

COGNITIVE ANALYTICS LAB

Submitted by: Submitted to:

Wajid Ali Hashmi Sugandha Sharma

500096923

B2(H)

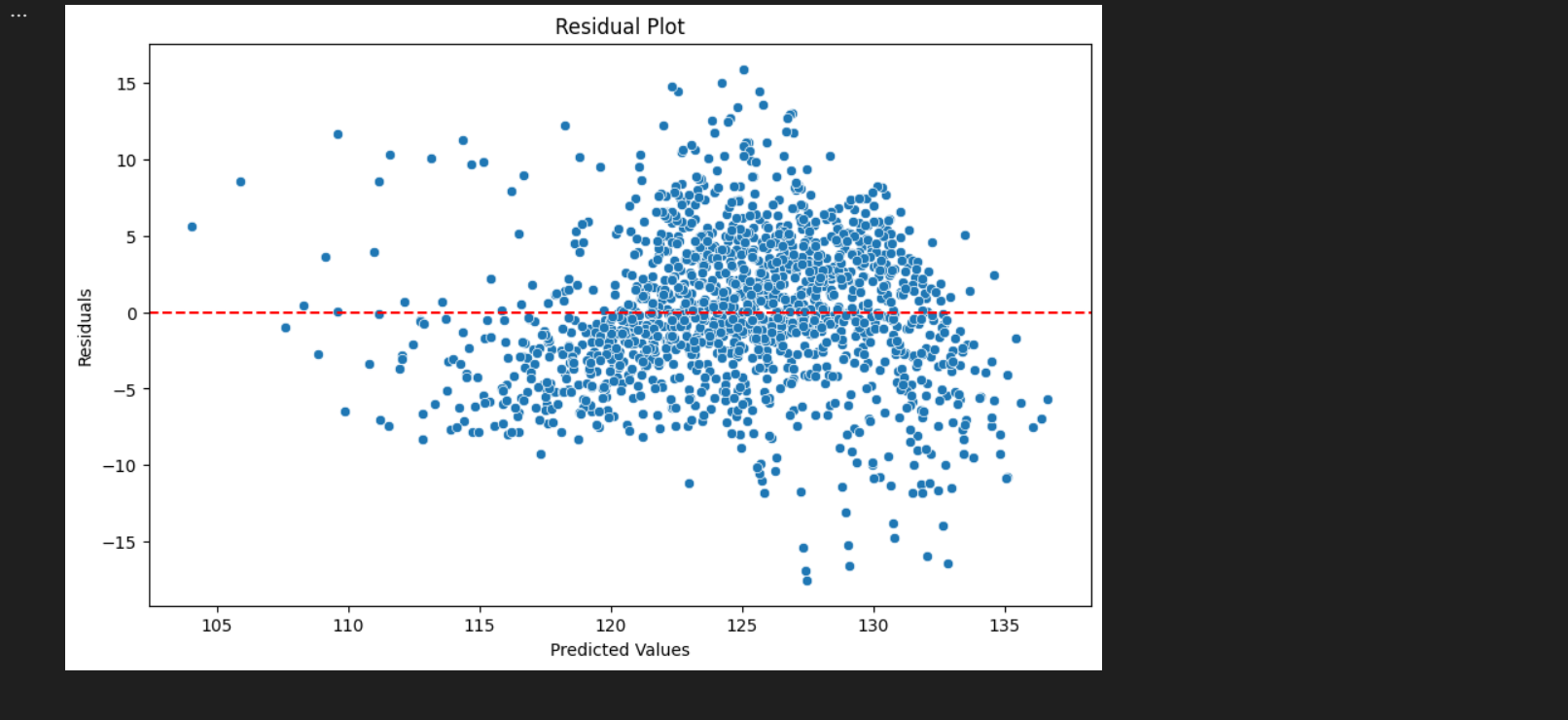
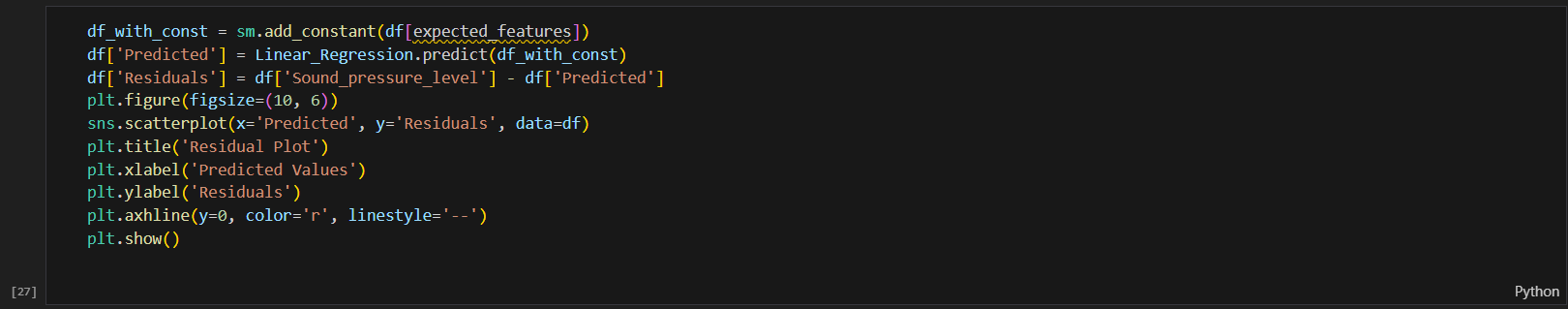
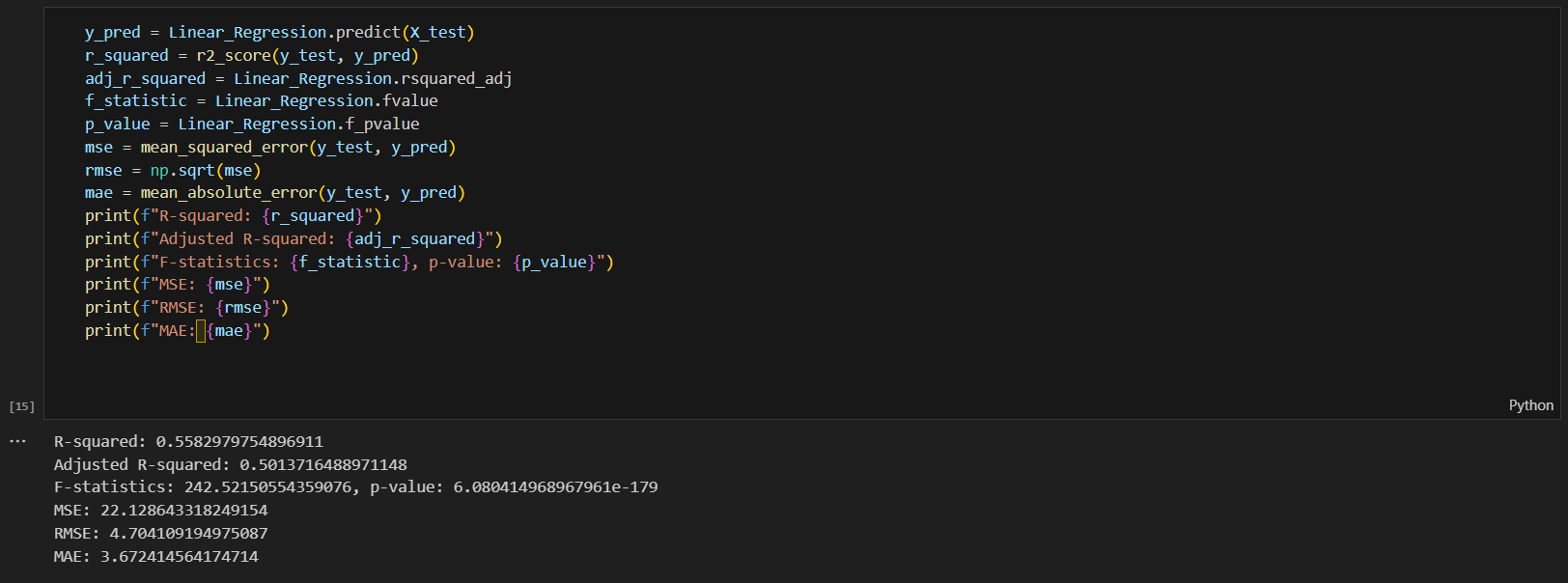
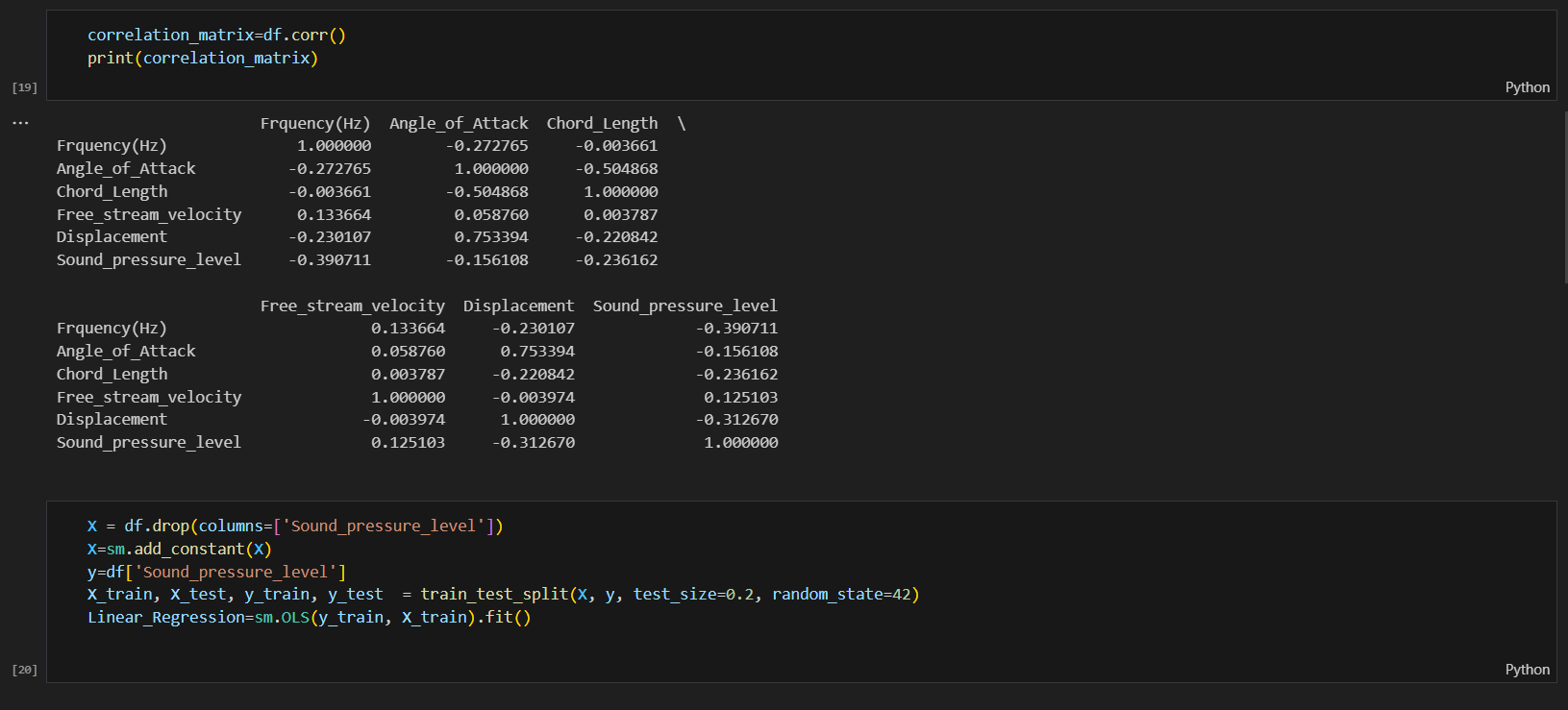
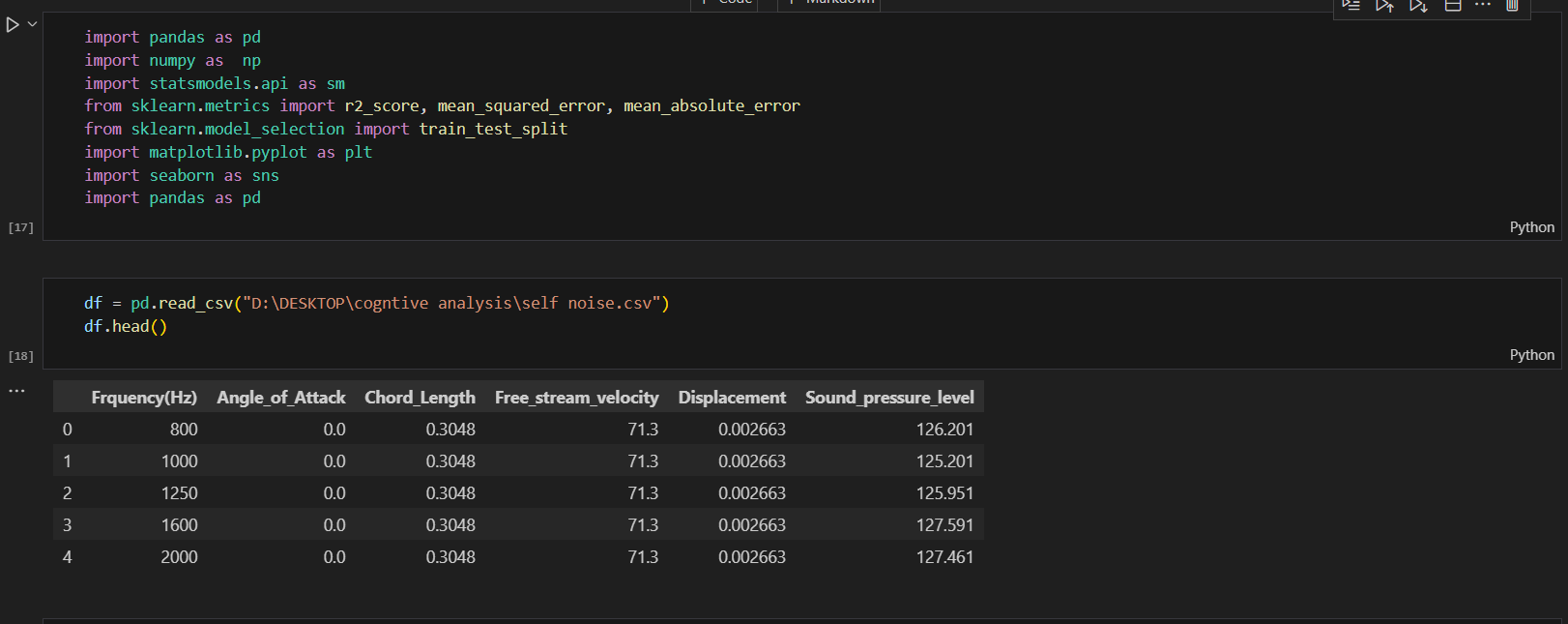
Index

|  |  |  |  |
| --- | --- | --- | --- |
| Sno. | Name | Date | Page |
| 1. | Analysing the dataset | 17-01-2024 | 3 |
| 2. | Ridge and lasso algorithm | 24-01-2024 | 6 |
| 3. | Logistic and Polynomial Regression | 31-01-2024 | 11 |
| 4. | Implementing Decision Tree on diabetes dataset | 14-02-2024 | 15 |
| 5. | Implementing Random Forest | 21-02-2024 | 17 |
| 6. | Implementing Gradient Boosting | 20-03-2024 | 20 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Experiment 1

