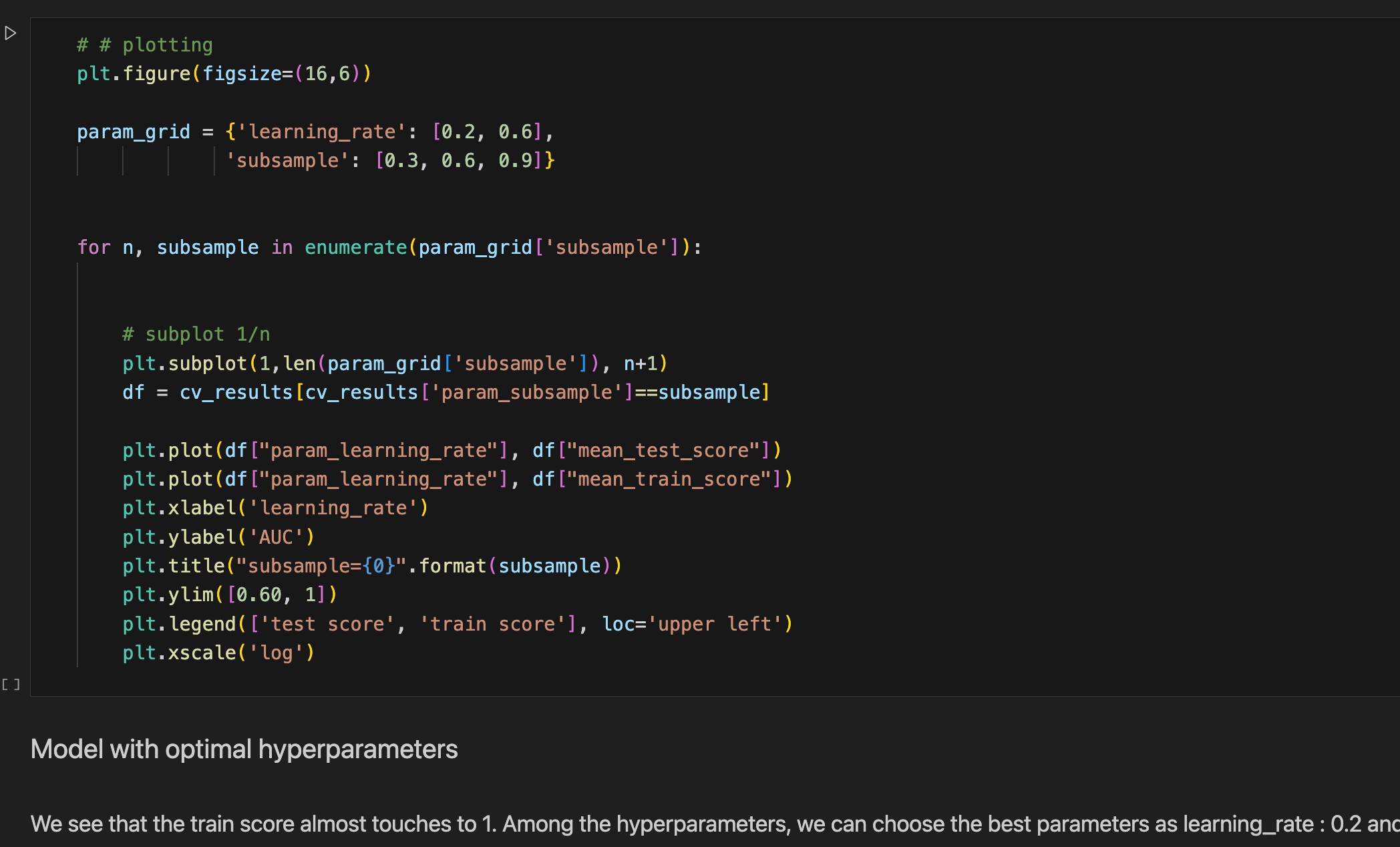


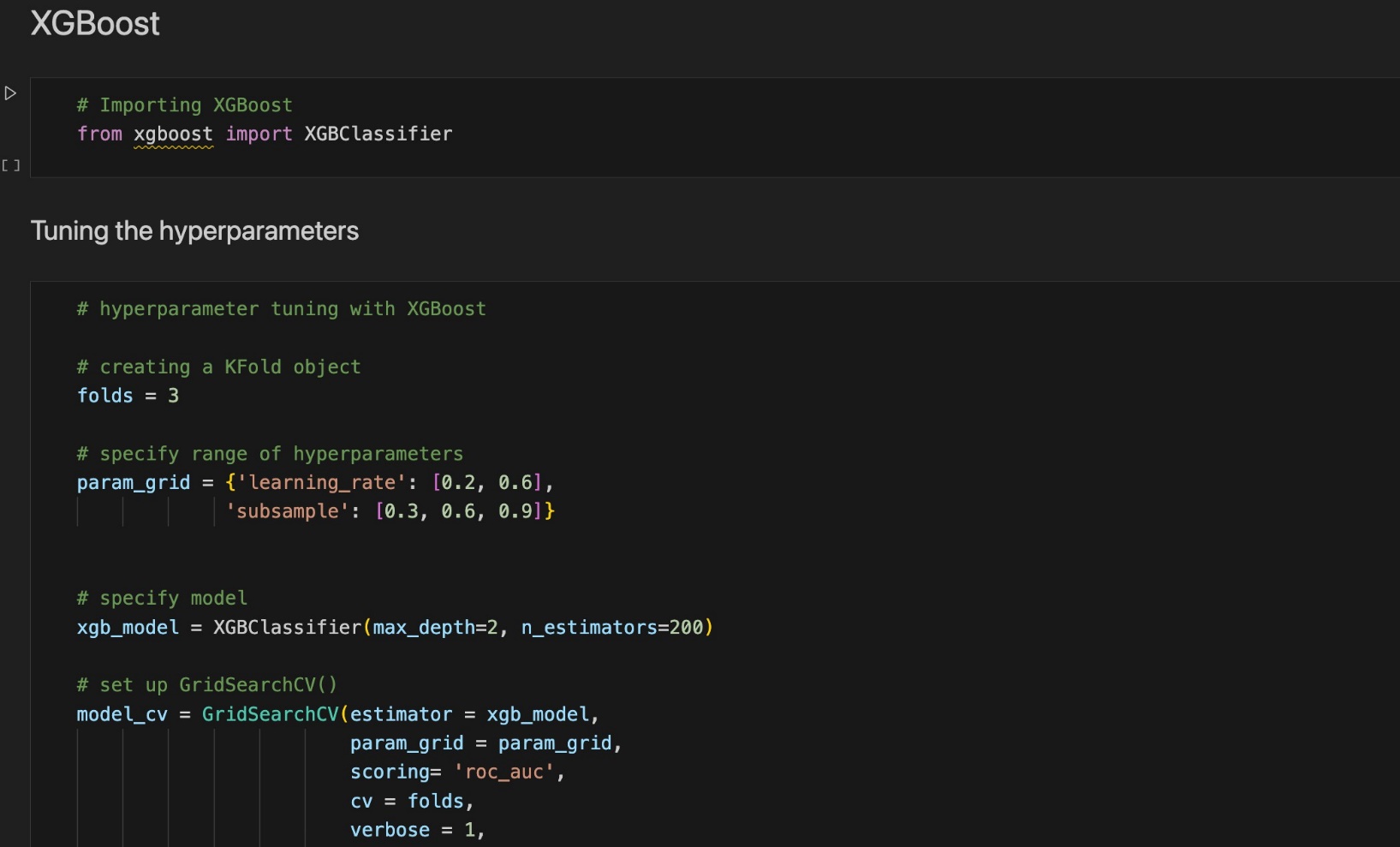
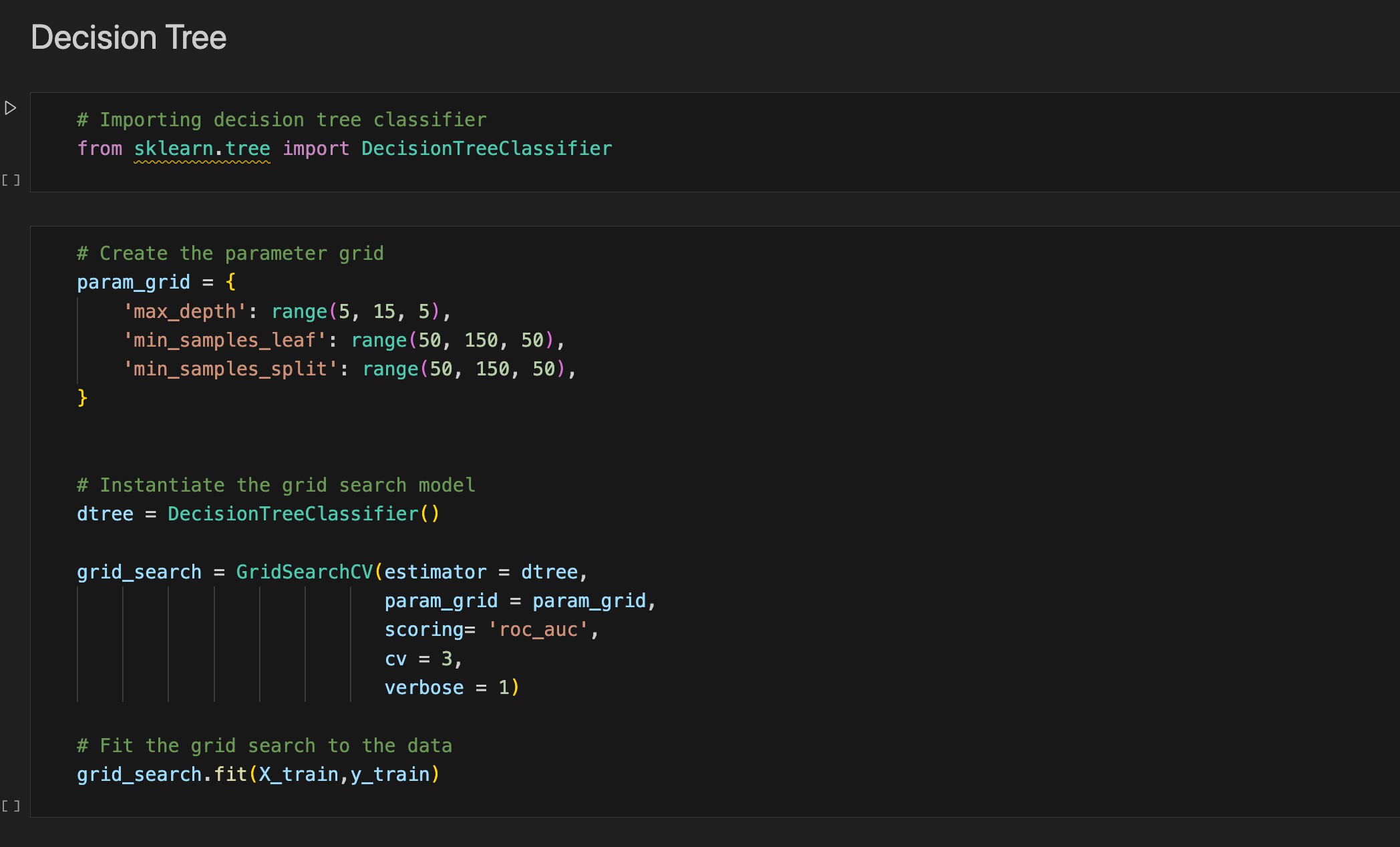


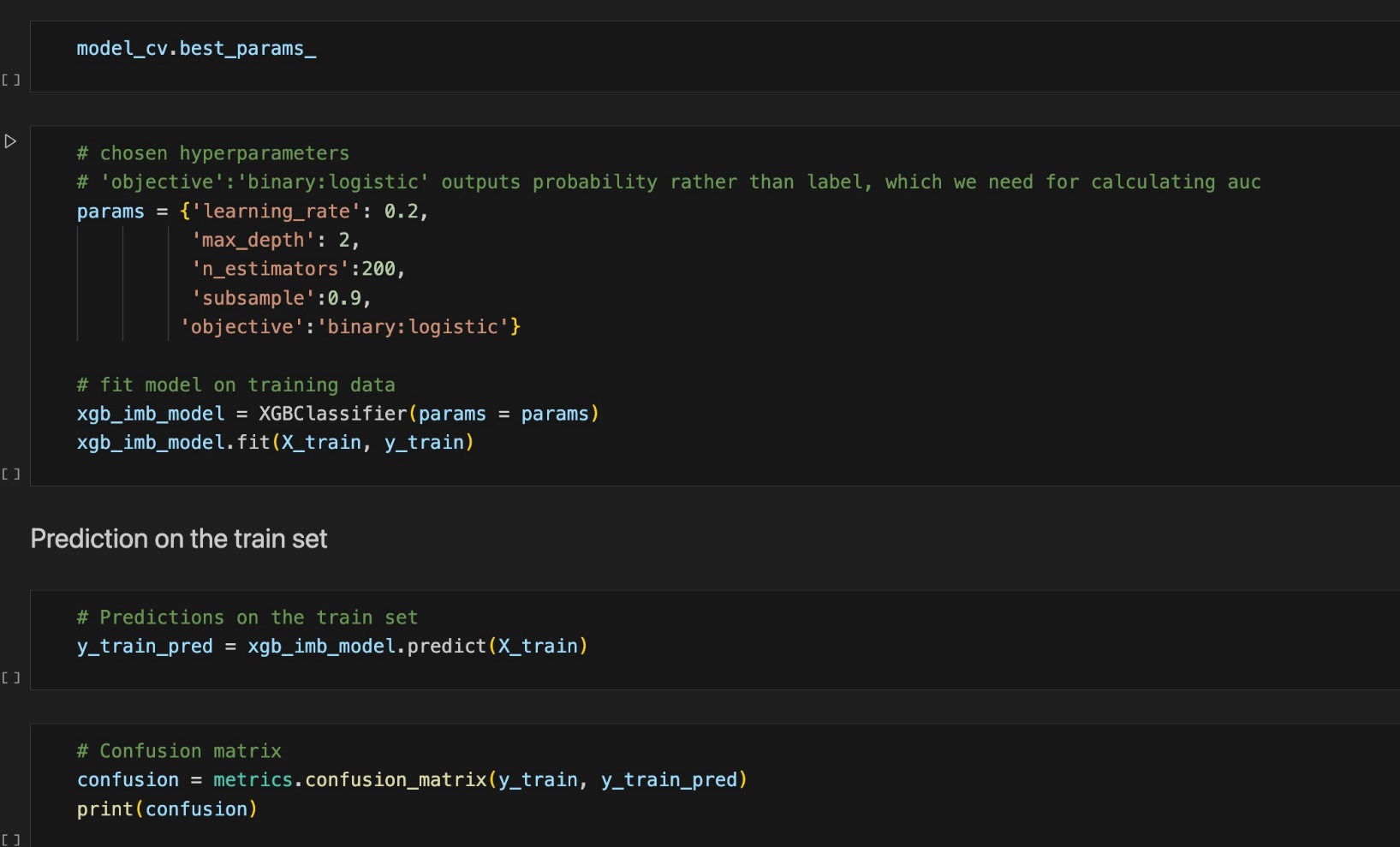


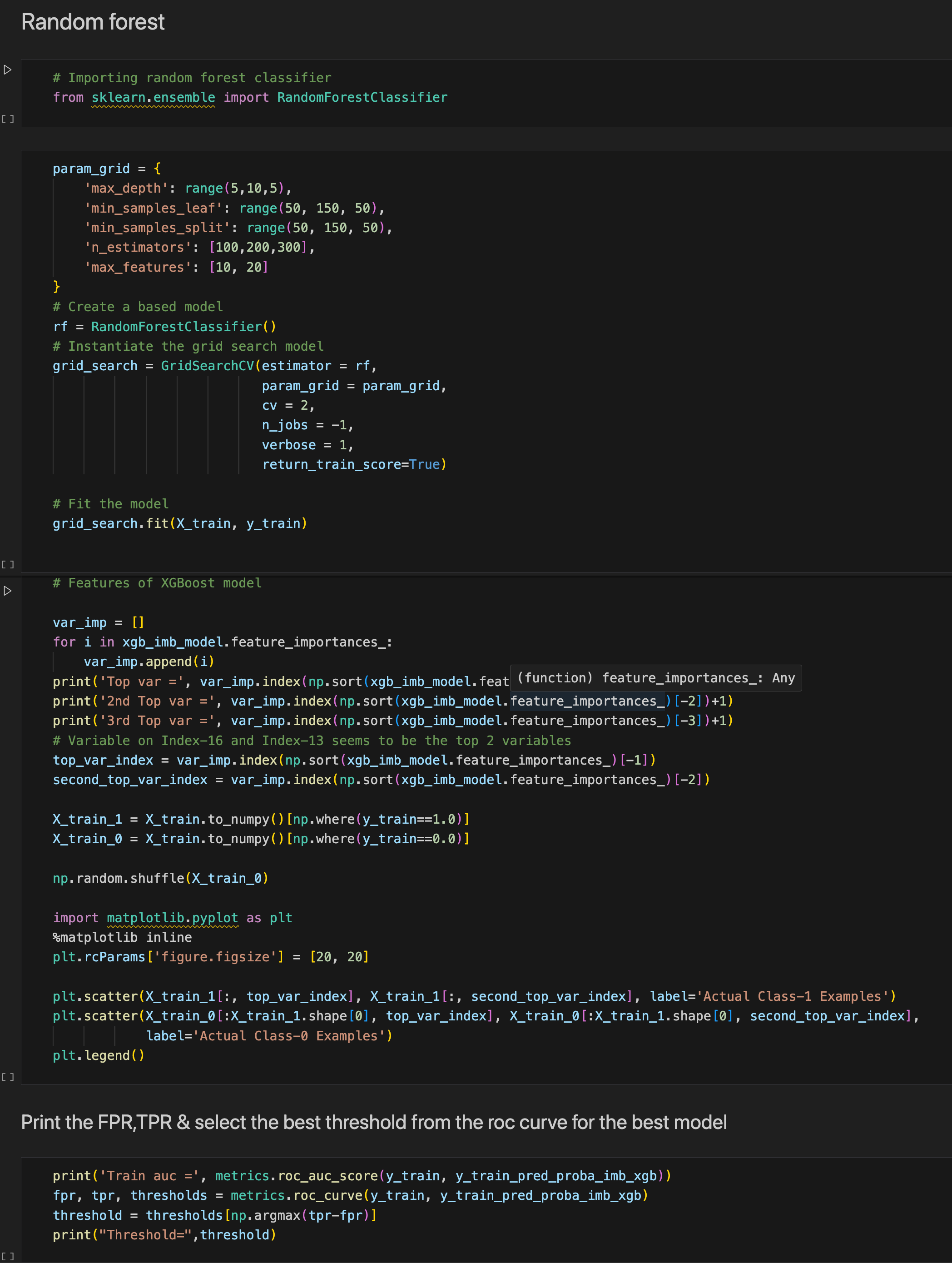


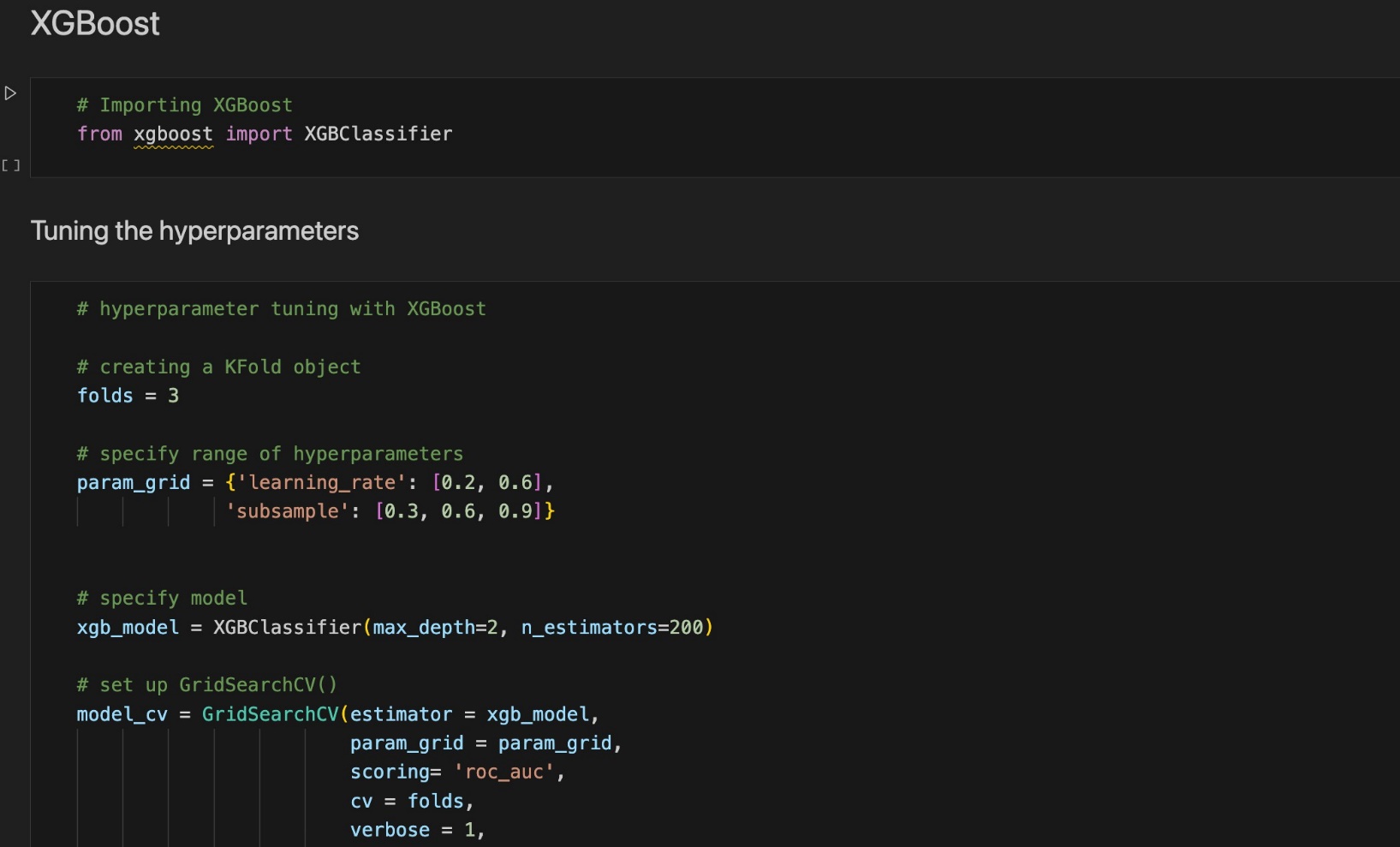


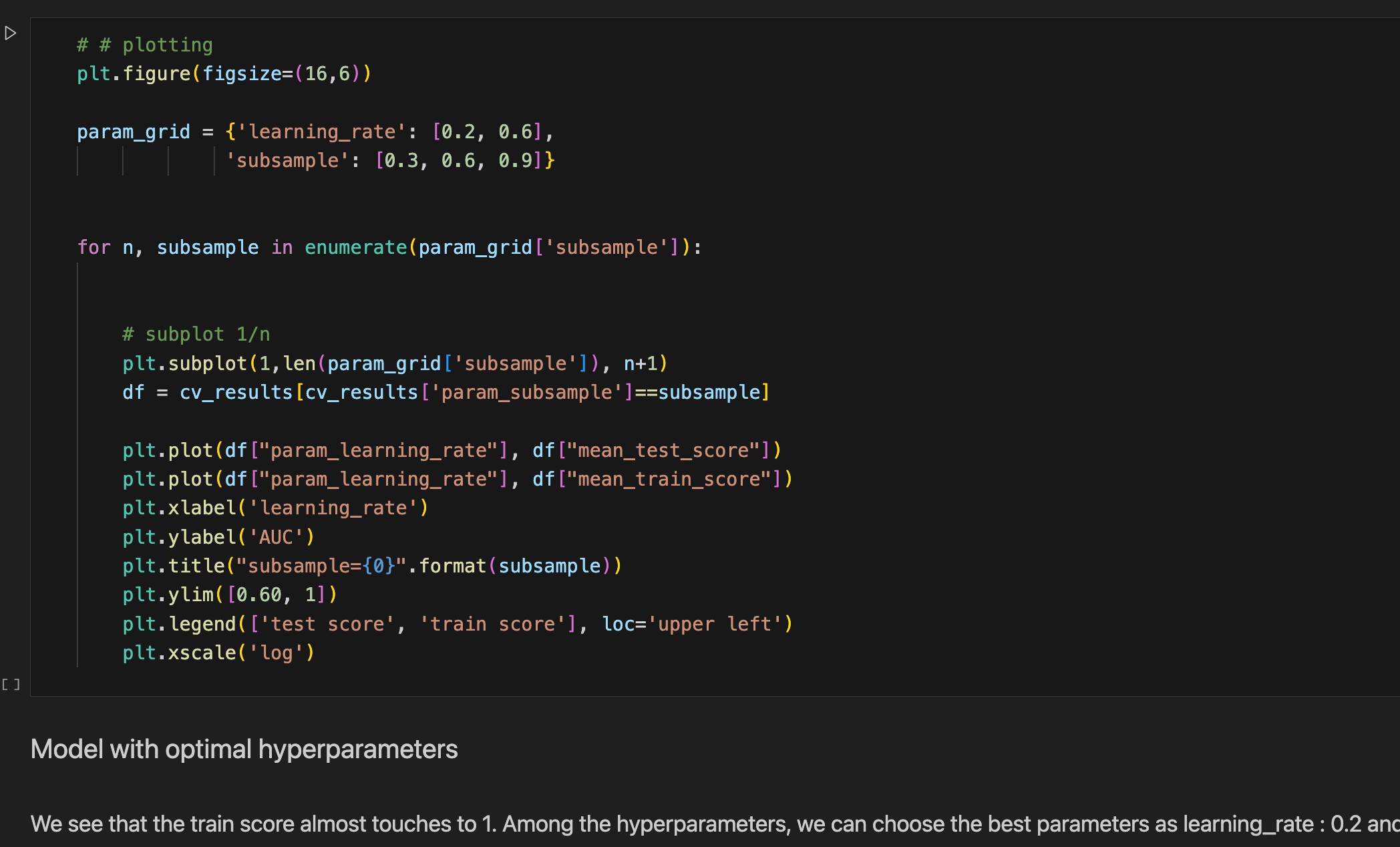


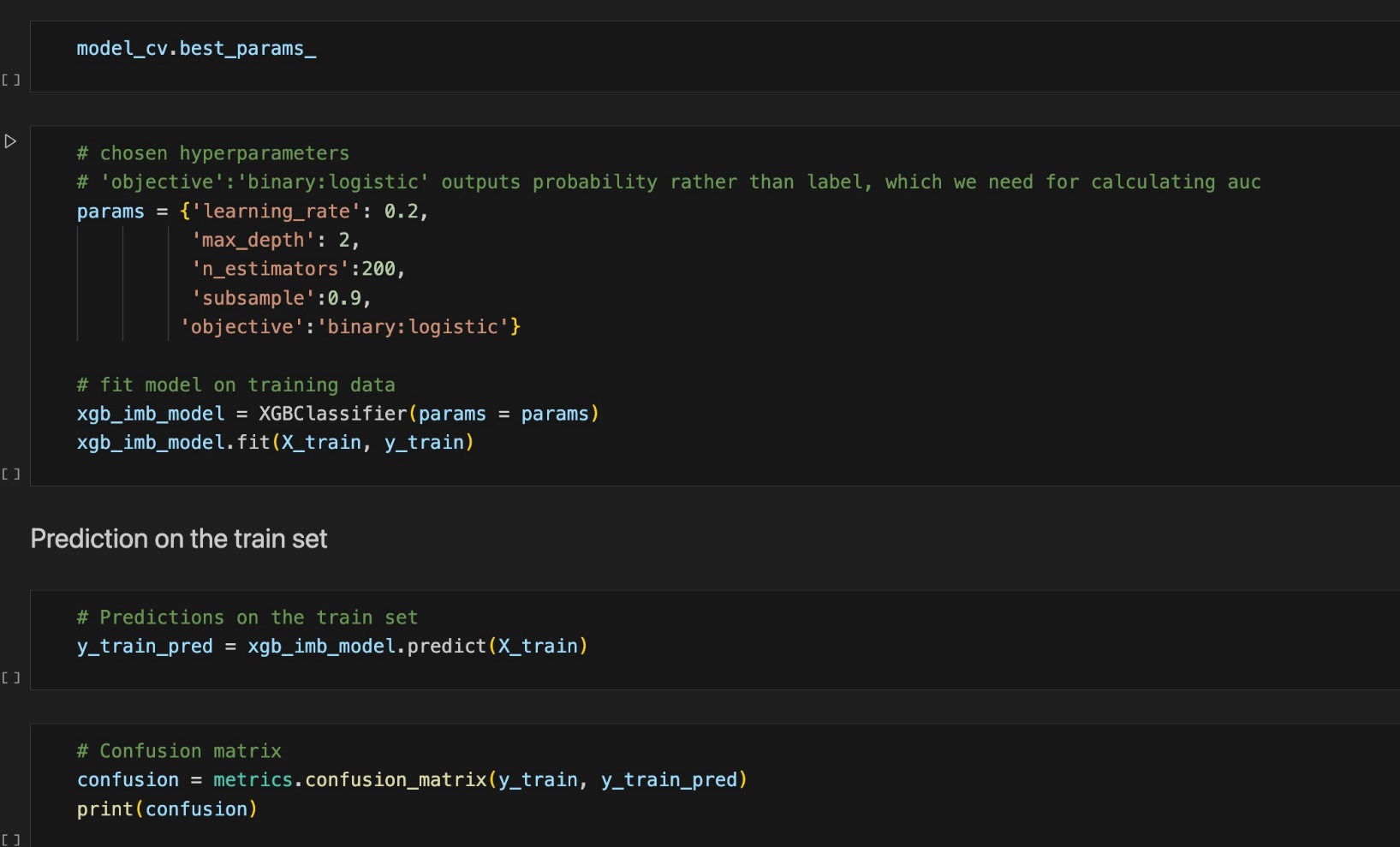
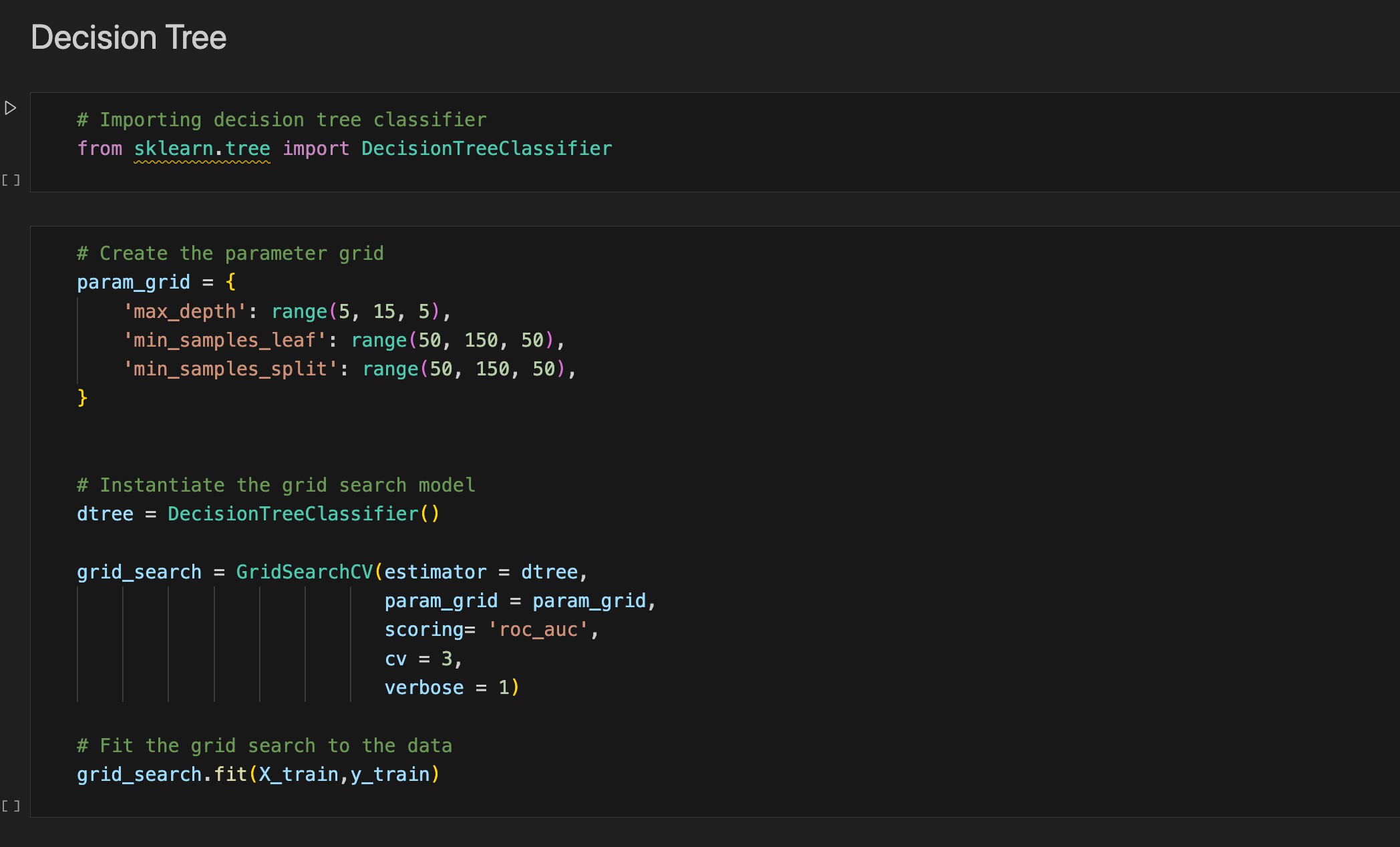


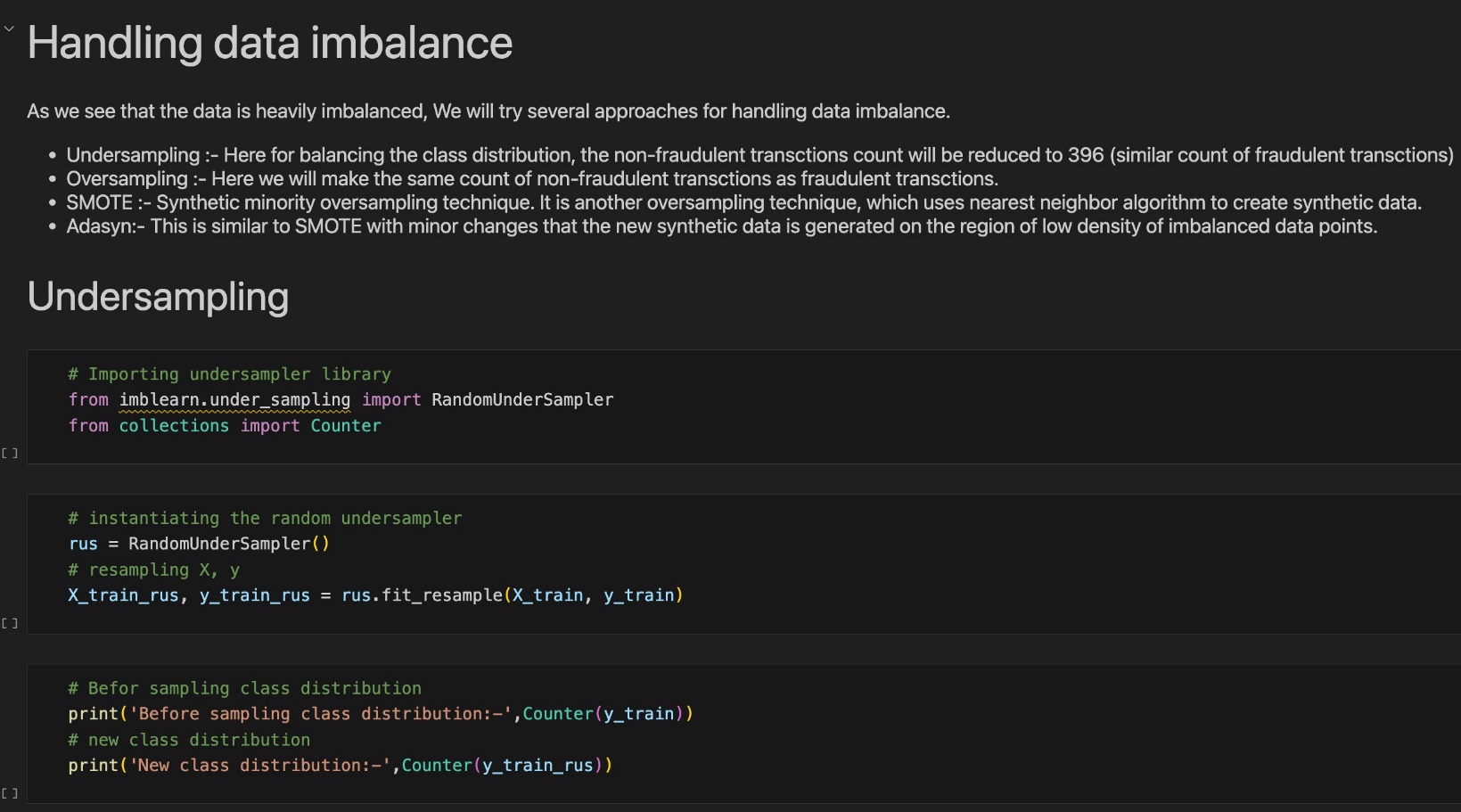
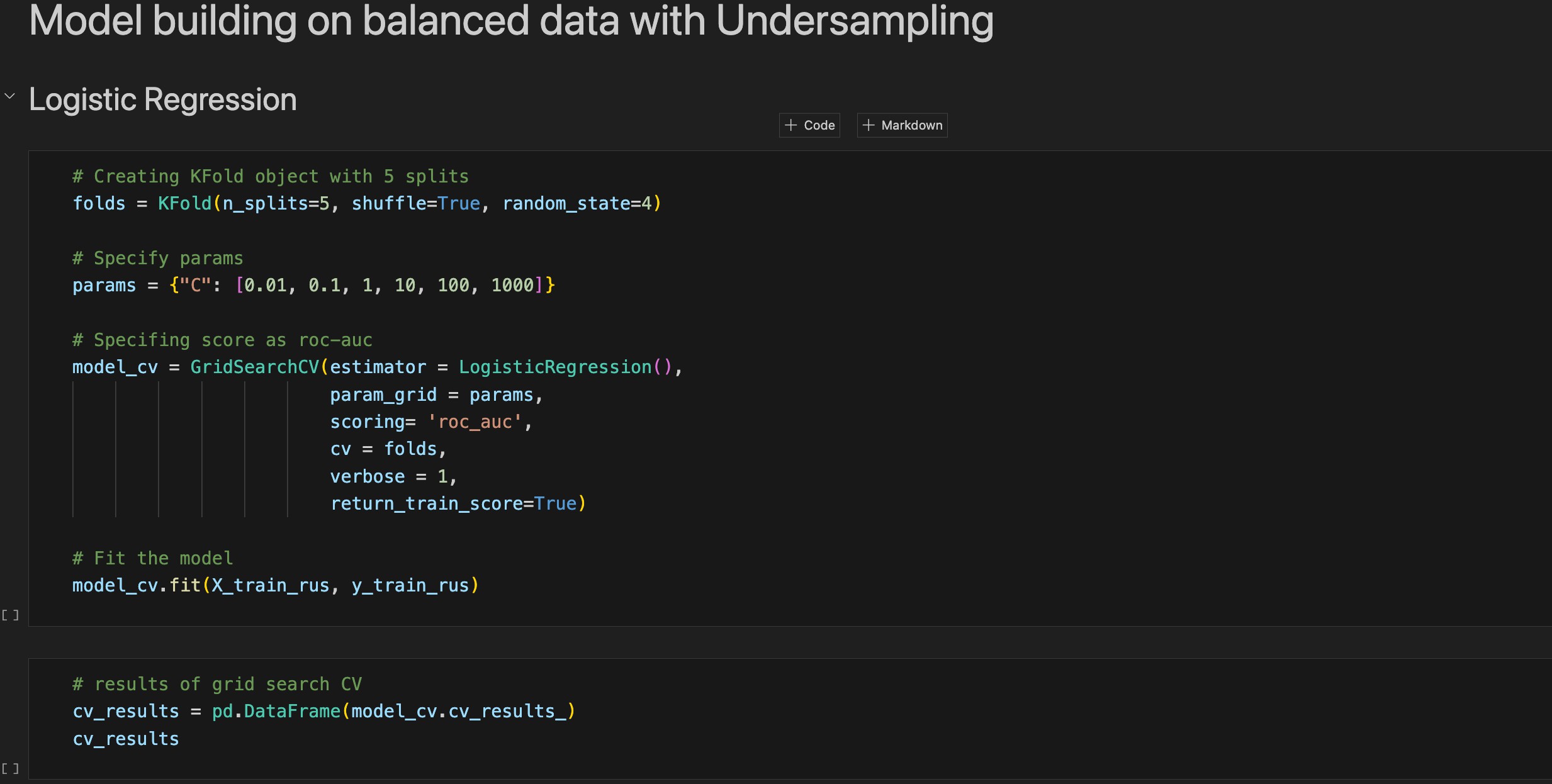




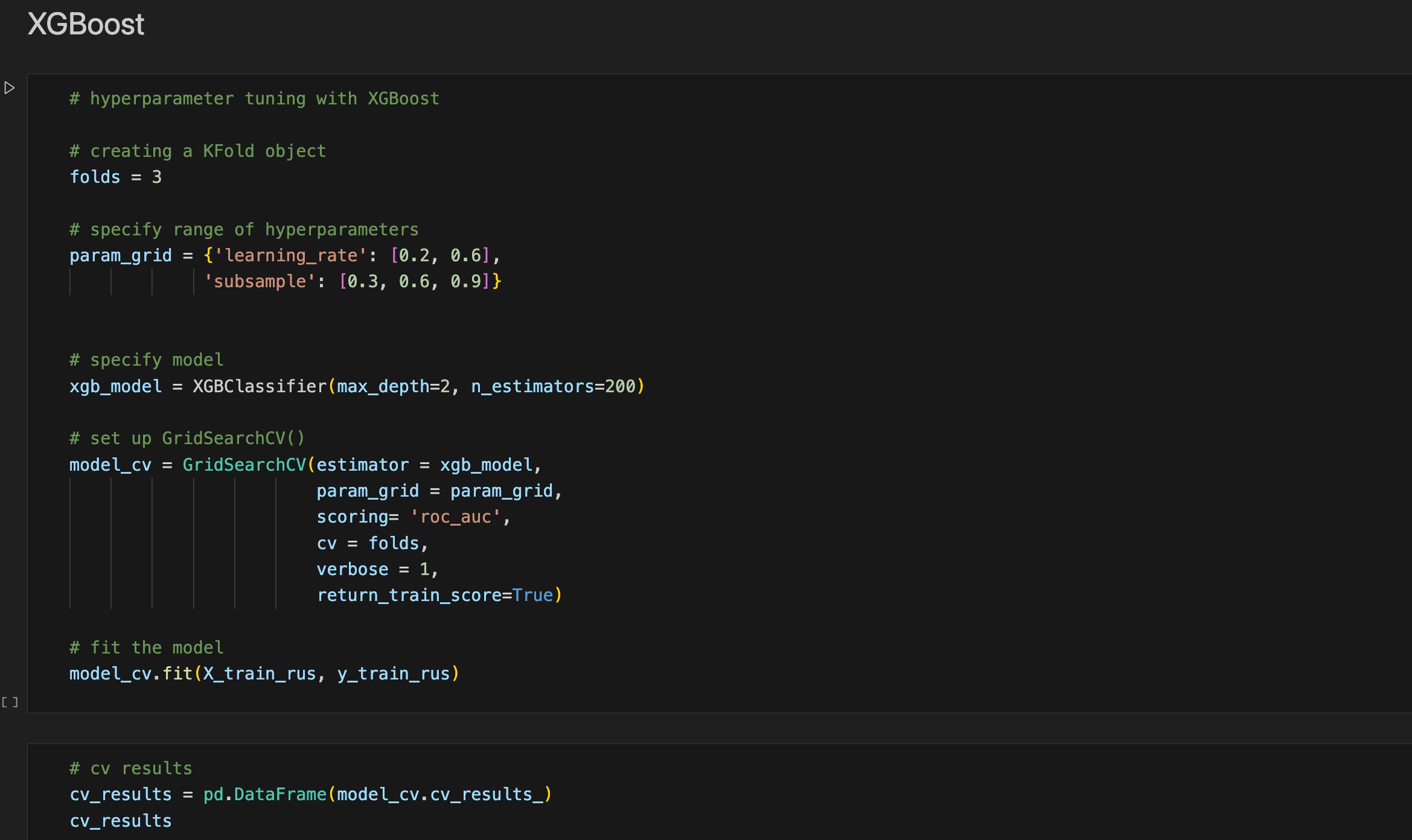


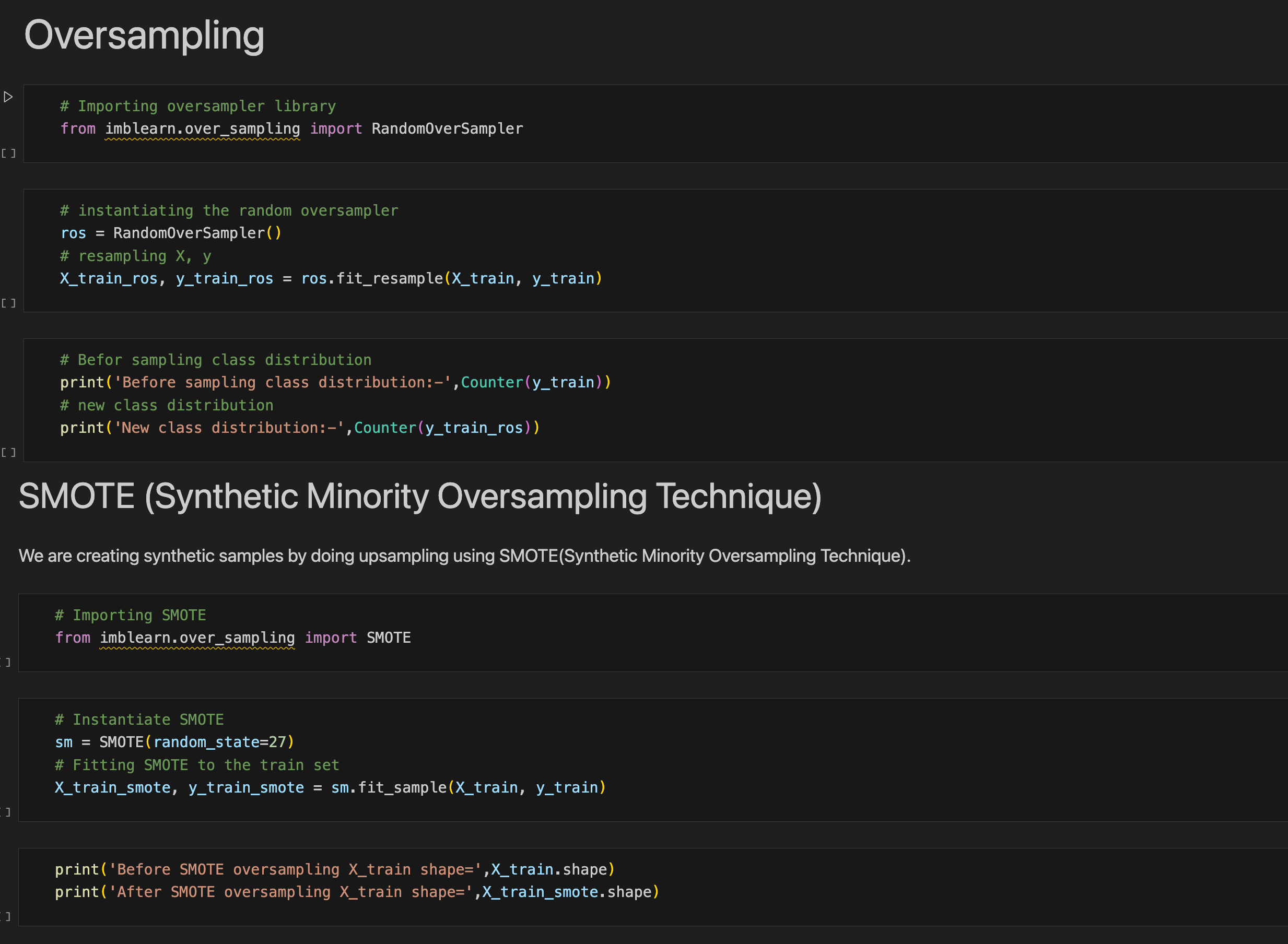


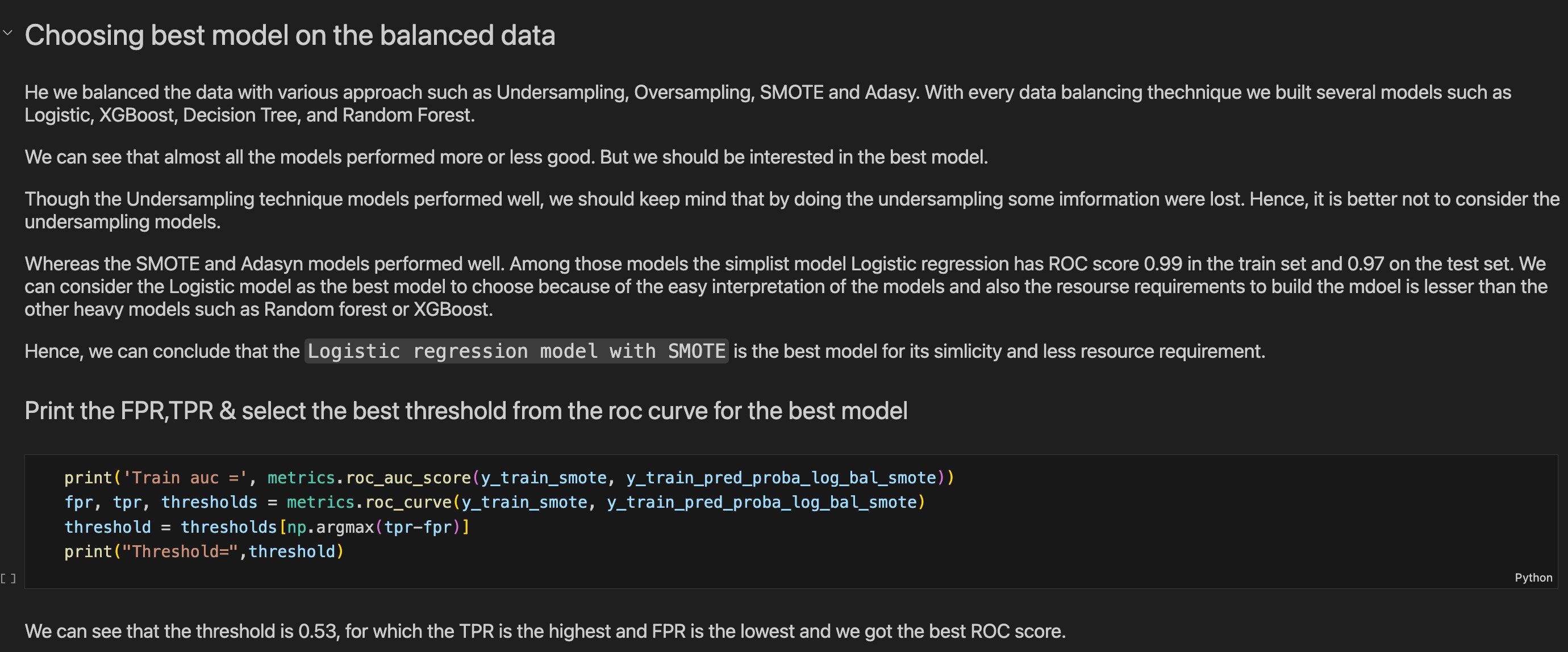












