Detailed readme on predictive edge using Stacking Ensemble for smarter HR decision making

**README: Predictive Edge Using Stacking Ensemble for Smarter HR Decision-Making**

**Overview:** This README provides a detailed guide on how the Stacking Ensemble technique can offer a predictive edge for smarter HR decision-making. It explains the process, tools, and key benefits of applying stacking ensemble in HR analytics, especially in scenarios such as employee attrition, performance prediction, and workforce optimization.

**1. Introduction to Predictive HR Analytics**

Predictive HR analytics uses statistical models and machine learning to anticipate workforce trends, such as employee turnover, performance levels, and engagement. By using predictive models, HR teams can make more informed decisions, improve employee retention, optimize recruitment, and enhance employee satisfaction.

**Challenges:**

* HR data often exhibits high dimensionality and class **imbalance (e.g., attrition data where only a small percentage leaves).**
* A single algorithm may not provide the best accuracy across all performance metrics (precision, recall, F1-score and auc/roc).

**2. Why Use Stacking Ensemble?**

**Stacking Ensemble** combines the strengths of multiple machine learning models to improve predictive performance. Instead of relying on a single algorithm, stacking uses several models (base learners) and a meta-learner to predict the outcome. This approach can provide higher predictive power and generalization compared to individual models.

**Key Advantages:**

* **Improved Accuracy:** Stacking ensembles tend to outperform single models by leveraging the strengths of multiple algorithms.
* **Better Generalization:** Reduces overfitting by combining models that make different assumptions.
* **Versatile Prediction:** Can be used for various HR applications such as predicting employee attrition, recruitment success, or promotion likelihood.

**3. Stacking Ensemble Architecture**

The architecture consists of two levels:

1. **Base Learners:** These are multiple machine learning models, each trained to predict the target variable (e.g., logistic regression, decision trees, random forests, gradient boosting machines, etc.).
2. **Meta-Learner (Blender):** This model takes the outputs from the base learners as input features and learns to make the final prediction. Commonly, logistic regression or XGBoost model is used as the meta-learner.

**4. Steps to Build a Stacking Ensemble for HR Decision-Making**

**Step 1: Data Collection and Preprocessing**

* Data Source is from 10Alytics
* Collect relevant HR data (e.g., employee demographics, performance ratings, engagement scores, job history).
* Exploratory Data Analysis (EDA): Through exploratory data analysis (EDA) and Feature Engineering, unique insights and patterns specific to the dataset were uncovered, enabling a deeper understanding of the data's underlying structure and relationships. There was no missing data or duplicates. Data Preprocessing involved dropping redundant features, encoding categorical variables, segmenting the dataset into data and target labels, scaling numeric features, and plotting feature importance visualizations to understand features that may be important predictors.

**Step 2: Model Selection, Training and Evaluation of results**

* Splitting data into training and evaluation datasets sets (80% training, 20% test).
* Oversampling because the dataset is imbalance
* Perform cross-validation with StratifiedKFold (This ensures that the class distribution is maintained in each fold, providing a more accurate estimate of performance.
* Implementing 7 Machine Learning algorithms such as Random Forest Classifier, Support Vector Classifier, Logistic Regression Classifier, Extreme Gradient Boosting Classifier, Stochastic Gradient Descent Classifier, Gaussian Naïve Bayes Classifier, Decision Tree Classifier.
* Hyper Parameter Optimization of the 7 algorithms and Best Parameter Combination for each were selected.

Logistic Regression (LR) stands out as the best model for predicting employee attrition. It has the highest recall (0.77) for the **minority class**, which is crucial in imbalanced classification problems. This high recall ensures that most employees likely to leave are correctly identified, which is critical in attrition prediction. Moreover, the AUC-ROC (0.76) indicates that it can effectively distinguish between those who stay and those who leave.

Random Forest Classifier (RFC) and XGBoost (XGBC) are good for the **majority class** but underperform for the minority class. Their precision and recall for class 1 are lower than LR, making them less suitable for detecting employee attrition cases in imbalanced datasets.