Analysing the dataset based on some values

Dataset source: <http://blog.hackerearth.com/wp-content/uploads/2016/12/airfoil_self_noise.csv>



**Interpretations:**

Based on the analysis, the following can be deduced about the dataset:

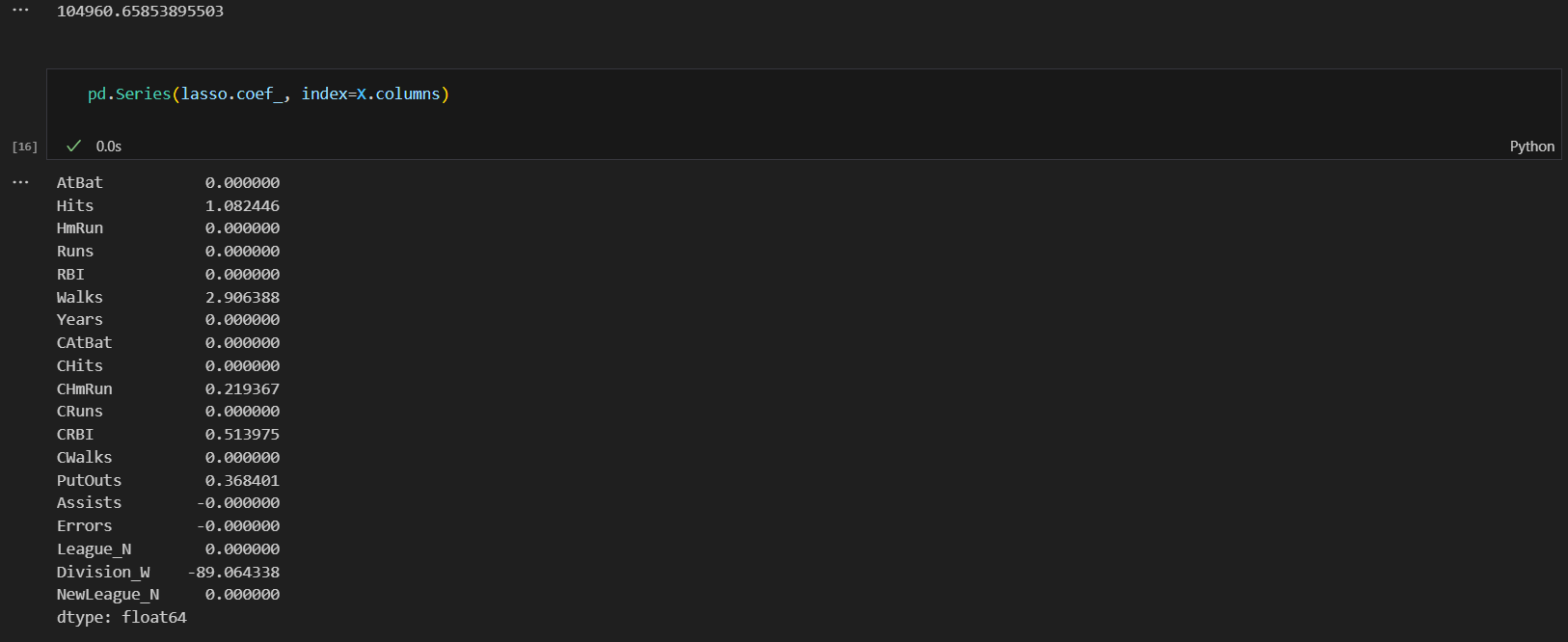
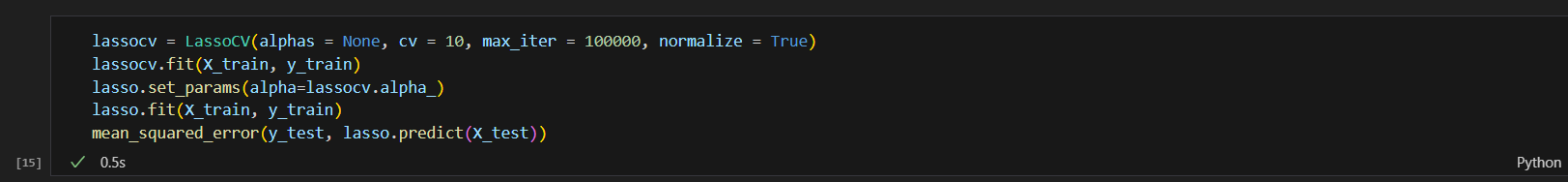
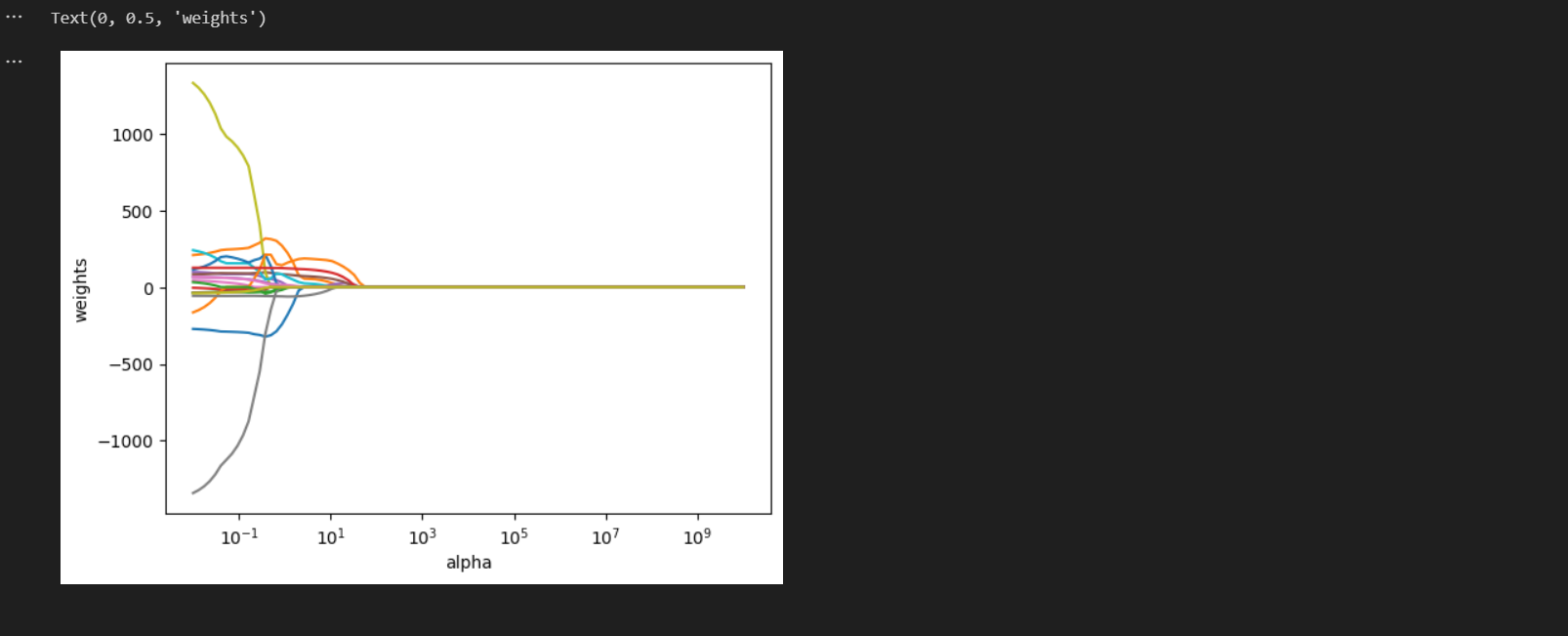
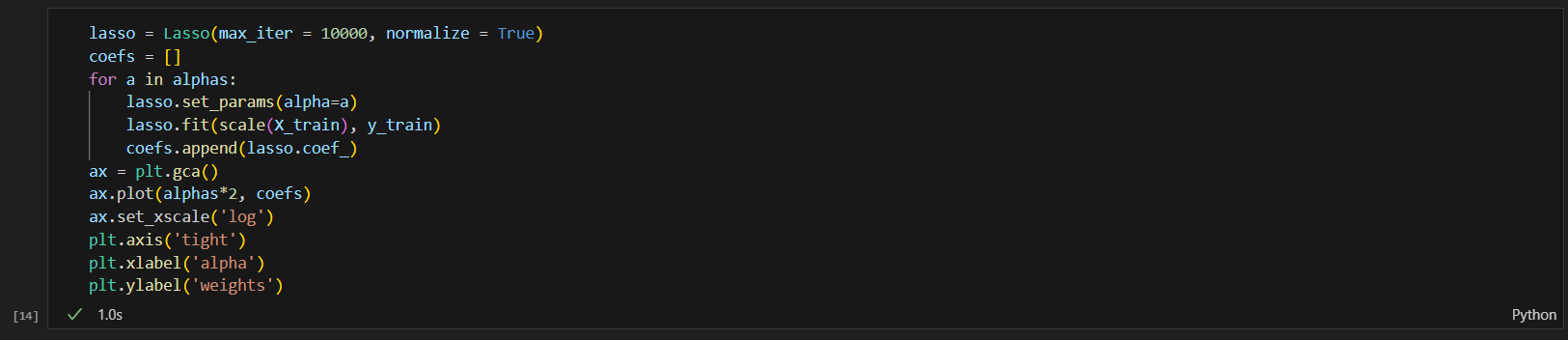
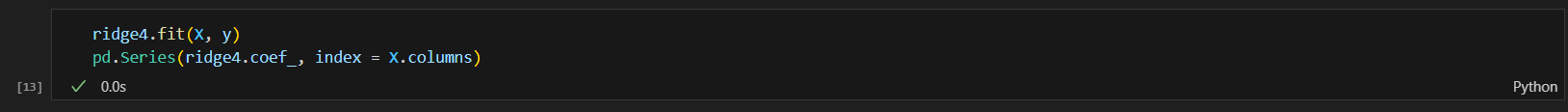
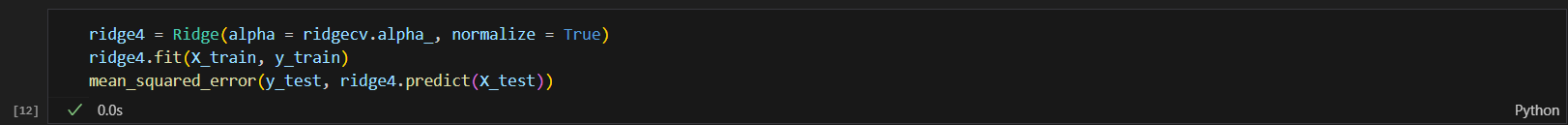
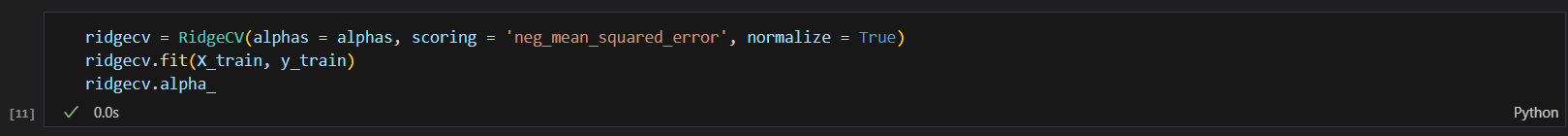
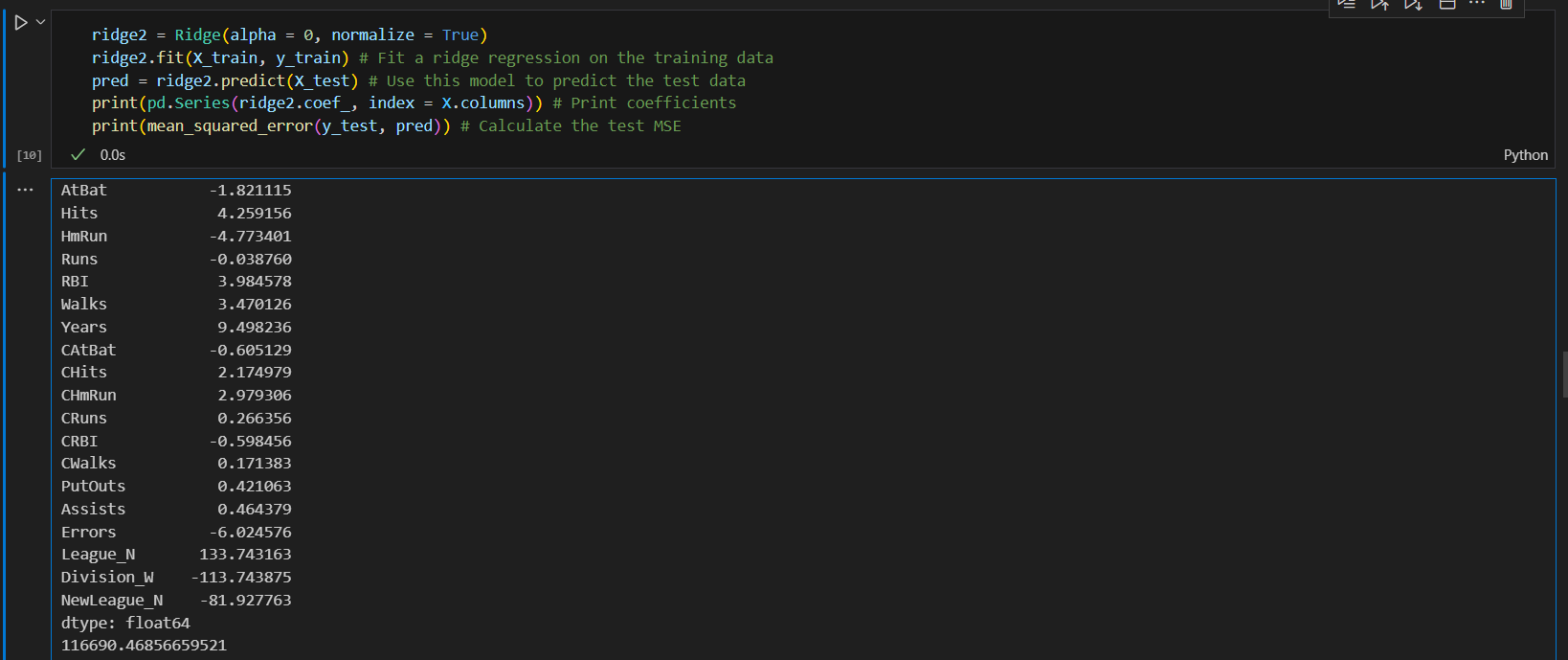
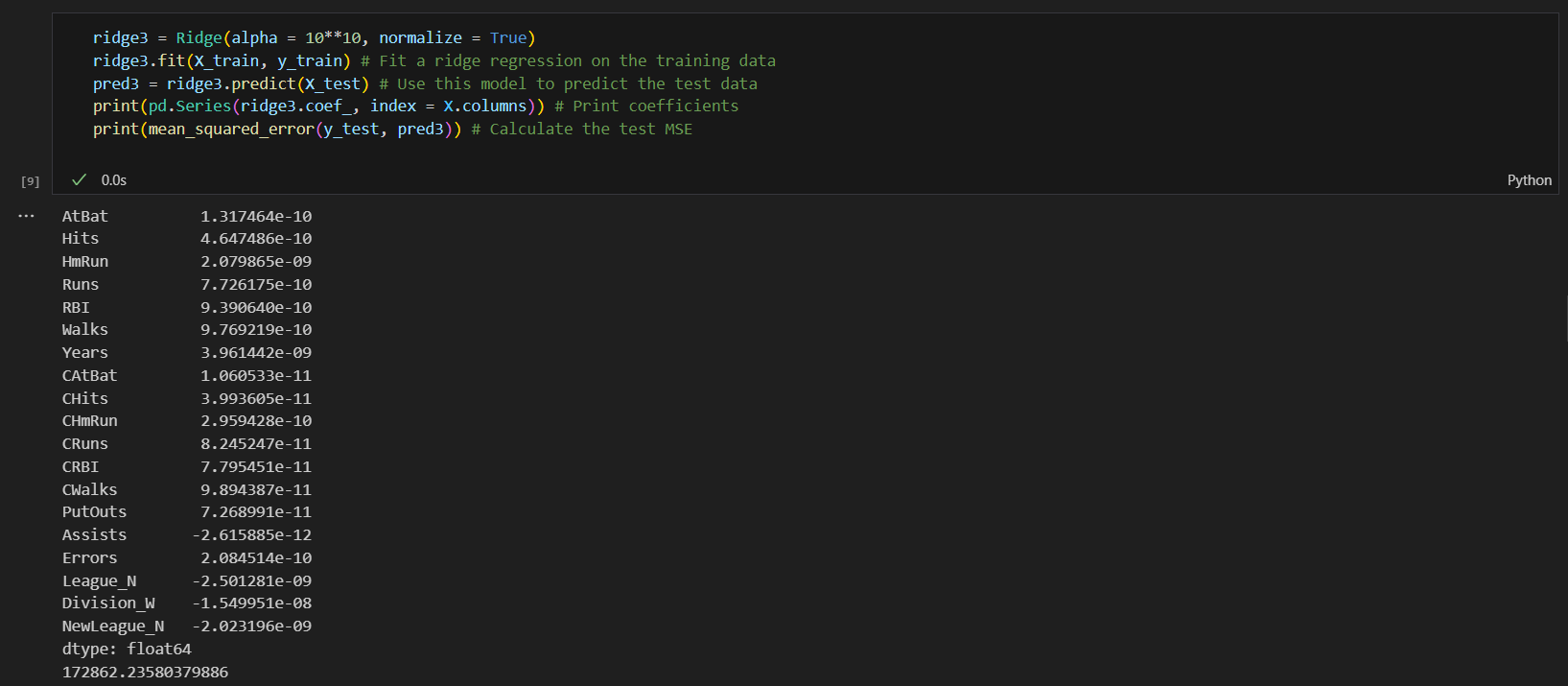
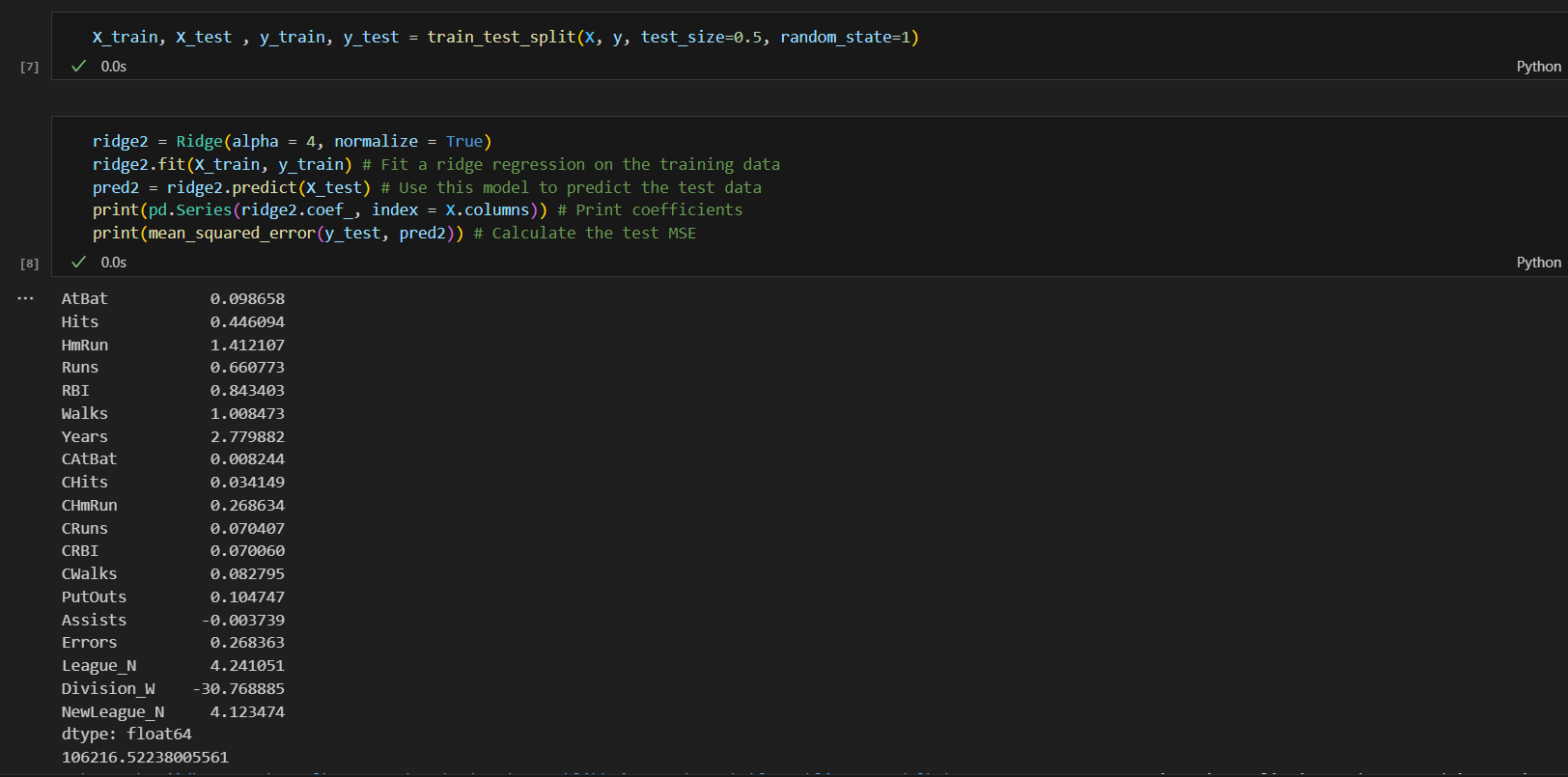
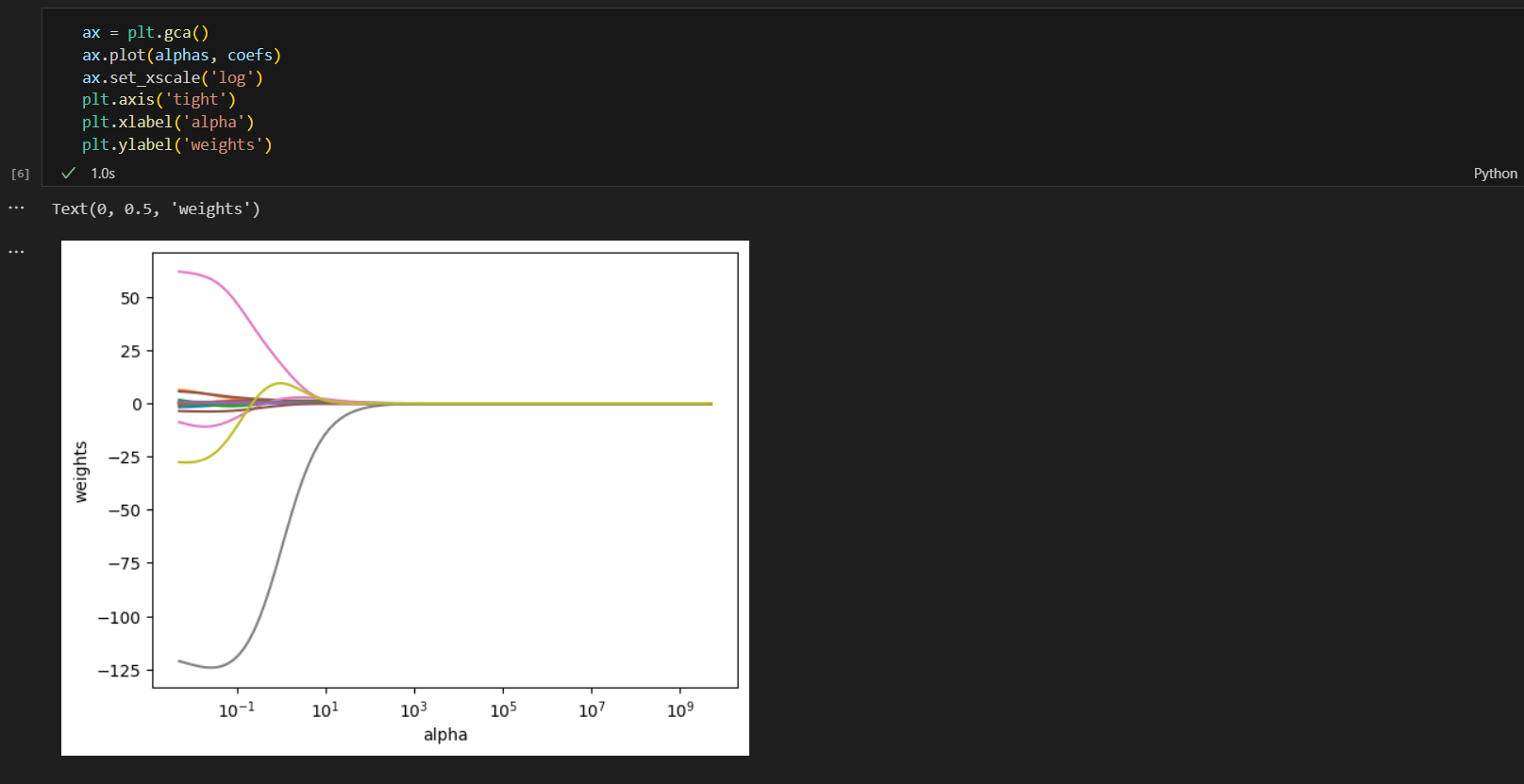
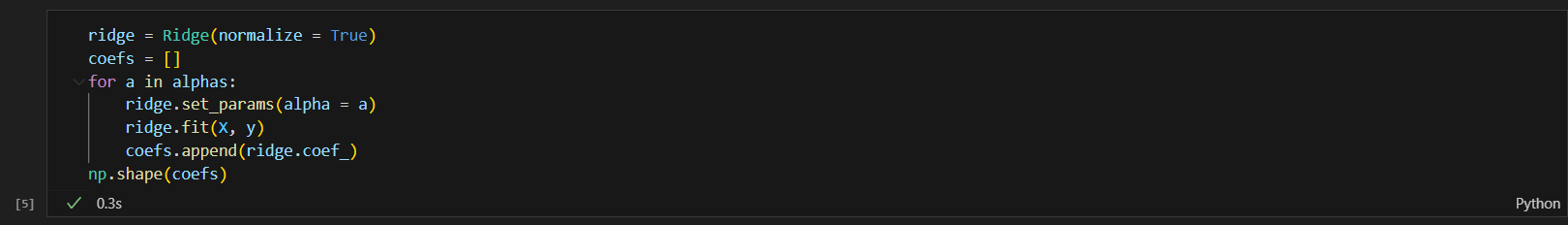
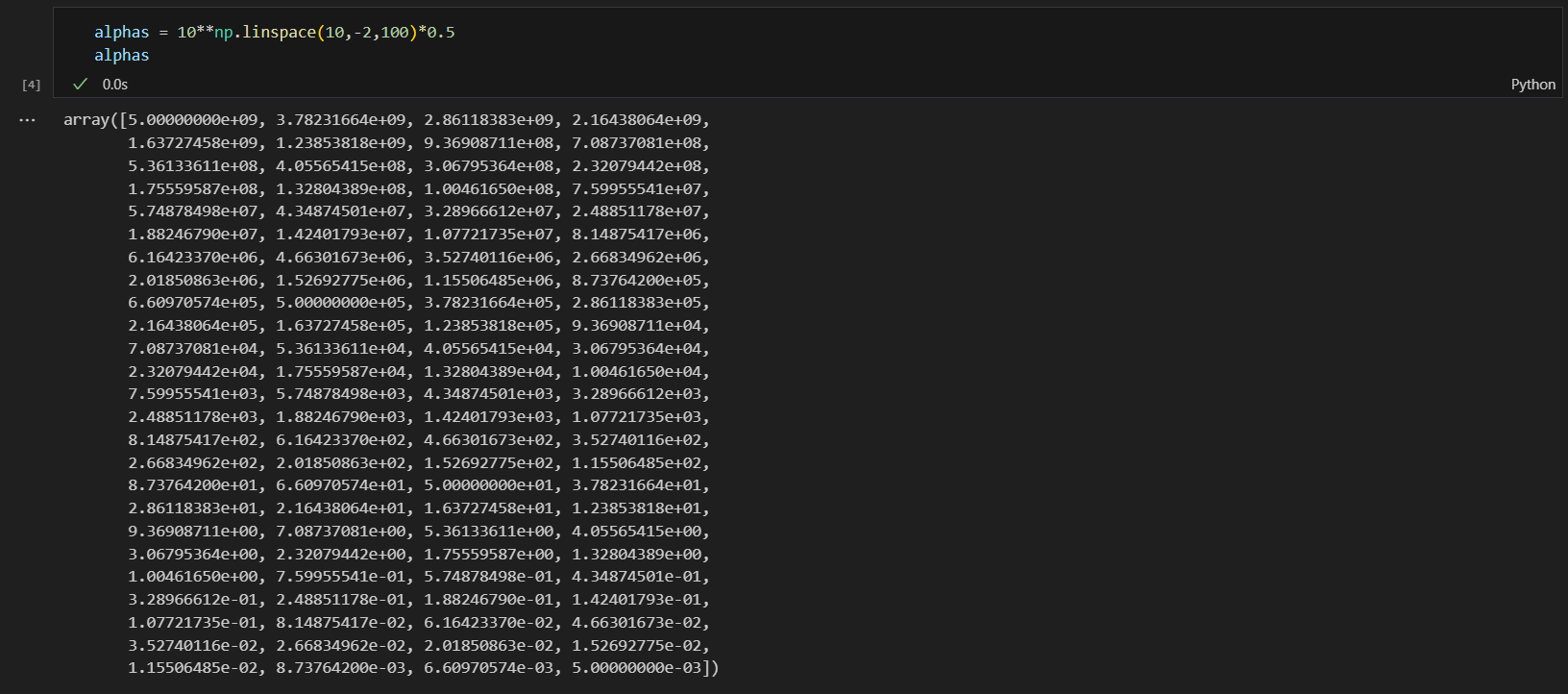