## **REFERENCES**

1. F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in *IEEE Access*, vol.10,
2. pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.
3. R. San Miguel Carrasco and M. -Á. Sicilia-Urbán, "Evaluation of Deep Neural Networks for Reduction of Credit Card Fraud Alerts," in IEEE Access, vol. 8, pp. 186421-186432, 2020, doi: 10.1109/ ACCESS.2020.3026222.
4. H. Tingfei, C. Guangquan and H. Kuihua, "Using Variational Auto Encoding in Credit Card Fraud Detection," in IEEE Access, vol. 8, pp. 149841-149853, 2020, doi: 10.1109/ACCESS.2020.3015600.
5. S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. -S. Hacid and H. Zeineddine, "An Experimental StudyWith Imbalanced Classification Approaches for Credit Card Fraud Detection," in IEEE Access, vol. 7,
6. pp.93010-93022, 2019, doi: 10.1109/ACCESS.2019.2927266.
7. [5]F. A. Ghaleb, F. Saeed, M. Al-Sarem, S. N. Qasem and T. Al-Hadhrami, "Ensemble Synthesized Minority Oversampling-Based Generative Adversarial Networks and Random Forest Algorithm for Credit Card Fraud Detection," in IEEE Access, vol. 11, pp. 89694-89710, 2023, doi: 10.1109/ACCESS.2023.3306621.
8. Daly, L. (2021, October 27). Identity theft and credit card fraud statistics for 2021: The ascent. The Motley Fool. Retrieved from https://[www.fool.com/the-](http://www.fool.com/the-) ascent/research/identity-theft-credit-card-fraud-statistics/
