Machine Learning

Shahid Beheshti University

Spring 2025

Employee Retention Classification

1 Theoretical

1. a) What happens if you set the regularization parameter C=0 in soft-margin SVM? How does the model behave in that case?

   b) Can the SVM margin ever go to infinity? Under what conditions might this happen, and what would it imply?

2. a) Explain the role of the constant c in SVM with polynomial kernel. What changes if c=0 versus c=1? Provide a brief example or geometric interpretation illustrating the impact of including or excluding the constant term.

   b)What actions should you take if you have trained an SVM classifier using an RBF kernel but notice that it underfits the training set? Would it be appropriate to increase or decrease the value of (gamma) or C, or both?

   c)In the figure below , identify what the solid and the dashed lines represent. Using the RBF kernel equation, derive the mathematical expressions that explain their behavior in the feature space by an example and a visual figure.

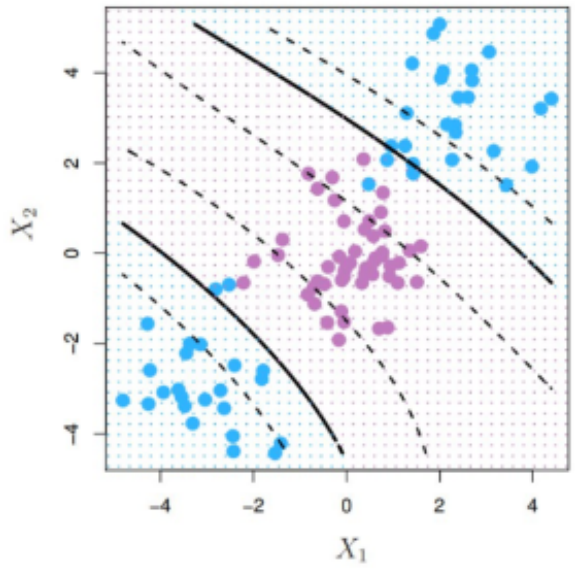

Figure 1: Question 2, c

3. Explain the difference between pre-pruning and post-pruning in decision tree learning. In what situations would each approach be preferred, and what are the trade-offs in terms of bias, variance, and computational cost?

4. Is a node's Gini impurity generally lower or greater than its parent's? Is it generally lower/greater, or always lower/greater?

5. Prove that in the Bagging method only about 63% of the total original examples (total training set) appear in any of sampled bootstrap datasets. Provide proper justification.

6. a) Explain Adaboost and Gradient boosting, what advantages do they have over traditional tree-based algorithms such as decision tree and random forest? and when do we use each of them?

   b)Specify how do you tweak the hyperparameters of the following model in mentioned circumstances:

   • AdaBoost - Underfitting

   •Gradient Boosting - Overfitting

   c) How do categorical features get handled in boosting frameworks?

2 Practical

In this assignment, you will build and compare a variety of classification models to predict whether an employee will leave the company (‘left_company‘), using the provided HR dataset.

Dataset Overview

The data files are as follows:

- `train.csv` (1341 rows, 35 columns): 33 feature columns + `ID` + target `left_company`
- `test.csv` (336 rows, 34 columns): same features + `ID`, no target
- `sample_submission.csv` (336 rows, 2 columns): template for submission (`ID`, `left_company`)

Features include numerical, ordinal, and categorical variables such as age, salary, travel frequency, job satisfaction ratings, and tenure measures.

Assignment Tasks

1. Data Preprocessing and EDA

- Clean the data: handle any anomalies, convert categorical variables (e.g., one-hot or mean encode).
- Perform exploratory data analysis to understand feature distributions and correlations.
- Visualize class imbalance and study relationships between features and the target.

2. Baseline Model

- Split the training set into train/validation (e.g., 80/20).
- Train a **Logistic Regression** (with and without class weights).
- Evaluate using:

$$\text{Precision}, \quad \text{Recall}, \quad F_1\text{-score}, \quad \text{ROC-AUC}$$

- Plot the ROC curve and confusion matrix.

3. Advanced Models

Train and compare at least two of the following:

- **Support Vector Machine (SVM)** with different kernels (linear, RBF).

- **Random Forest** or **Gradient Boosting** (XGBoost, LightGBM).

  bonus Implement one of the models above(**Random Forest** or **Gradient Boosting** (XGBoost, LightGBM)) from scratch and compare with the result given by libraries implementation.

- **k-Nearest Neighbors** or **Naïve Bayes** (for contrast).

For each model:

- Apply appropriate hyperparameter tuning (grid search or randomized).
- Use class weight adjustments or sampling techniques (SMOTE, oversampling, undersampling) to address imbalance.
- Report validation performance in terms of F_1-score and ROC-AUC.

4. Handling Imbalanced Data

Machine learning models often struggle when one class is much rarer than the other. In this section you should:

- **Implement** at least two different imbalance-handling techniques, such as:
  - Random undersampling of the majority class
  - Random oversampling or SMOTE for the minority class
  - Class weighting in your estimator's objective function
  - Cost-sensitive learning or ensemble approaches like EasyEnsemble
- **Compare** their effect on performance metrics (precision, recall, F_1, ROC-AUC) and discuss trade-offs.

5. Model Stacking (Optional)

- Create a simple stacking ensemble combining your best two base models.
- Use cross-validated predictions on the training set to train a meta-learner.
- Compare ensemble performance to individual models.

Leaderboard & Bonus

After you generate your final predictions on `test.csv` and submit to the course leaderboard:

- **Leaderboard**: Ranks will be updated based on F_1-score on hidden test data.
- **Bonus Points**: Top ranked scores will get a considerable bonus score in their grade.

Final Submission

- Retrain your best-performing model(s) on the full training set.
- Generate predictions on `test.csv`.
- Create `submission.csv` with columns `ID,left_company`.

Deliverables

- A PDF report including:
  - EDA findings and preprocessing steps.
  - Model descriptions and hyperparameter settings.
  - Performance comparison (tables and plots).
  - Discussion of class imbalance handling and its impact.
  - Final leaderboard rank and reflection.
- A Jupyter notebook (or script) with all code and outputs.
- `submission.csv` file for final predictions.