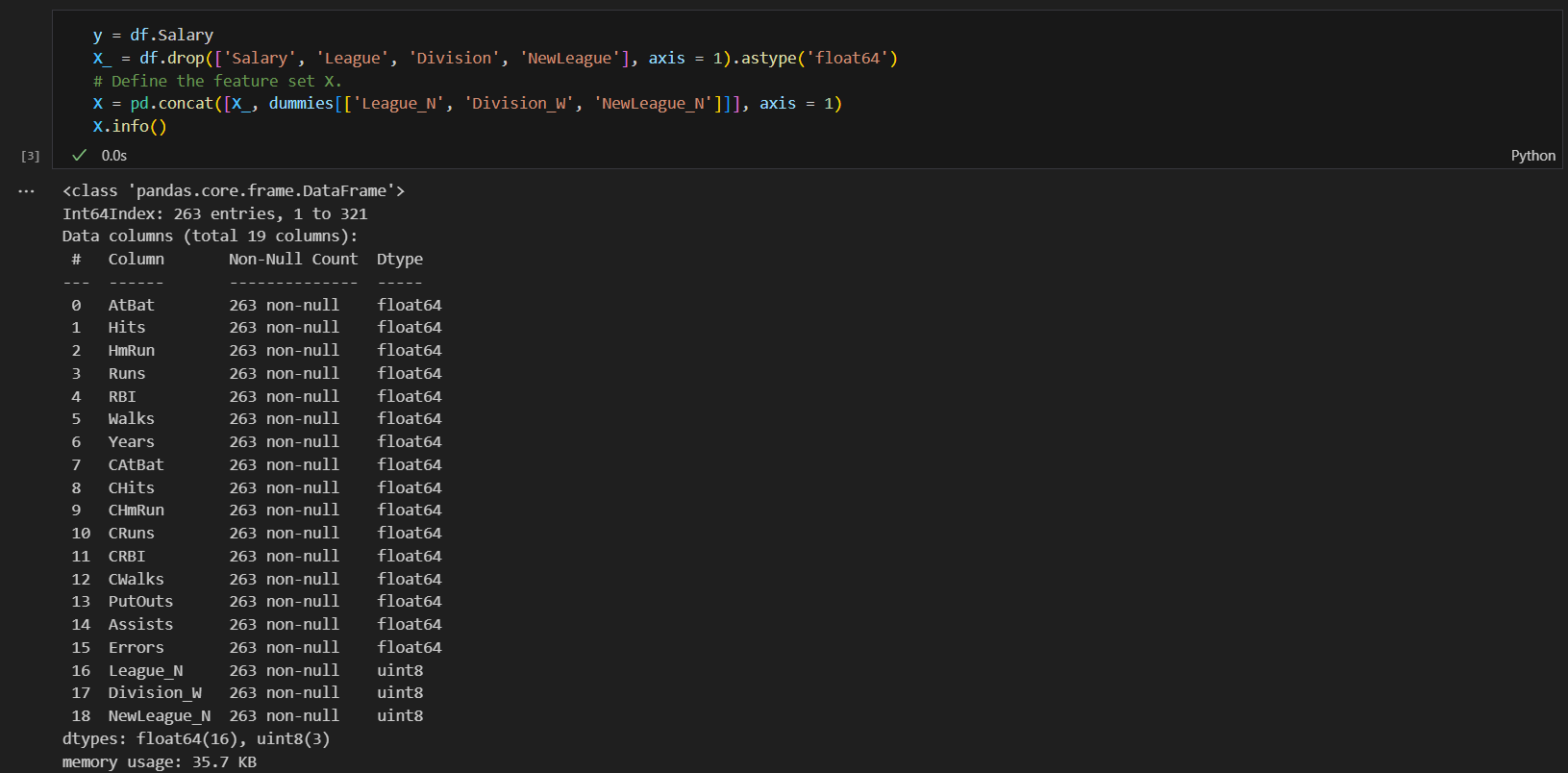
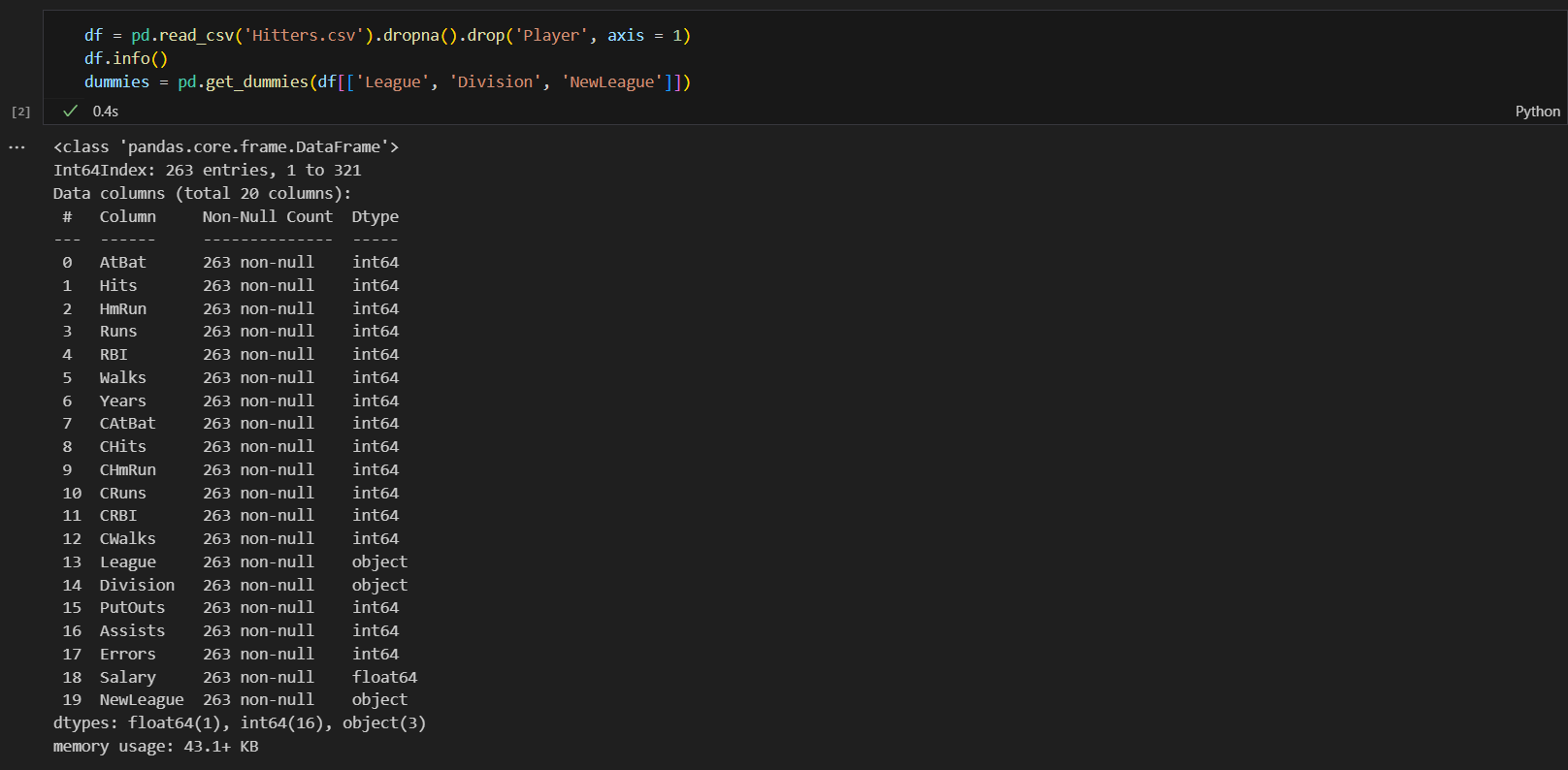
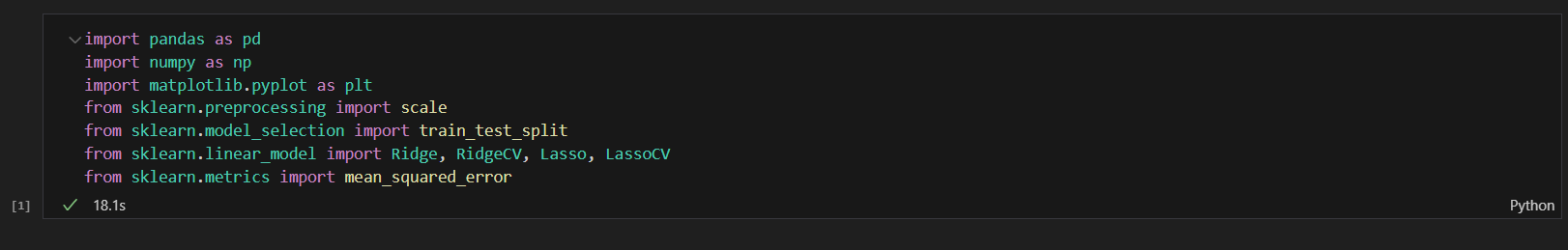
* The R-squared value of 0.558 indicates that the linear regression model explains 55.8% of the variance in the sound pressure level. This is a good fit for a linear model.
* The adjusted R-squared value of 0.501 is slightly lower than the R-squared value, but it still indicates a good fit for the model.
* The F-statistic of 242.52 and the p-value of less than 0.0001 both indicate that the model is statistically significant. This means that the coefficients of the model are not equal to zero.
* The MSE of 22.12indicates that the average squared difference between the predicted and actual sound pressure levels is 22.12.
* The RMSE of 4.704 is the square root of the MSE, and it indicates that the average difference between the predicted and actual sound pressure levels is 4.704.
* The MAE of 3.672 indicates that the average absolute difference between the predicted and actual sound pressure levels is 3.672.
* The residual plot shows that the residuals are randomly scattered around the zero line. This indicates that the model does not have any heteroscedasticity or autocorrelation problems.
* The coefficients of the model can be used to identify which features have the most impact on the sound pressure level.
* The model can be used to make predictions about the sound pressure level for new data points.
* The model can be used to identify outliers in the data.