**To enhance the rigor of our modeling, we must employ a Stacking Ensemble approach that leverages the synergy of strengths and weaknesses among multiple models, resulting in a robust and reliable prediction. Base Learners**

**Step 3: Model Selection, Training of Base Learners and Stacking Ensemble Implementation**

* Choose Logistic Regression and XGBoost machine learning algorithms to act as base learners.
* Train each base learner on the training dataset using a Fine-tune parameter to achieve optimal performance.
* Creating a base learner predictions file by importing joblib, load the saved model. Make predictions to create meta-features for training the meta-learner.
* Train the meta-learner on the predictions of the base models using Random Forest Classifier and Save the trained meta-learner to a file using joblib.dump

**Step 4: Prediction on unseen Testing data**

* Ensure that `testing data` is preprocessed (dropping off the redundant data, encoding the categorical features to numerical ones using pd.get\_dummiesscaling just like your training dataset.
* Make predictions with the base learners (Logistic Regression and XGBoost)
* Create meta-features for the testing data by combining the predictions from each of the base learners
* Load the meta-learner model
* Make final predictions on the testing data using the meta-learner
* Cross-validation

**Step 5: Evaluate Performance of Stacking Ensemble on unseen Testing data**

**Accuracy: 0.9745**

**Precision (Class 0): 0.9666 Precision (Class 1): 0.9827**

**Recall (Class 0): 0.9830 Recall (Class 1): 0.9660**

**F1 Score (Class 0): 0.9747 F1 Score (Class 1): 0.9743**

**AUC-ROC: 0.9925**

**Confusion Matrix:**

**[[694 12]**

**[ 24 682]]**

The stacking ensemble method has delivered impressive results based on the provided metrics. Here's an analysis of the performance:

Accuracy: 0.9745

The overall accuracy of 97.45% indicates that the model correctly classifies the vast majority of both classes (those who stay and those who leave). However, accuracy alone can be misleading in an imbalanced dataset, so we need to focus more on class-wise precision, recall, and F1 scores.

Precision (Class 0: 0.9666, Class 1: 0.9827)

Precision for Class 0 (0.9666): This means that when the model predicts an employee will stay (class 0), it is correct 96.66% of the time. There are a few positives for employees predicted to stay. Precision for Class 1 (0.9827): When the model predicts an employee will leave (class 1), it is correct 98.27% of the time, meaning very few false positives. This is excellent for attrition prediction, as it minimizes cases where the model incorrectly predicts someone will leave.

Recall (Class 0: 0.9830, Class 1: 0.9660)

Recall for Class 0 (0.9830): The model correctly identifies 98.30% of employees who will stay. This is critical for not misclassifying the majority class, as a high recall ensures that most employees who are not leaving are correctly predicted. Recall for Class 1 (0.9660): The model correctly identifies 96.60% of the employees who will leave. This high recall is crucial because it means the model is effectively catching most of the employees likely to leave, which is important in an imbalanced dataset like employee attrition.

F1 Score (Class 0: 0.9747, Class 1: 0.9743)

The F1 scores for both classes are extremely close (0.9747 for Class 0 and 0.9743 for Class 1). This indicates a near-perfect balance between precision and recall for both classes. A high F1 score means that the model is not sacrificing one for the other, which is ideal for handling class imbalance.

AUC-ROC: 0.9925

The AUC-ROC score of 0.9925 is outstanding. It shows the model’s ability to discriminate between the two classes (employees who stay vs. employees who leave) is almost perfect. An AUC-ROC score close to 1 means the model is excellent at distinguishing between those who will leave and those who will stay.

Confusion Matrix:

True Positives (TP): 682 – Employees who actually left and were correctly predicted to leave. True Negatives (TN): 694 – Employees who stayed and were correctly predicted to stay. False Positives (FP): 12 – Employees who were predicted to leave but actually stayed (false alarm). False Negatives (FN): 24 – Employees who were predicted to stay but actually left (missed predictions). The confusion matrix shows low false positives and low false negatives, meaning the model performs very well on both classes.

Analysis and Conclusion:

Overall Performance: The stacking ensemble has provided an exceptional level of performance, with high precision, recall, and F1 scores for both classes, as well as an almost perfect AUC-ROC score. This means that the model is very reliable for predicting employee attrition, which is challenging in imbalanced datasets.

Class Imbalance Handling: Both class 0 (stay) and class 1 (leave) were handled well. The model has a good balance of predicting those who will leave (class 1) with high recall (96.60%) and precision (98.27%), which is crucial in this context since identifying employees who are likely to leave is the main goal.

