**Preprocessing Strategy & Selection of Models**

We need to process our data before we can feed it into any ML model. We also have to perform whatever preprocessing steps that we take in our training data, into the testing data, as well when we are predicting on our test data.

**Strategies for Preprocessing Our Data:**

Our main goal is "Reusability"; as such we will be creating functions for each preprocessing step that we take. In this way we will only need to call our functions on the predict.ipny notebook to do the same preprocessing steps.

In our preprocessing steps we had successfully created functions that handled missing values, outliers, encoded categorical features. In our dataset we did not have any missing values but we still made a function to take care of the missing values for future data. For missing values, we had imputed missing values with mean for numerical features and mode for our categorical features. To handle the outliers we used Z-Score index to find outliers and handled them using Interquartile Range (IQR). We found 73 outliers and clipped almost all of them.

We encoded our categorical features using one-hot encoding method so that most of our data is retained and we can easily convert the categorical features into a numerical one without much hassle.

We needed to scale the data first so that all of our data are in the same scale and easily comparable. Since our data has not so vastly different scales, a minmax scaler was used.

After that, we needed to handle the class imbalances in our dataset. We used up -sampling technique like SMOTE to handle the imbalances and made sure our ML model can learn the patterns from all the classes and not just the majority class.

We did not need to do principal component analysis to reduce the features as there were not too many features and we had the computing we need to complete the training process

**Strategies for Implementing Machine Learning Models:**

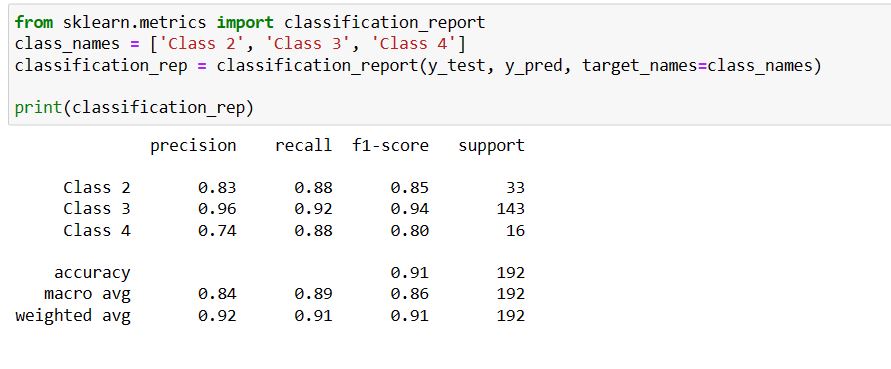
We trained our models; first we trained the data with basic classification models and then moved on to more advanced boosting techniques like XGBoost and CatBoost etc.

We then ran a loop to train all the models and train and tune hyperparameters in this loop and selected the best model.

We saved this tuned model to make predictions in our previously unseen test dataset which we set aside in the very beginning on the project.

**Model Performance Report**

**Model Performance Report on Training Data:**

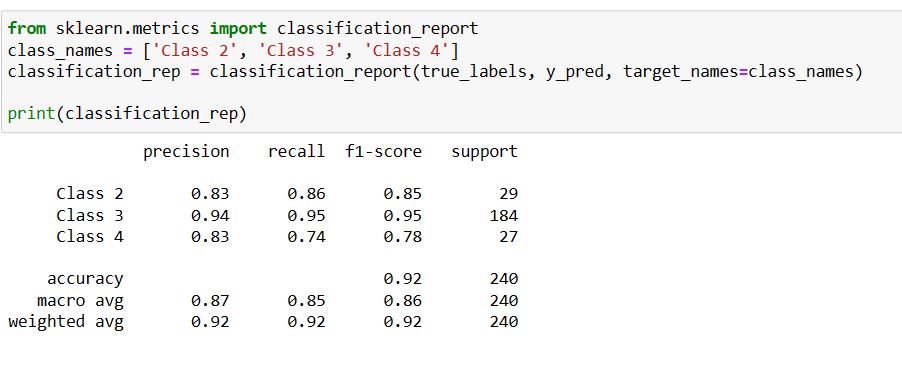


**Class 2 (Employee Performance Rating 2):** This class shows good precision and recall, indicating that the model is effective at identifying employees with a performance rating of 2. However, there is a slight trade-off between precision and recall.

**Class 3 (Employee Performance Rating 3):** The model demonstrates strong precision and recall for employees with a performance rating of 3, suggesting accurate identification of this category.

**Class 4 (Employee Performance Rating 4):** While the recall is relatively high for employees with a performance rating of 4, the precision is lower. This indicates that the model is capturing a significant portion of these instances, but there may be some false positives. Overall, the model appears to perform well for classes 2 and 3, with balanced precision and recall. For class 4, there is a trade-off between precision and recall, suggesting room for further improvement, possibly by adjusting the model's decision threshold or exploring other techniques to address the class imbalance.

**Model Performance Report on Testing Data:**



**Class 2 (Employee Performance Rating 2):**

Precision: This class shows good precision, indicating that when the model predicts a performance rating of 2, it is often correct.

Recall: The recall for this class is also good, indicating that the model effectively identifies employees with a performance rating of 2.

Overall: The model performs well in identifying employees with a performance rating of 2, with a good balance between precision and recall.