In conclusion, the linear regression model is a good fit for the data and can be used to predict the sound pressure level based on the other features in the dataset.

Experiment 2

Implementation of Ridge and lasso algorithm in Python:

Dataset source: <https://github.com/JWarmenhoven/ISLR-python/blob/master/Notebooks/Data/Hitters.csv>



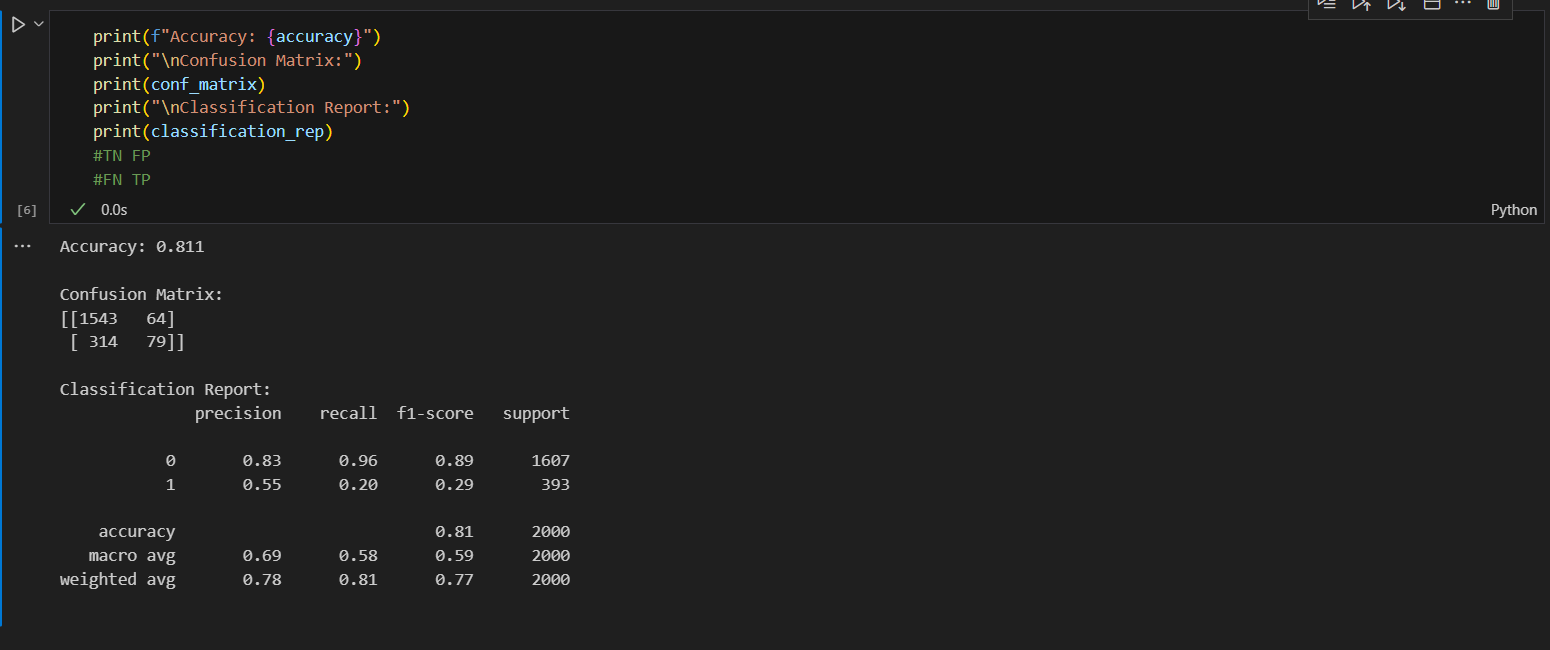
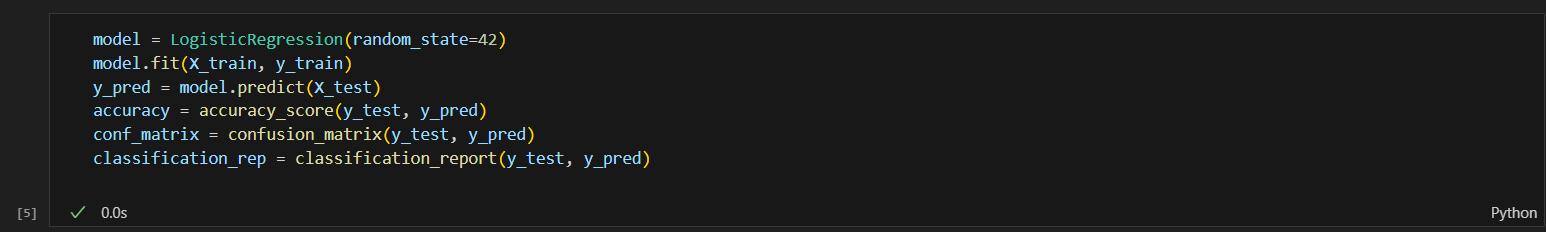
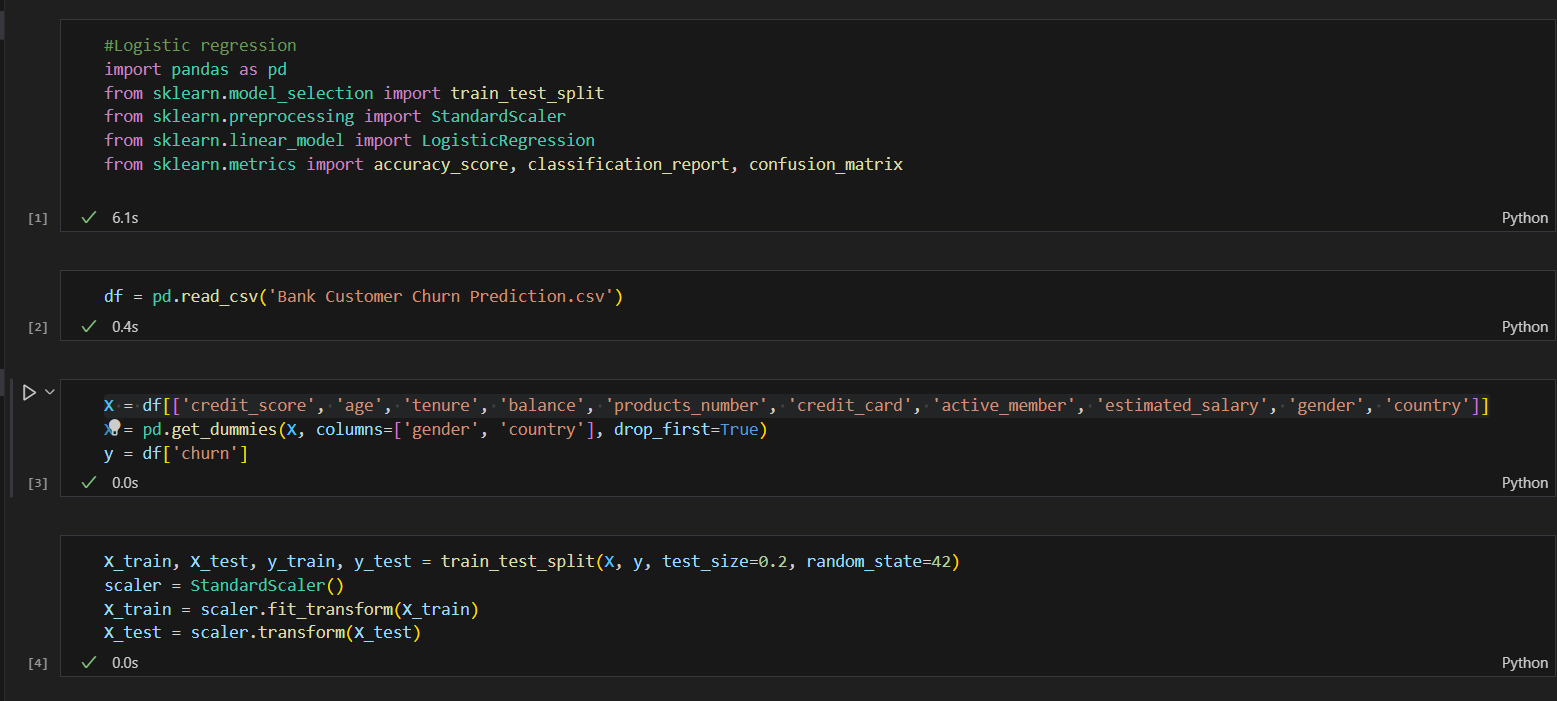
**Interpretation:**

