**Model Optimization and Tuning Phase Template**

|  |  |
| --- | --- |
| Date | 3 May 2024 |
| Team ID | 737820 |
| Project Title | EcoForecast: AI-Powered Prediction of Carbon Monoxide Levels |
| Maximum Marks | 10 Marks |

**Model Optimization and Tuning Phase**

The Model Optimization and Tuning Phase involves refining machine learning models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

### Hyperparameter Tuning Documentation (6 Marks):

|  |  |  |
| --- | --- | --- |
| **Model** | **Tuned Hyperparameters** | **Optimal Values** |
| Decision Tree Regressor | n\_estimators, max\_depth, learning\_rate, random\_state | n\_estimators=100  , max\_depth=3,learning\_rate=0.1, random\_state=0 |
| Random Forest Regressor | n\_estimators, random\_state  . | n\_estimators=20, random\_state = 0 |
| XGBoost | n\_estimators,learning\_rate, max\_depth, random state | n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=0 |
| K-Nearest Neighbour | n\_neighbors | n\_neighbors=15 |
|  |  |  |
|  |  |  |
|  |  |  |

### Performance Metrics Comparison Report (2 Marks):

|  |  |  |
| --- | --- | --- |
| **Model** | **Baseline Metric** | **Optimized Metric** |
| Decision Tree | training Accuracy= 0.4111602033098315 | Test Accuracy= 0.368581227731857 |
| Linear Regression | Training Accuracy= 0.06503968505710755 | Test Accuracy= 0.061523854715849424 |
| Random Forest | Training Accuracy 0.4102458601234813 | Test Accuracy= 0.36726404580326355 |
| K-Nearest Neighbour | Training Accuracy= 0.3583124866059221 | Test Accuracy= 0.32659350142943666 |
| XGBoost | Training Accuracy 0.29635162610429255 | Test Accuracy 0.27036413286966887 |

### Final Model Selection Justification (2 Marks):

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |
| Linear Regression | Linear regression was considered due to its simplicity and interpretability. It assumes a linear relationship between the input features and the target variable, which can be advantageous if the relationship is indeed linear or can be reasonably approximated as such. However, it may not perform well if the relationship is highly non-linear. |
| Decision Tree Regressor | The Decision Tree Regressor was explored because it can capture non-linear relationships between features and the target variable. Decision trees are interpretable and can handle both numerical and categorical data. However, they are prone to overfitting, especially if the tree is allowed to grow too deep. |
| Random Forest Regressor | Random Forest Regressor emerged as a contender due to its ability to mitigate overfitting and handle noisy data. It is an ensemble method that combines multiple decision trees, providing robust predictive accuracy. Random forests are less sensitive to outliers and noise compared to individual decision trees. |
| XGBoost | XGBoost was considered for its advanced implementation of gradient boosting, which often outperforms other algorithms in terms of predictive accuracy. It is highly customizable and can handle missing values. XGBoost is particularly suitable for modeling complex relationships in the data. |
| KNN(K-Nearest Neighbors) | KNN was assessed for its simplicity and intuitive approach. It makes predictions based on the average of the k-nearest neighbors in the feature space, without making strong assumptions about the underlying data distribution. KNN is suitable for capturing complex, non-linear relationships, especially in smaller datasets. |
|  |  |
|  |  |