9. Dheepa, V., & Dhanapal, R. (2012). Behaviour-based credit card fraud detection using support vector machines. ICTACT Journal on Soft Computing, 02(04), 391–397. https://doi.org/10.21917/ijsc.2012.0061
10. Domínguez-Almendros, S., Benítez-Parejo, N., & Gonzalez-Ramirez, A. R. (2011). Logistic regression models. Allergologia et immunopathologia, 39(5), 295-305.
11. Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. D. (2019). Credit card fraud detection using machine learning and Data Science. Credit Card Fraud Detection Using Machine Learning and Data Science, 08(09). https://doi.org/10.17577/ijertv8is090031
12. Sahin, Y., & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. Proceedings of the International MultiConference of Engineers and Computer Scientists,
13. Zareapoor, M., Seeja. K.R., S. K. R., & Afshar Alam, M. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. International Journalof Computer Applications, 52(3), 35–42. https://doi.org/10.5120/8184-1538
14. Garcia, S., Grill, M., Stojanovic, A., & Osmotic, A. (2014). Anomaly detection in credit card transactions. In Machine Learning and Data Mining in Pattern Recognition (pp. 175-189). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-662-43898-5\_14
15. Aggarwal, C. C., & Yu, P. S. (2012). Outlier detection for high dimensional data. In Outlier Analysis (pp. 237-264). Springer, Boston, MA. doi: 10.1007/978-1-4614-6396-2\_10
16. Lu, Y., Ma, J., Ma, W., & Wong, K. C. (2015). Effective and efficient outlier detection in categorical data: A novel neighborhood-based approach. Information Sciences, 303, 130-145. doi: 10.1016/j.ins.2014.09.045
17. Gao, Y., & Huang, J. (2015). Credit card fraud detection using AdaBoost and decision trees. Expert Systems with Applications, 42(7), 3634-3642. doi: 10.1016/j.eswa.2014.12.026