Trade-off Between Precision and Recall: The model achieves an excellent balance between precision and recall for both classes, particularly for the minority class (class 1), which is usually harder to predict. This indicates that the ensemble method has successfully learned from the base models and meta-learner to optimize performance across the classes. In summary, the stacking ensemble method is highly effective for this imbalanced dataset and is well-suited for predicting employee attrition. It strikes a good balance between minimizing false positives (predicting someone will leave when they won’t) and false negatives (failing to predict someone will leave), which is critical in employee retention strategies.

**Step 6: Applying the Stacking Ensemble technique to the 7 optimized individual algorithms**

* Performance Evaluation:

The output shows performance results for seven optimized models (Logistic Regression, Random Forest, XGBoost, Support Vector Classifier, Stochastic Gradient Descent Classifier, Gaussian Naive Bayes, and Decision Tree Classifier) used in a stacking ensemble method to predict a classification task. Let's analyze these results.

1. General Performance Overview

Across all models, the performance metrics are consistently strong, with accuracies around 96-97%, and high precision, recall, and F1 scores. This indicates that each model individually performs well, likely contributing positively to the stacking ensemble. Below is an analysis of key metrics for each model.

1. Accuracy

All models have very similar accuracy, ranging between 96% and 97%. This suggests that no model is significantly underperforming. In particular, Logistic Regression, Random Forest, Support Vector Classifier, Gaussian Naive Bayes, and Decision Tree all have top-tier accuracy (97%).

1. Precision

Class 0 Precision: The precision of identifying class 0 (majority class) is consistently high across all models, with values hovering around 0.95-0.97. This shows that false positives (predicting class 0 when it's actually class 1) are minimal. Class 1 Precision: Precision for class 1 (minority class) also performs very well, typically around 0.96-0.98. This is crucial for minimizing false positives (predicting class 1 when it’s actually class 0), especially for models like Random Forest (0.98) and Gaussian Naive Bayes (0.98).

1. Recall

Class 0 Recall: For class 0, the models show excellent recall, typically ranging from 0.96 to 0.99. Logistic Regression and Decision Tree Classifier show the highest recall for class 0 (0.99), meaning they are very good at correctly identifying class 0 instances. Class 1 Recall: For class 1, recall is similarly high across the models, typically around 0.95 to 0.98. Logistic Regression has a standout performance with a recall of 0.96, meaning it correctly identifies almost all class 1 instances.

1. F1 Score

The F1 scores are also closely matched across all models, typically around 0.96-0.97 for both classes. This balance between precision and recall indicates that the models are performing robustly in both predicting true positives and avoiding false positives for both classes.

1. AUC-ROC

Logistic Regression: 0.9738 Random Forest: 0.9745 XGBoost: 0.9603 SVC: 0.9717 SGD Classifier: 0.9631 Gaussian Naive Bayes: 0.9703 Decision Tree Classifier: 0.9681 AUC-ROC values for the models are generally excellent, ranging from 0.96 to 0.97. Random Forest has the highest AUC-ROC score (0.9745), making it slightly better at distinguishing between the classes compared to others, though all models perform well.

1. Confusion Matrices Insights

The confusion matrices reveal that:

Class 0 (Majority Class): All models consistently show low false positive rates (misclassifying class 1 as class 0), with errors generally ranging from 0.7% to 1.9%. Random Forest and Logistic Regression perform exceptionally well in this aspect.

Class 1 (Minority Class): The models show a similar trend, with false negatives (misclassifying class 0 as class 1) ranging from 1.6% to 2.3%. Logistic Regression, Random Forest, and Support Vector Classifier have fewer false negatives than XGBoost and SGD.

Conclusion: Suitability for Stacking Ensemble

Logistic Regression, Random Forest, Support Vector Classifier, and Decision Tree Classifier emerge as the top performers across most metrics, with high AUC-ROC, accuracy, and balanced precision and recall.

XGBoost shows slightly lower performance in terms of precision, recall, and AUC-ROC compared to the other models, but still performs well enough to be a valuable base learner.

SGD Classifier and Gaussian Naive Bayes also perform well but might introduce more variability in predictions, making them useful for increasing diversity in a stacking ensemble.

The stacking ensemble approach has achieved excellent results by combining the strengths of multiple algorithms, including Logistic Regression, XGBoost, and others. The ensemble has demonstrated robust predictive performance, generalizing well on test data. The success of the ensemble can be attributed to the complementary strengths of its base models, with Logistic Regression and XGBoost providing strong generalizations, and other models offering nuanced decision boundaries. With each base model performing well, the ensemble is likely to be robust and outperform individual models. Overall, the stacking ensemble approach has yielded a powerful and generalizable predictive model.