* Ridge and lasso regression are regularization techniques used to prevent overfitting in linear regression models.
* The choice of alpha determines the strength of regularization, with higher alpha leading to more regularization.
  + As we can see from the output of the code, when the value of alpha is small the output values are significantly small but when the alpha is increased the output values decreases with higher rate.
* The code assesses model performance using mean squared error on the test set for both ridge and lasso regression.
* Cross-validation is employed to find optimal alpha values for ridge and lasso, enhancing model generalization.

Experiment 3

Logistic Regression:

Dataset URL: <https://www.kaggle.com/datasets/gauravtopre/bank-customer-churn-dataset>



**Interpretations on the basis of the values:**

Accuracy:

Accuracy is the ratio of correctly predicted instances to the total instances. It is a general measure of the model's overall correctness. In this case, an accuracy of 0.811 (or 81.1%) means that the model correctly predicted the churn status for approximately 81.1% of the instances in the dataset.

Confusion Matrix:

A confusion matrix is a table that describes the performance of a classification model. It consists of four terms: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). In your case:

* True Positive (TP): 79 instances where the model correctly predicted churn (1).
* True Negative (TN): 1543 instances where the model correctly predicted no churn (0).
* False Positive (FP): 64 instances where the model incorrectly predicted churn when there was no churn.
* False Negative (FN): 314 instances where the model incorrectly predicted no churn when there was churn.

Classification Report:

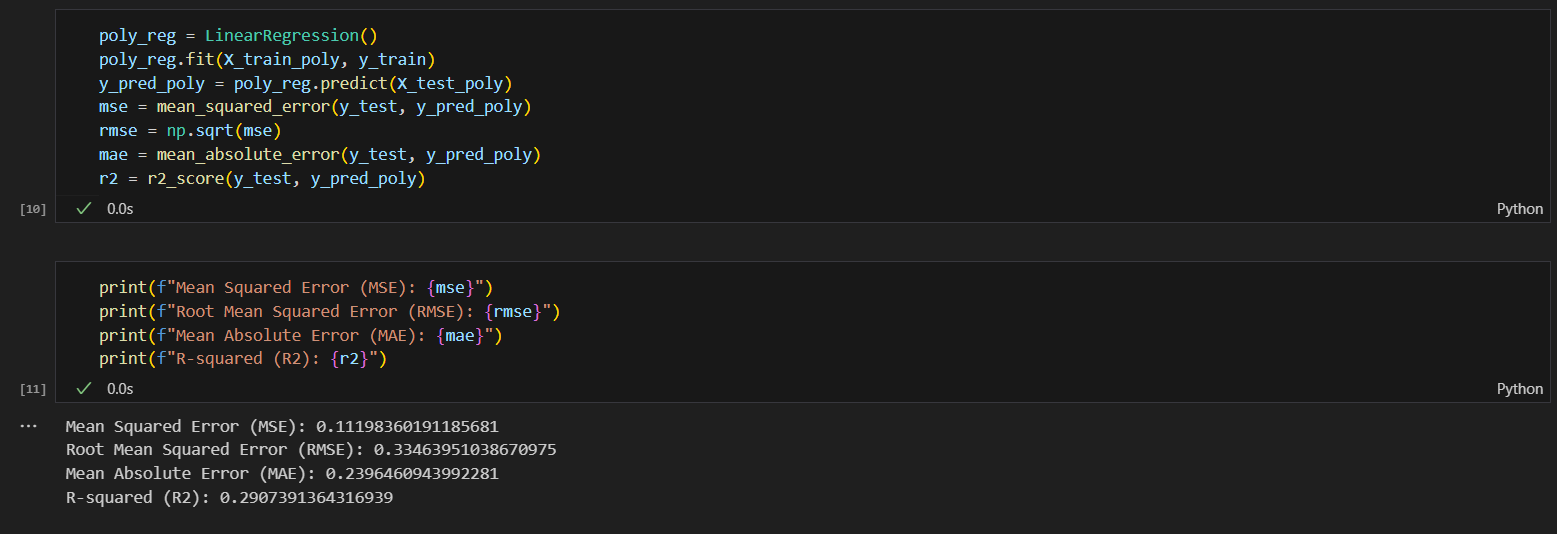
Precision, recall, and F1-score are metrics that provide more insights, especially in imbalanced datasets.

* Precision:
  + Precision is the ratio of correctly predicted positive observations to the total predicted positives.
  + Precision = TP / (TP + FP)
  + In this case, the precision for class 1 (churn) is approximately 0.55, indicating that 55% of the instances predicted as churn were actually churn.
* Recall (Sensitivity):
* Recall is the ratio of correctly predicted positive observations to the total actual positives.
* Recall = TP / (TP + FN)
* In this case, the recall for class 1 (churn) is approximately 0.20, indicating that the model correctly identified 20% of the actual churn instances.
* F1-score:
  + F1-score is the weighted average of precision and recall.
  + F1-score = 2 \* (Precision \* Recall) / (Precision + Recall)

It balances precision and recall, providing a single metric that considers both false positives and false negatives.

The weighted average of the F1-scores for both classes is given in the report.

* Support:
* Support is the number of actual occurrences of each class in the specified dataset.
* Macro Average and Weighted Average:
* Macro average calculates the unweighted average of precision, recall, and F1-score for all classes.
* Weighted average considers the number of instances for each class, providing more weight to the majority class.

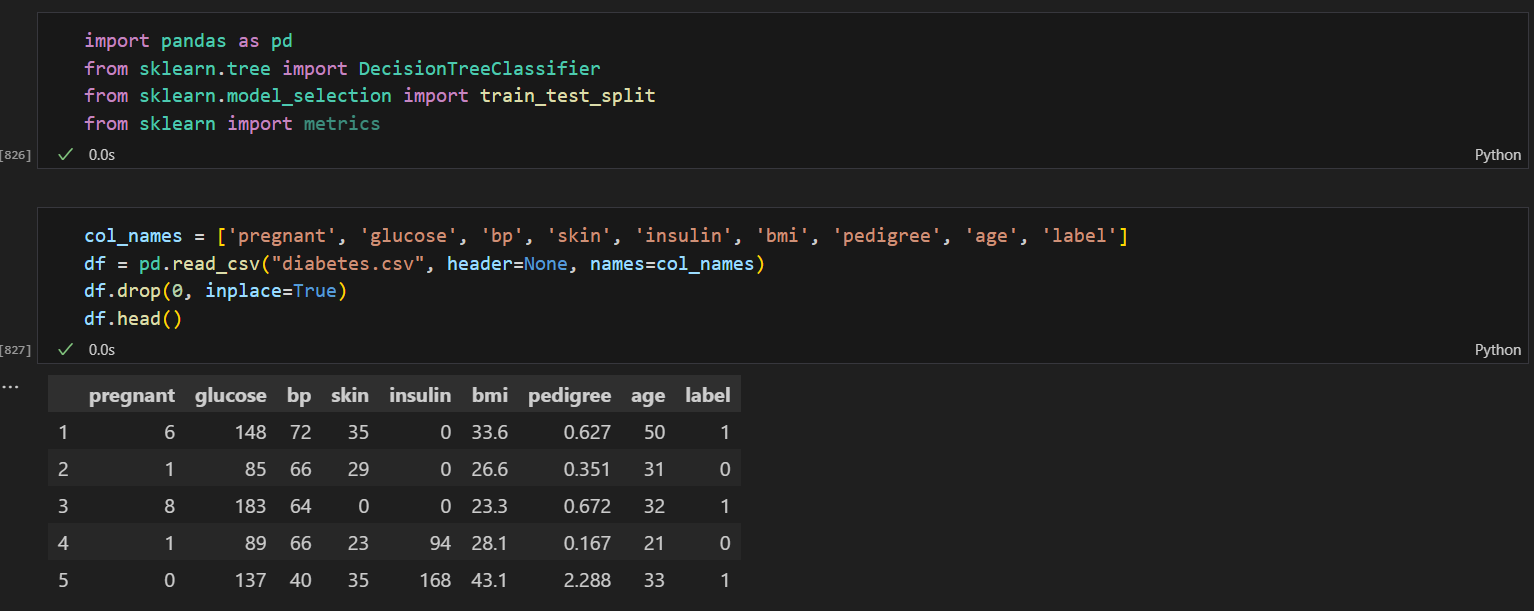
Polynomial Regression:

* Mean Squared Error (MSE):
* MSE measures the average squared difference between the actual and predicted values.
* A lower MSE indicates better model performance.
* In this case, an MSE of approximately 0.112 means, on average, the squared difference between the actual and predicted churn values is 0.112.
* Root Mean Squared Error (RMSE):
* RMSE is the square root of the MSE and provides a measure of the average absolute error.
* It is expressed in the same units as the target variable.
* In this case, an RMSE of approximately 0.335 means, on average, the absolute difference between the actual and predicted churn values is around 0.335.
* Mean Absolute Error (MAE):
* MAE is the average absolute difference between the actual and predicted values.
* It is less sensitive to outliers compared to MSE.
* In this case, an MAE of approximately 0.240 means, on average, the absolute difference between the actual and predicted churn values is around 0.240.
* R-squared (R2):
* R-squared measures the proportion of the variance in the dependent variable (churn) that is predictable from the independent variables (features).
* R-squared values range from 0 to 1, where 1 indicates a perfect fit.
* In this case, an R-squared of approximately 0.291 means that around 29.1% of the variance in churn can be explained by the features in the model.

Experiment 4

Implementing Decision Tree on diabetes dataset:

Dataset URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>



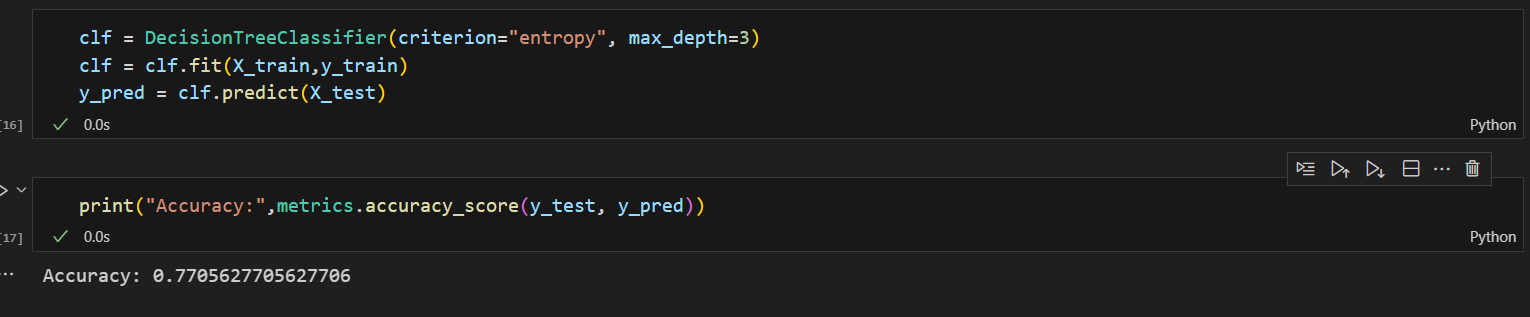


Interpretations:

In this code we can see the following things:

* The value 0.7056277056277056 represents the accuracy score, which is a metric used to evaluate the performance of a machine learning model.
* The higher the accuracy score, the better the model is at making correct predictions.