**Class 3 (Employee Performance Rating 3):**

Precision: The model demonstrates strong precision for employees with a performance rating of 3, suggesting accurate identification of this category.

Recall: The recall for this class is also high, indicating that the model effectively captures most instances of employees with a performance rating of 3.

Overall: The model performs very well in identifying employees with a performance rating of 3, with both high precision and recall.

**Class 4 (Employee Performance Rating 4):**

Precision: While the recall is relatively high for employees with a performance rating of 4, the precision is lower. This suggests that the model correctly captures a significant portion of these instances but may also produce some false positives.

Recall: The recall for this class is good, indicating that the model effectively identifies employees with a performance rating of 4.

Overall: The model performs reasonably well in identifying employees with a performance rating of 4, with a trade-off between precision and recall. Further improvements could involve adjusting the model's decision threshold or addressing class imbalance.

In summary, the model demonstrates strong performance for classes 2 and 3, with a good balance between precision and recall. For class 4, there is a trade-off between precision and recall, suggesting room for further optimization.

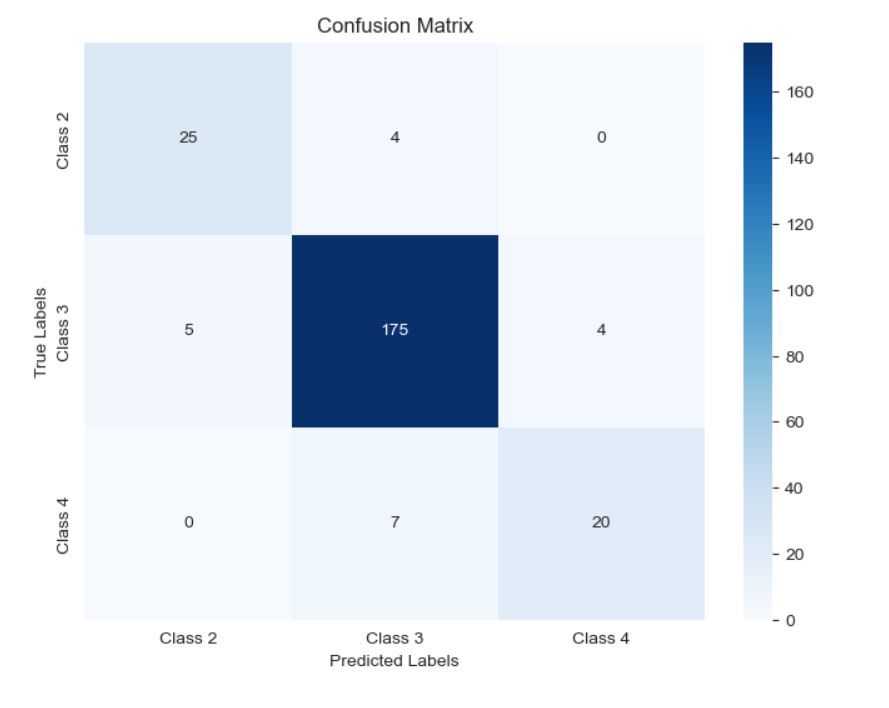
**Strategies for Evaluating Our Machine Learning Models:**

After applying the preprocessing steps using the functions we predicted on our test data, which is previously unseen data. We then evaluated the model predictions.

We evaluated our trained model by classification reports and confusion metrics and found out in which classes the model is performing poorly.

Finally, after we have selected our best model, we will find out the feature importances and find out the top 3 factors that are affecting the employee performances

Below is our confusion matrix for testing data using trained model.



Here we see the misclassification of all three different classes.

**Recommendations To Management on Trained Model**

**Overall Model Performance:** The model shows promising results with an overall accuracy of 92%, indicating that it is capable of correctly predicting the employee ratings in most cases.

**Class-Specific Performance:** Class 3 (Rating 3) has the highest precision, recall, and F1-score among all classes. This suggests that the model is highly accurate in predicting employees with a rating of 3. Class 2 (Rating 2) has decent precision and recall, indicating that the model performs well in identifying employees with a rating of 2. Class 4 (Rating 4) has lower precision and recall compared to the other classes. This could indicate that the model has more difficulty correctly identifying employees with a rating of 4.

**Recommendations:** The model's performance varies among different employee rating classes. It might be beneficial to investigate why Class 4 predictions have lower precision and recall. Are there specific features or patterns that contribute to this discrepancy? Addressing these issues could potentially improve the model's performance for Class 4 predictions.

**Use Case and Business Impact:** Assess the business impact of misclassifications for each rating category. For instance, misclassifying a high-performing employee (Class 4) could have different consequences compared to misclassifying an average-performing employee (Class 3). Consider the consequences of false positives and false negatives in the context of employee performance evaluations.

**Imbalanced Classes:** Class 3 has the largest number of instances, potentially causing an imbalance in the training data. If accurate prediction of other classes is equally important, consider strategies such as oversampling or using different evaluation metrics to address this imbalance.

**Continuous Monitoring:** Regularly monitor the model's performance over time and validate its predictions against new data. Periodic updates and model retraining can help maintain accuracy and relevance.

**Feedback Loop:** Establish a feedback loop with the team that generates the employee ratings. Collect feedback on the model's predictions and refine the model based on the input from domain experts.