* Use performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curve to evaluate the model’s predictive power.
* Ensure the model performs well on both the training and test datasets to avoid overfitting.

**Step 7a: Comparative Evaluation Performance on stack ensemble technique class 1 vs individual tuned/optimized algorithms class 1 (Attrition)**

This compares the performance of a stacked ensemble technique and individual tuned/optimized algorithms in predicting employee attrition. The ensemble technique outperforms individual tuned algorithms in:

1. Accuracy: Ensemble (96%-97%) vs. Individual algorithms (61%-83%)

2. Precision (Class 1 - Attrition): Ensemble (0.96-0.99) vs. Individual algorithms (0.30-0.42)

3. Recall (Class 1 - Attrition): Ensemble (0.95-0.98) vs. Individual algorithms (0.28-0.77)

4. F1-Score (Class 1 - Attrition): Ensemble (0.96-0.97) vs. Individual algorithms (0.33-0.54)

5. AUC/ROC: Ensemble (0.96-0.97) vs. Individual algorithms (0.59-0.76)

The ensemble technique provides more consistent and accurate predictions, making it a more reliable approach for HR decision-making, especially in predicting employee turnover or identifying high-potential candidates for promotions. Additionally, the ensemble technique handles class imbalance more effectively, which is critical in HR settings where false negatives (missed attrition cases) are costly.

**Step 7b: Comparative Evaluation Performance on stack ensemble technique class 0 vs individual tuned/optimized algorithms class 0 (Non – Attrition)**

This compares the performance of a stacked ensemble technique and optimized algorithms in predicting class 0 (non-attrition). The metrics used are precision, recall, F1-score, and AUC/ROC. The results show that the stacked ensemble consistently outperforms the optimized algorithms in all metrics, demonstrating better accuracy, reliability, and discrimination between non-attrition and attrition cases.

Key findings:

- Precision: Stacked ensemble (0.95-0.97) vs. Optimized algorithms (0.85-0.94)

- Recall: Stacked ensemble (0.97-0.99) vs. Optimized algorithms (0.56-0.95)

- F1-Score: Stacked ensemble (0.96-0.97) vs. Optimized algorithms (0.70-0.90)

- AUC/ROC: Stacked ensemble (0.96-0.97) vs. Optimized algorithms (0.61-0.76)

The stacked ensemble shows superior performance in predicting non-attrition, with higher precision, recall, F1-score, and AUC/ROC values. This indicates that the ensemble is better at identifying true non-attrition cases while minimizing errors, making it a more reliable approach for predicting class 0 (non-attrition).

**Step 7c: Comparative Evaluation of the Confusion Matric Performances on stack ensemble technique vs individual tuned/optimized algorithms**

* **Stacked Ensemble**: Across models, the number of misclassified instances (both false positives and false negatives) is consistently low. For instance, for Logistic Regression in the ensemble, there are approximately **2 false negatives and 1 false positive**, which is highly efficient for HR decision-making.
* **Tuned/Optimized Algorithms**: For individual models, the number of misclassified instances is notably higher. For example, the **Random Forest Classifier** shows **13 false negatives and 7 false positives**, and **Gaussian Naive Bayes** shows an even more imbalanced misclassification, with **35 false positives**.

**Step 8: Model Deployment:**

* Once deployed, the stacking ensemble model can be integrated into HR decision-making systems for real-time predictions and scenario planning.

**Step 9. Conclusion**

By implementing a stacking ensemble approach, HR teams can gain a predictive edge, ensuring more accurate, reliable, and data-driven decisions. This methodology enables smarter resource allocation, personalized employee engagement strategies, and proactive retention efforts, contributing to overall workforce optimization and a healthier organizational culture.

**Step 10: Future Work and Improvements**

* Incorporate deep learning models into the ensemble to capture complex relationships in the data.
* Apply natural language processing (NLP) on unstructured data such as employee feedback or performance reviews.
* Continuously monitor the model's performance, tunning and updating it with new data to maintain accuracy over time.

This README is intended to help HR professionals and data scientists understand the power of stacking ensemble models in HR analytics and how to leverage them for better decision-making.