18. Wang, J., & Wang, H. (2014). A fuzzy support vector machine approach for credit card fraud detection. In Proceedings of the 6th International Conference on Machine Learning and Computing (pp. 273-278). ACM. doi: 10.1145/2554849.2554861
19. Bhattacharyya, S., Jha, S., & Tharakunnel, K. (2013). An ensemble learning approach for credit card fraud detection. In Proceedings of the 2013 International Conference on Machine Learning and Applications (pp. 154-159). IEEE. doi: 10.1109/ICMLA.2013.183
20. Patel, V. M., Patil, B. V., & Naik, G. S. (2016). A novel approach for credit card fraud detection using artificial immune system. Procedia Computer Science, 79, 533-540. doi: 10.1016/j.procs.2016.03.068
21. Gunes, S., & Seker, S. E. (2015). An efficient credit card fraud detection technique based on multiclass classification. Applied Soft Computing, 27, 583-592. doi: 10.1016/j.asoc.2014.11.048
22. Özkan, M. (2016). Credit card fraud detection using hidden Markov model and neural networks. Expert Systems with Applications, 55, 233-243. doi: 10.1016/j.eswa.2016.02.001
23. Shen, F., Shen, C., Shi, X., & Sun, J. (2015). Credit card fraud detection using a profile-based approach. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 943-950). ACM. doi: 10.1145/2808797.2809423