## Customer Churn Prediction

**Phase 2**

**Problem Definition:**

The objective of this problem statement is to develop a predictive model that can accurately identify customers who are at risk of churning. By proactively identifying potential churners, businesses can take targeted actions to retain these customers, thus reducing the overall churn rate. The prediction model should leverage historical customer data and relevant features to make accurate churn predictions

Improving the prediction accuracy of customer churn using advanced machine learning techniques like ensemble models and feature engineering is an excellent approach. Customer churn prediction is a crucial task for businesses, as it helps in retaining valuable customers. Here's a step-by-step guide on how to incorporate these techniques:

1. Data Pre-processing:

* Data Cleaning: Start by cleaning your dataset. Remove or impute missing values and handle outliers if necessary.
* Feature Scaling: Standardize or normalize numerical features to ensure they have the same scale.
* Feature Encoding: Encode categorical variables using techniques like one-hot encoding or label encoding.

2. Feature Engineering:

* Domain Knowledge: Understand your industry and business to create relevant features. For example, you might create features like customer tenure, average transaction value, or customer lifetime value.
* Feature Selection: Use techniques like correlation analysis, feature importance from ensemble models, or recursive feature elimination to select the most important features.

3. Ensemble Models:

* Random Forest:

Random Forest is an ensemble of decision trees that can handle both classification and regression tasks.

It's robust to overfitting and can provide feature importances.

Tune hyperparameters like the number of trees and tree depth.

Gradient Boosting:

Algorithms like XGBoost, LightGBM, and CatBoost often perform well in churn prediction tasks.

These models are strong learners and can capture complex relationships in the data.

Hyperparameter tuning is essential for optimal performance.

* Stacking:

Combine multiple models (e.g., logistic regression, random forest, gradient boosting) and use another model to predict based on their outputs.

Stacking can often yield better results by leveraging the strengths of different models.

4. Cross-Validation:

* Use techniques like k-fold cross-validation to assess the model's performance and avoid overfitting.

5. Evaluation Metrics:

* Choose appropriate evaluation metrics such as accuracy, precision, recall, F1-score, ROC AUC, or customer lifetime value (CLV) if it aligns with business goals.

6. Hyperparameter Tuning:

* Employ techniques like grid search or random search to find the best hyperparameters for your models.

7. Handling Class Imbalance:

* Churn datasets are often imbalanced, where the number of churned customers is much smaller than loyal customers. Use techniques like oversampling, undersampling, or SMOTE (Synthetic Minority Over-sampling Technique) to address this issue.

8. Regularization:

* Apply regularization techniques like L1 or L2 regularization to prevent overfitting, especially in logistic regression or linear models.

9. Monitoring and Maintenance:

* Continuously monitor the model's performance in a production environment and retrain it periodically with new data.

10. Interpretability:

* Consider using model interpretability techniques like SHAP values or LIME to understand why the model makes specific predictions. This is crucial for business stakeholders.

11. Deployment:

* Deploy the final model in a production environment, ensuring it integrates seamlessly with your business processes.

12. Feedback Loop:

* Establish a feedback loop to incorporate user feedback and continuously improve the model.