Enhancing the accuracy of the decision tree:

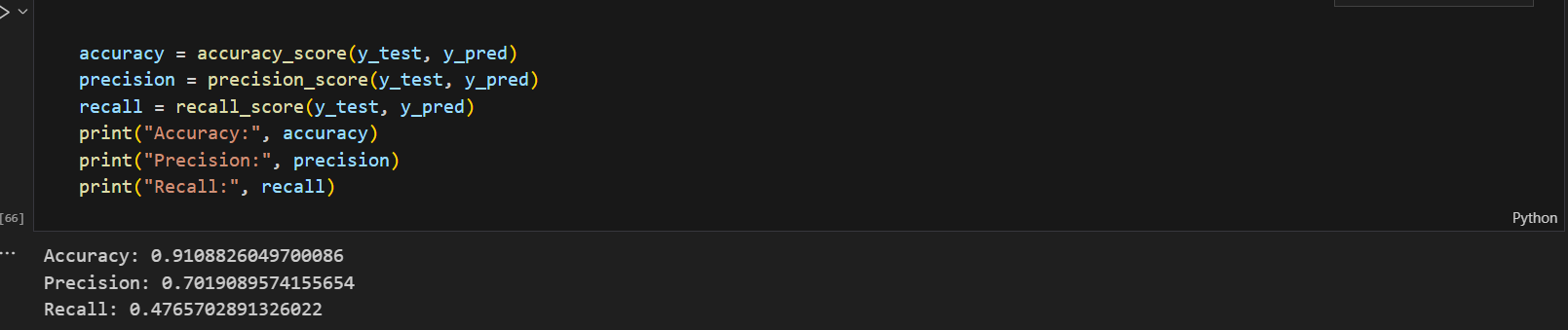
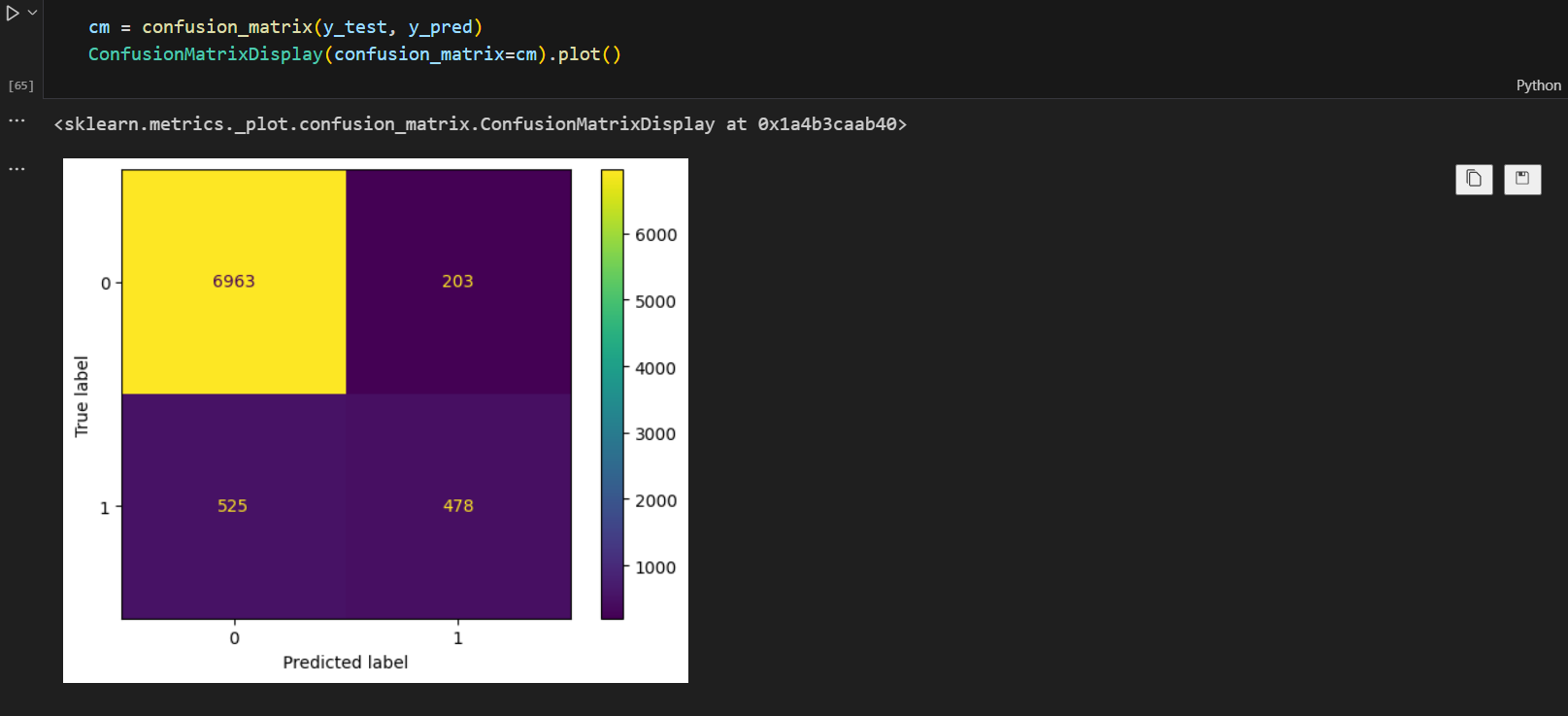
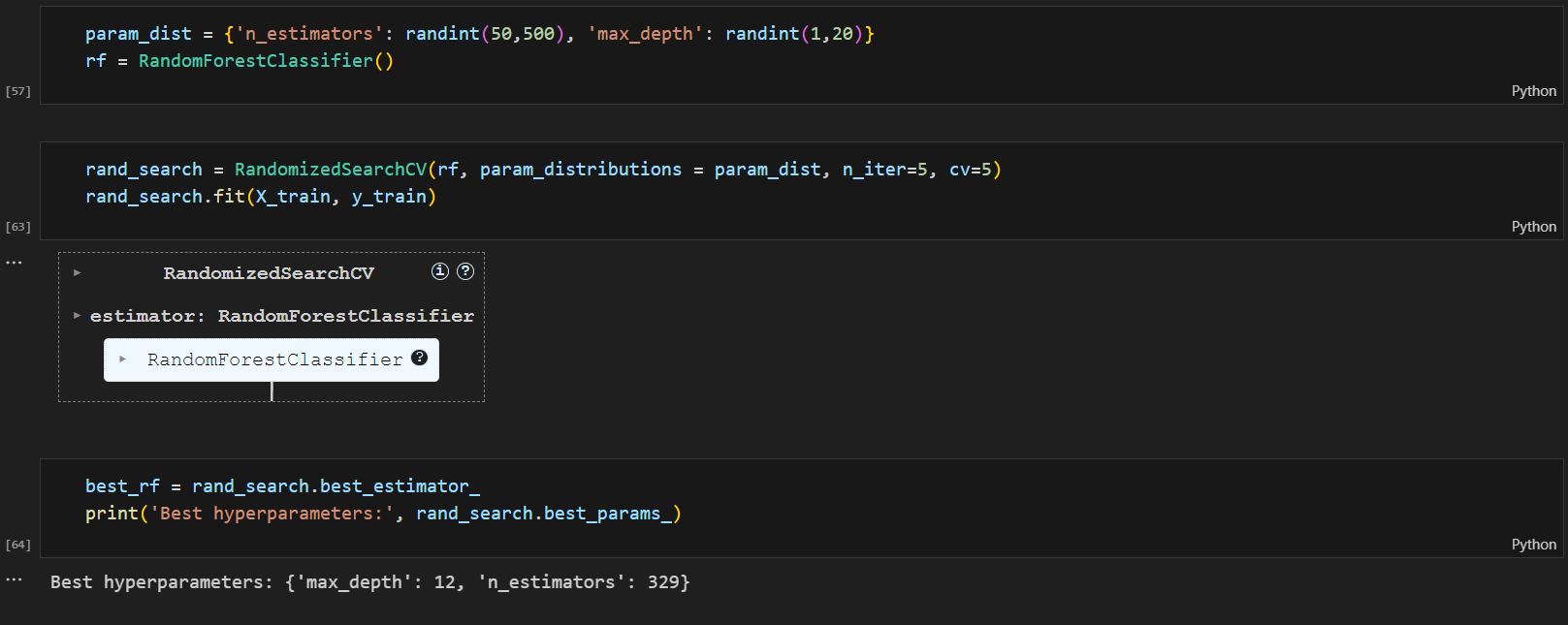
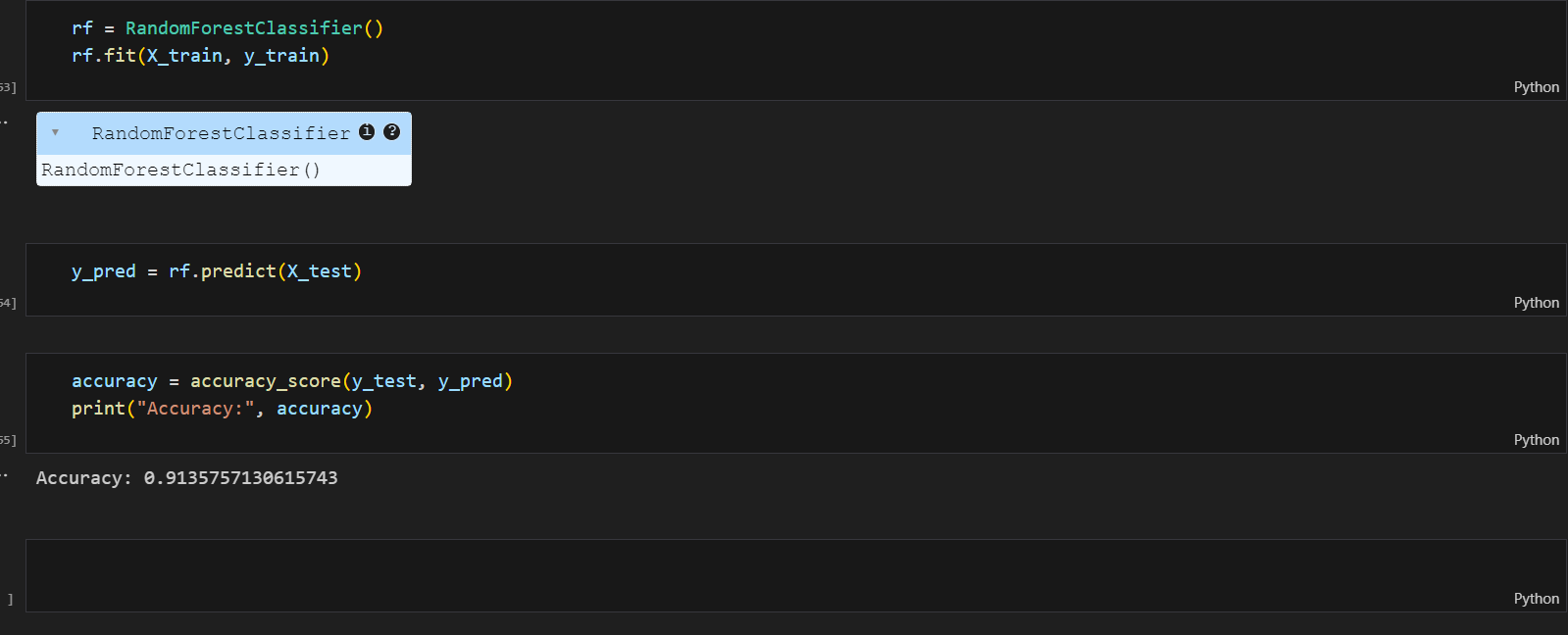
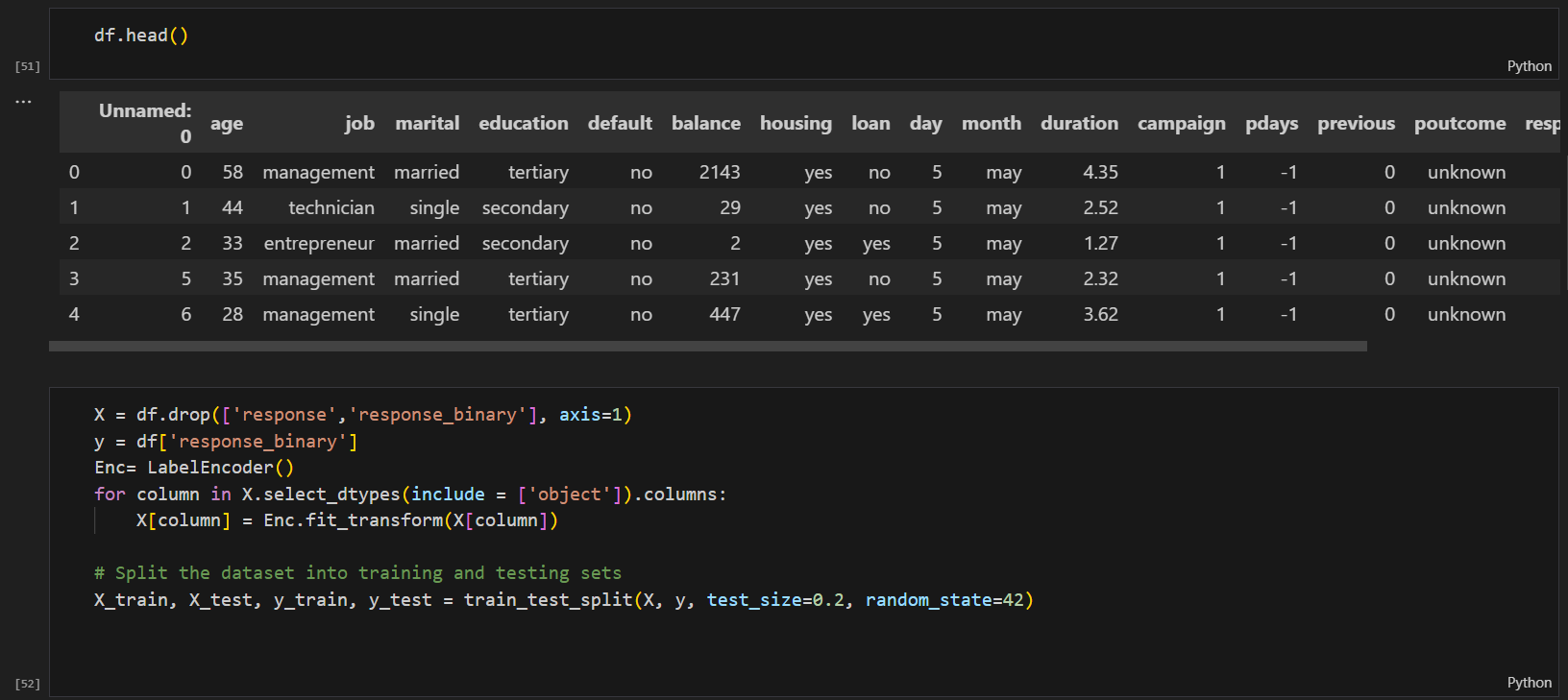
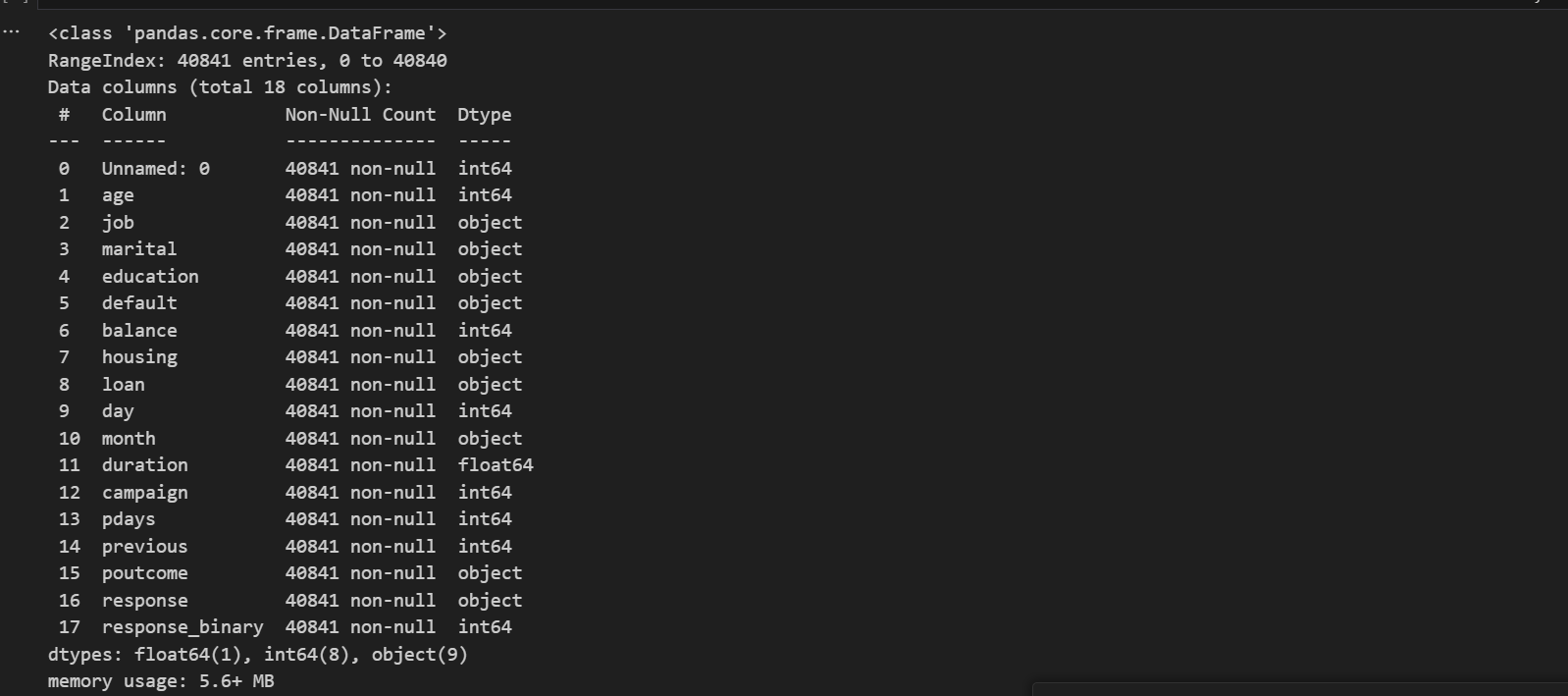
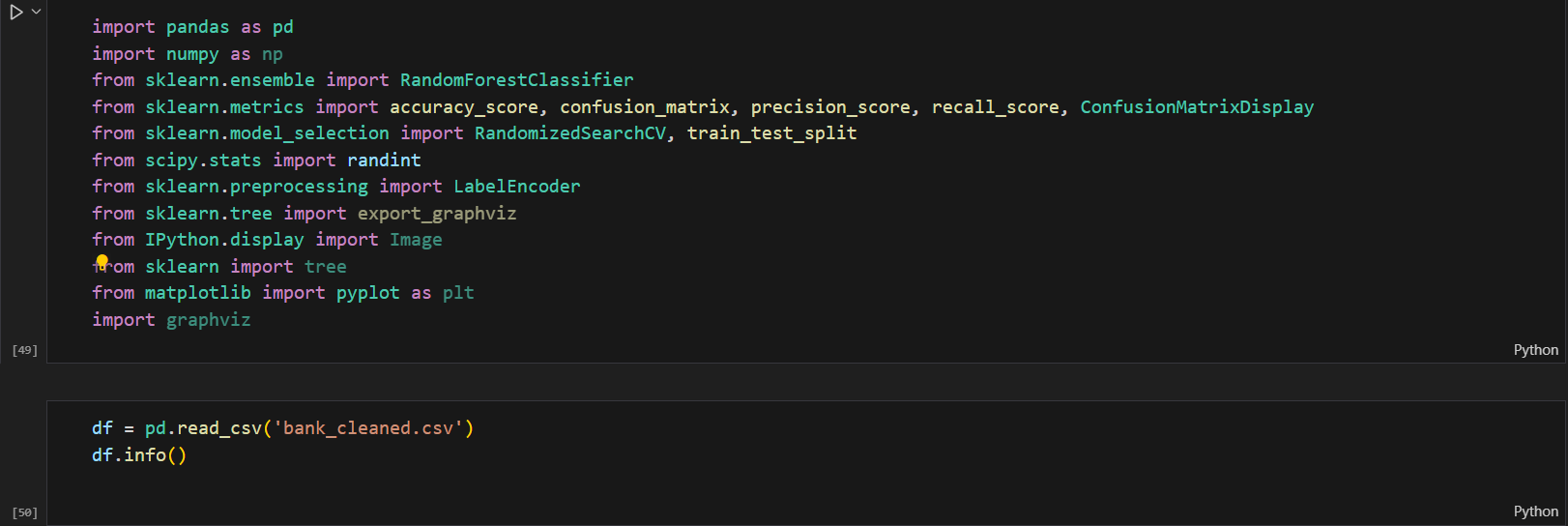


* Here we can see that the accuracy of the model increases when we ass the criterion of ‘entropy’ and ‘max\_depth’
* The higher the accuracy, the better the model is at making predictions.

Experiment 5

Implementing Random Forest:

Dataset URL: <https://www.kaggle.com/datasets/yufengsui/portuguese-bank-marketing-data-set>



Interpretation over the metrics calculated:

The confusion matrix shows the performance of a random forest model on a classification task. The rows represent the actual labels, and the columns represent the predicted labels. Each cell shows the number of instances in a particular category. In this case, the model is trying to predict whether a customer will churn or not (represented by 0 and 1).

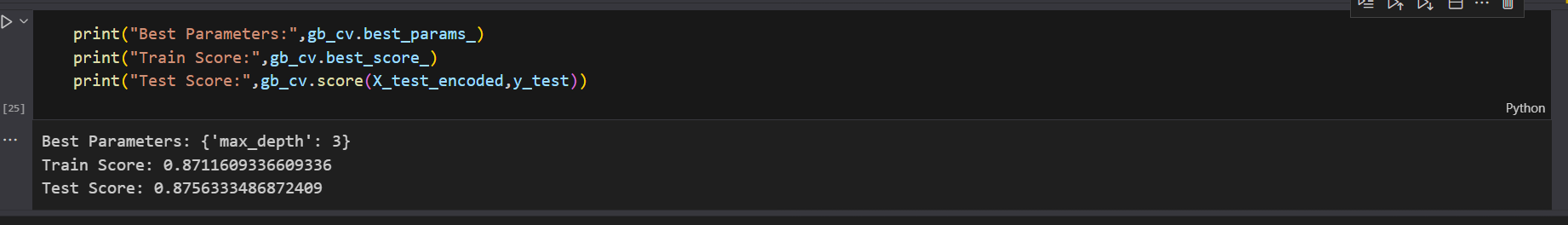
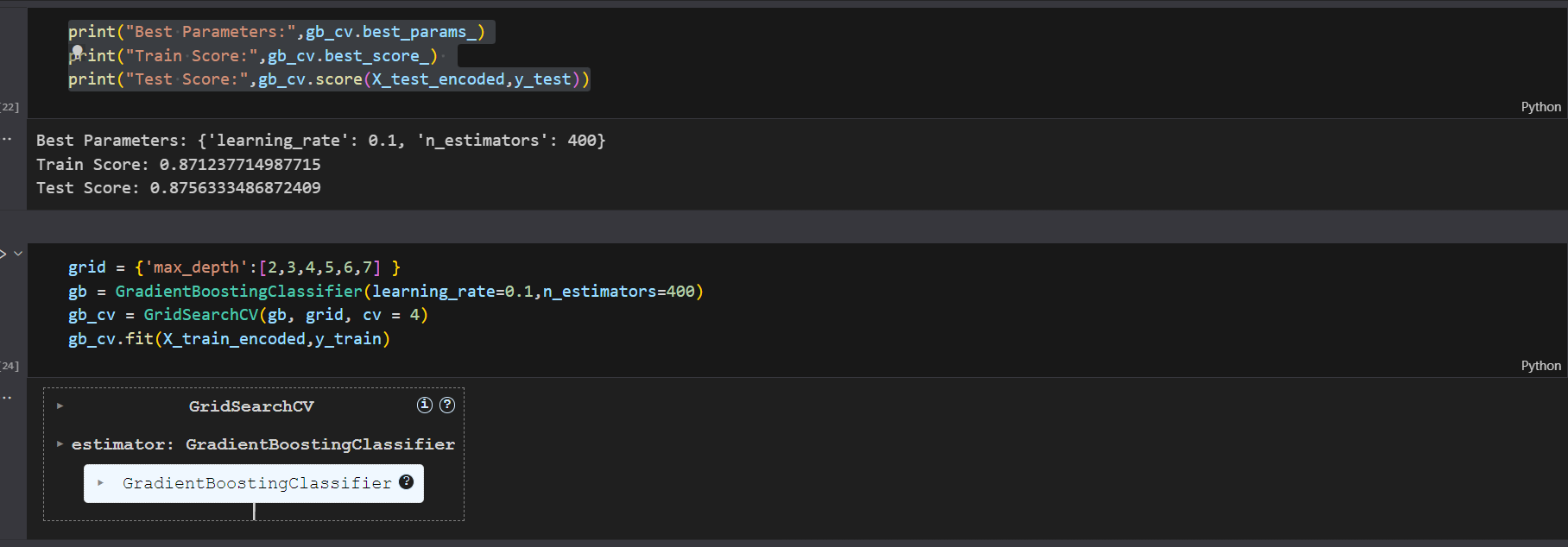
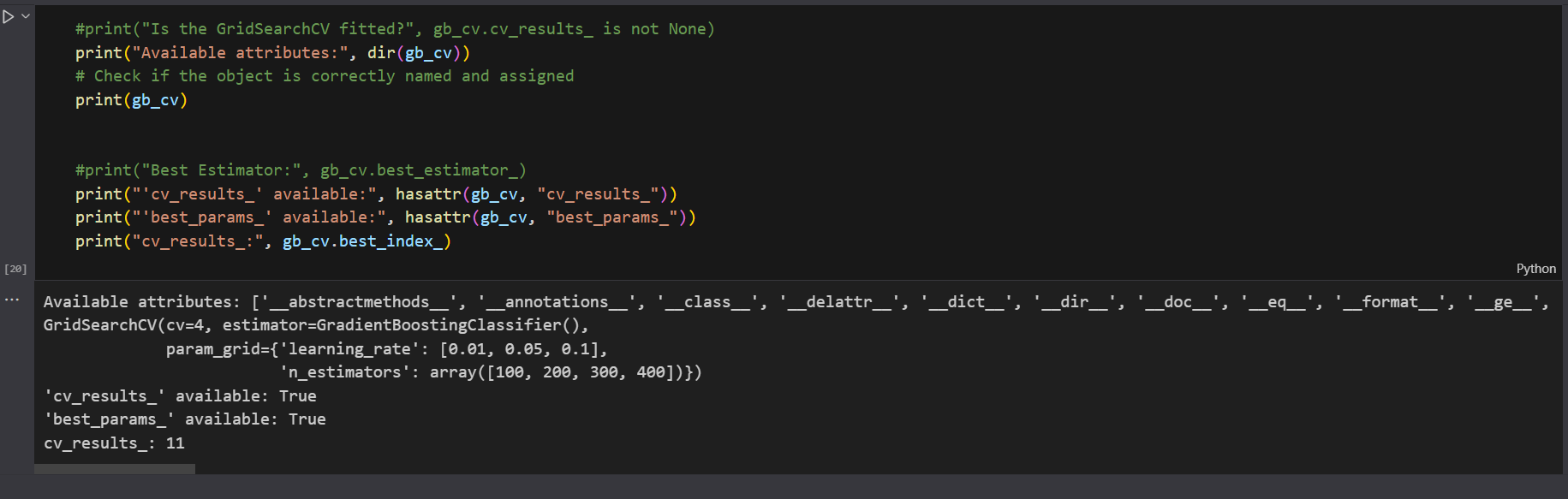
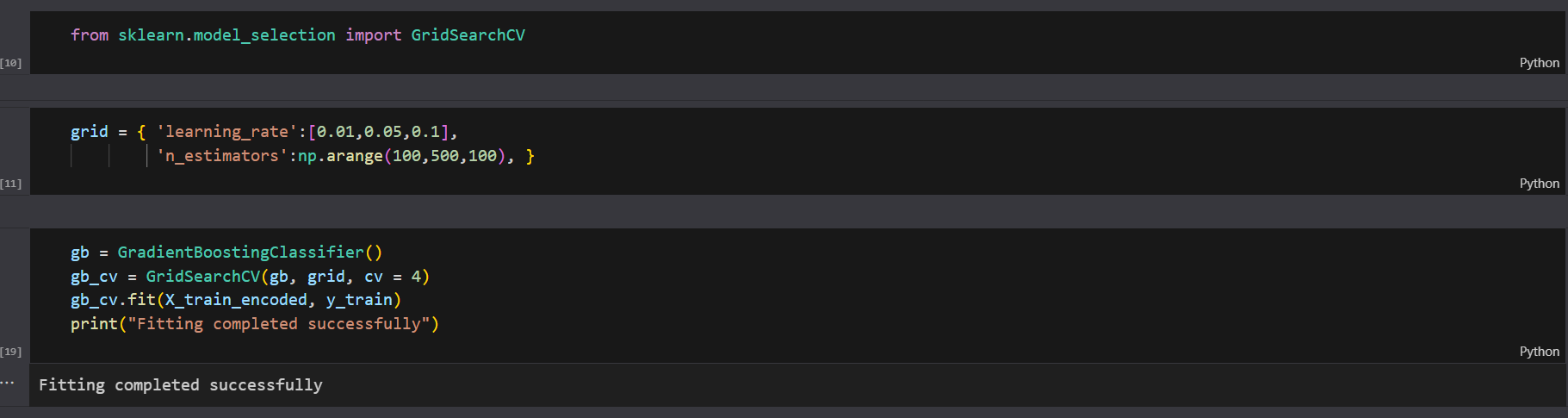
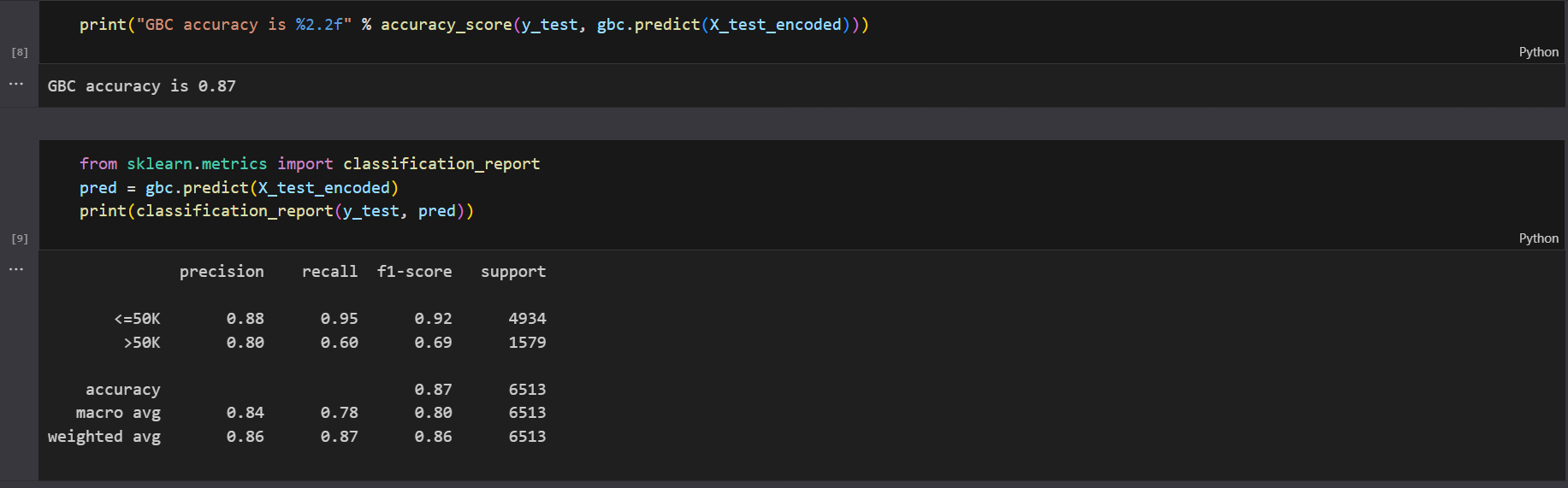
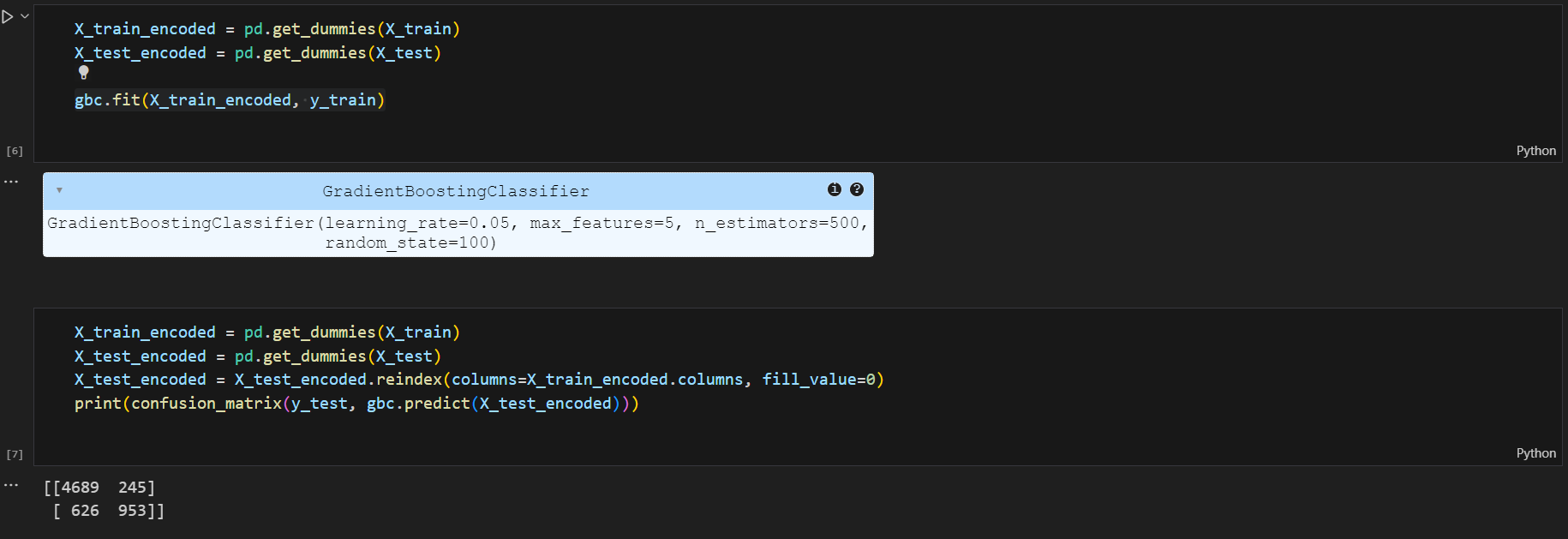
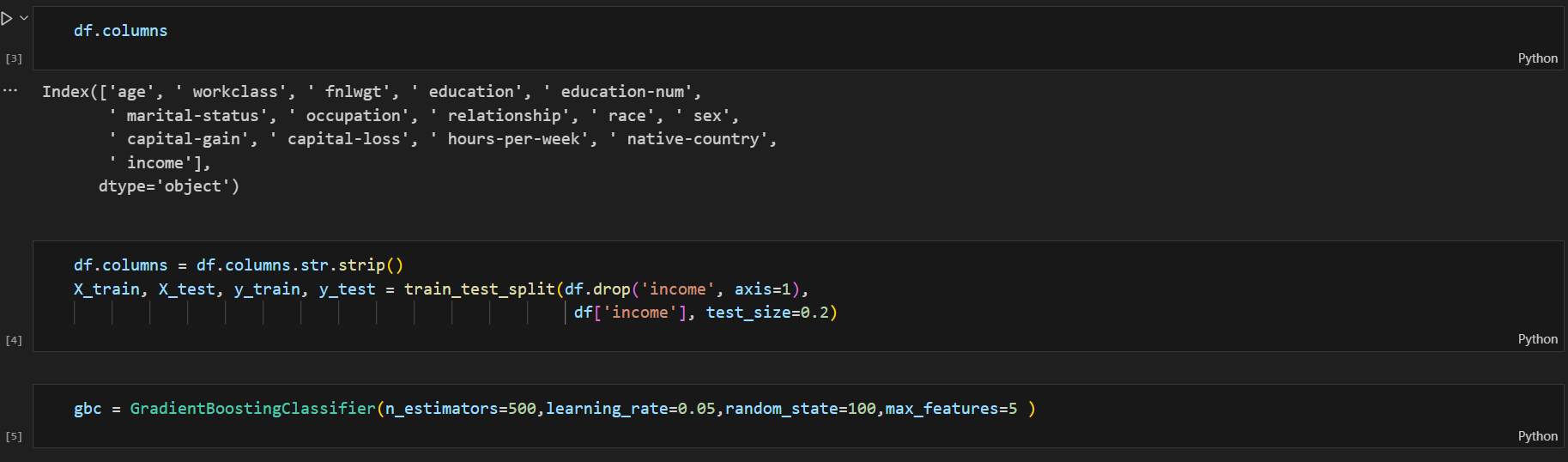
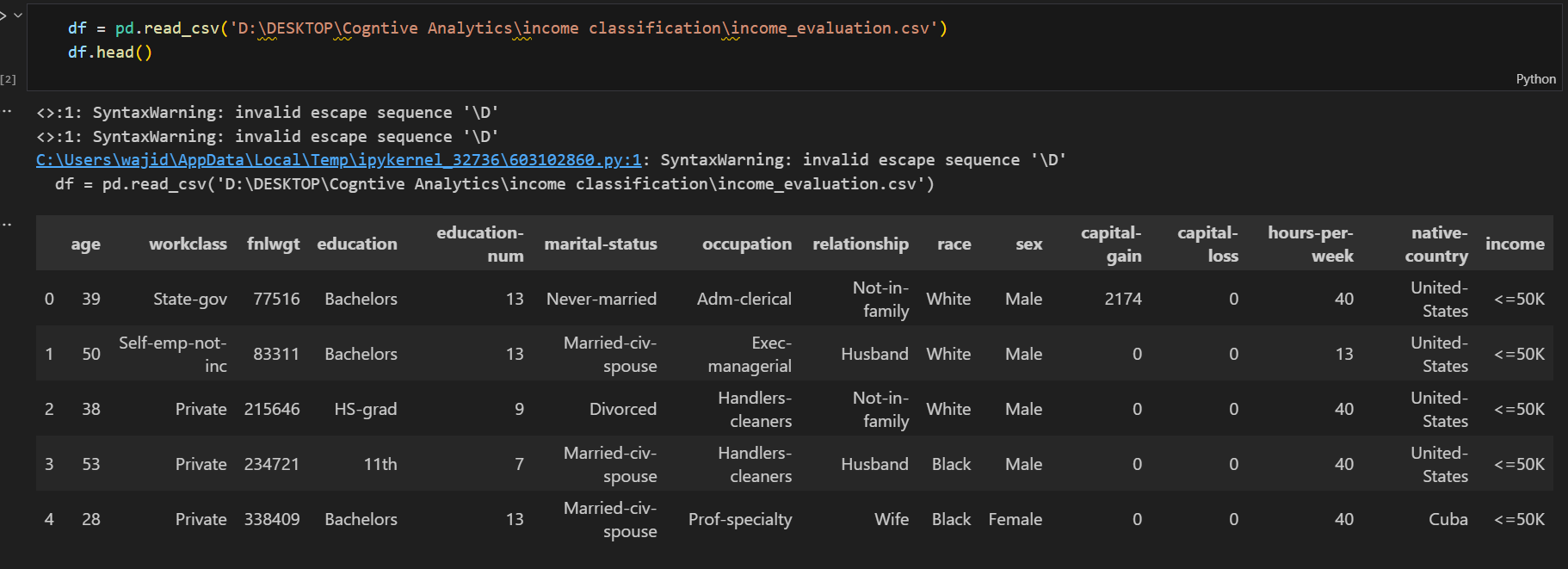
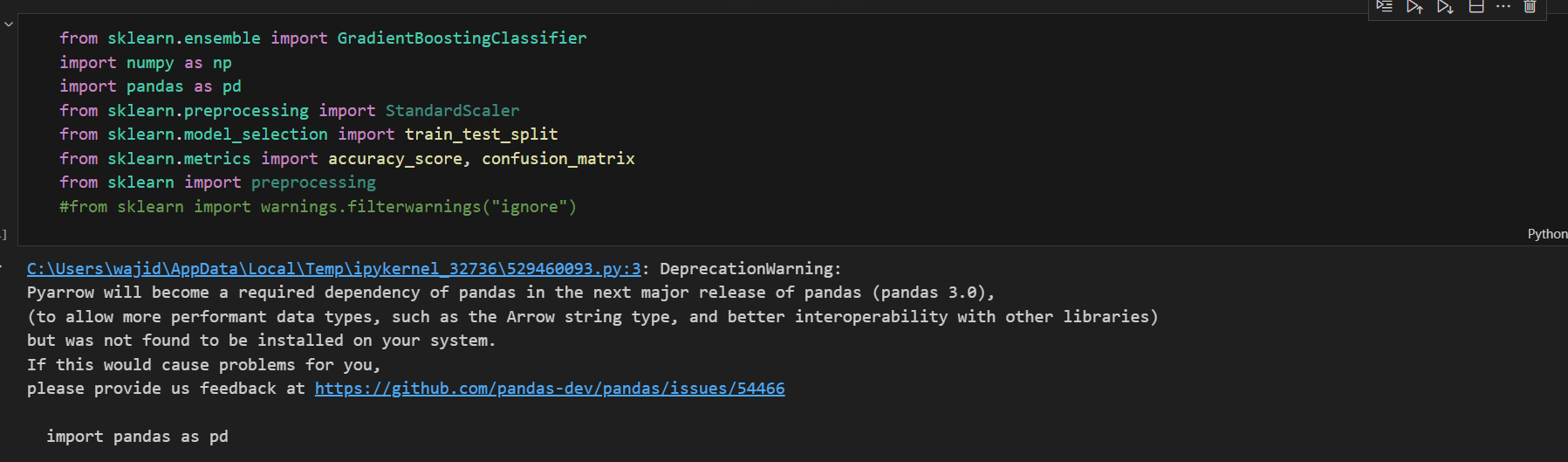
1. Accuracy (0.91): This metric tells us how well the model performed overall. An accuracy of 91% means that the model classified 91% of the instances correctly.
2. Precision (0.70): This metric tells us the proportion of positive predictions that were actually correct. A precision of 70% means that out of all the instances that the model predicted to churn, 70% of them actually churned.
3. Recall (0.48): This metric tells us the proportion of actual positive instances that were identified correctly. A recall of 48% means that out of all the customers who actually churned, the model identified only 48% of them.

In simpler terms, the model is very good at identifying non-churning customers (high accuracy), but it struggles to identify churning customers (low recall). This could be because churning customers are a smaller proportion of the overall dataset, so the model is less likely to learn their characteristics.

Experiment 6

Implementing Gradient Boosting:

Dataset: <https://www.kaggle.com/lodetomasi1995/income-classification>



Interpretation of the Confusion Matrix and Model Performance

Based on the confusion matrix and performance metrics, here's an interpretation of the model's behaviour on the dataset:

Confusion Matrix:

* Correct Classifications:
  + The model correctly classified 4689 data points in the <=50K class and 953 data points in the >50K class.
  + This indicates a good ability to identify data points belonging to each class.
* Incorrect Classifications:
  + The model misclassified 245 data points from <=50K class and 626 data points from the >50K class.
  + There's a higher number of misclassifications for the >50K class, suggesting the model struggles more with this category.

Performance Metrics:

* Overall Accuracy: 87% - This indicates a good overall performance of the model in classifying data points correctly.
* Precision:
  + <=50K: 88% - The model is good at identifying true positives for the <=50K class (out of all predicted positives, 88% are actually positive).
  + 50K: 80% - The model is less precise for the >50K class, meaning there might be more false positives (predicted positive but actual negative).

Recall:

* <=50K: 95% - The model captures most of the actual positives in the <=50K class (out of all actual positives, 95% are predicted positive).
* 50K: 60% - The model misses a significant portion (40%) of the actual positives in the >50K class (false negatives). This is the major contributor to the lower precision for this class.

F1-Score: This metric balances precision and recall. It's higher for the <=50K class (0.92) compared to the >50K class (0.69), again reflecting the model's better performance for the former.

Overall Interpretation:

The model performs well in classifying data points with an overall accuracy of 87%. It excels at identifying <=50K class data points with high precision (88%) and recall (95%). However, the model struggles with the >50K class, particularly with recall (60%), leading to more false negatives (missed positives). This suggests the model might be biased towards the majority class (<=50K) or have difficulty learning the patterns in the >50K class data.