**Homework 1: Credit Approval Classification with SVM & kNN**

Georgia Institute of Technology, Business Analytics

Introduction to Analytics Modeling

Professor Joel Sokol

August 28, 2025

Files submitted: homework1\_answers.pdf (this doc), homework1.Rmd

**Question 2.1 - Classification Model Scenario**

**Situation Overview**

Predicting whether a patient will show up (1) or miss (0) a scheduled medical appointment. The task is framed as a binary classification problem, where the model uses relevant predictors to classify the outcome.

**Selected Predictors**

To build a predictive model, the following five predictors are considered:

1. **Days of Delay Between Booking and Appointment Date:** Longer waiting periods between scheduling and the appointment may increase the chance of no-shows due to forgetfulness or changing priorities.
2. **Reminder Received** *(Binary: Yes/No):* Patients who receive reminders by text or email are more likely to attend.
3. **Past Attendance History:** The number of previously missed appointments is often a strong predictor of future attendance.
4. **Age of Patient**: Different age groups may exhibit different attendance patterns (e.g., younger patients with unpredictable schedules vs. older patients prioritizing healthcare visits).
5. **Distance from Clinic** *(in kilometers or miles):* Patients living farther away may be more likely to cancel or miss visits, especially in poor weather or without transportation.

**Discussion of Results**

This scenario demonstrates how everyday healthcare operations can benefit from machine learning. By identifying factors linked to no-shows, clinics could reduce missed appointments through targeted interventions, such as sending reminders, offering telehealth options, or prioritizing same-day scheduling. The predictors chosen are measurable, practical, and directly connected to patient behavior, making this an appropriate use case for classification modeling.

**Question 2.2 - Classification on Credit Approval Data**

Data. 654 observations; 10 predictors (A1, A2, A3, A8, A9, A10, A11, A12, A14, A15) and binary response R1 (0/1). No missing values. Features were standardized where noted.

**A) Support Vector Machine (SVM)**

**Methodology**

* Algorithm: kernlab::ksvm
* Settings: type = "C-svc", linear kernel (vanilladot), scaled = TRUE
* Hyperparameter search: 𝐶 ∈ { 10 − 3 , 10 − 2 , 10 − 1 , 1 , 10 , 100 , 1000 }
* Evaluation: training accuracy on full dataset (per instructions)
* Coefficient extraction:  
  a=∑i αiyixi via *colSums(model@xmatrix[[1]] \* model@coef[[1]]);* intercept 𝑎0=−𝑚𝑜𝑑𝑒𝑙@𝑏  
  (Coefficients correspond to the **scaled features** because scaled=TRUE.)

**Results**

* Best value: **C = 1** (several C’s tied at the top; 1 chosen as mid-range)
* **Training accuracy:** **0.8639**
* **Confusion matrix (train):**
  + Actual 0 → Pred 0: **286**, Pred 1: **72**
  + Actual 1 → Pred 0: **17**, Pred 1: **279**
* **Class-1 metrics:** Precision **0.795**, Recall **0.943**, F1 **0.862**
* **Classifier (on scaled features):**  
  f(z)=a0+∑j ajzj with 𝑎0 ≈ 0.081484 and weights (order A1,A2,A3,A8,A9,A10,A11,A12,A14,A15)

a≈[−0.001103,−0.000898,−0.001607,0.002904,1.004736,−0.002985,−0.000204,−0.000555,−0.001252,0.106444]

Most influential (by |weight|): **A9** (positive, strongest) and **A15** (positive).

**Discussion**

* Linear SVM performs strongly in-sample and emphasizes **recall** for class 1 (very few false negatives: 17) at the cost of more false positives (72).
* Because features are scaled, coefficients reflect each variable’s contribution in standardized units.
* Beyond requirements (optional work I performed): I briefly tested nonlinear kernels (RBF and polynomial). These increased training accuracy (e.g., polynomial degree 3, C=10 ≈ 0.97–0.98), suggesting a tighter fit but a risk of overfitting without cross-validation.

**B) k-Nearest Neighbors (kNN)**

**Methodology**

* Algorithm: kknn::kknn, with scale = TRUE
* Evaluation: **leave-one-out** with self-exclusion (for each row i, trained on the remaining n−1 rows and predicted i)
* Hyperparameter search: odd k∈{1,3,5,7,9,11,15}

**Results**

* Accuracies by k:  
  k={1,3,5,7,9,11,15}⇒{0.8150,0.8150,0.8517,0.8471,0.8471,0.8517,0.8532}
* **Chosen k=15** (highest LOOCV accuracy)
* **Confusion matrix (LOOCV, k=15)**:
  + Actual 0 → Pred 0: **308**, Pred 1: **50**
  + Actual 1 → Pred 0: **46**, Pred 1: **250**
* **Class-1 metrics:** Precision **0.833**, Recall **0.845**, F1 **0.839**

**Discussion**

* Accuracy improves as k increases (variance reduction), peaking at k=15.
* kNN yields a **more balanced** precision/recall trade-off than the linear SVM: fewer false positives but more false negatives.
* **Evaluation note**. My SVM accuracy is training (in-sample) per the prompt; my kNN accuracy is LOOCV. These aren’t directly comparable—SVM would likely score lower under the same validation scheme.
* **Fairness & data quality.** Real credit datasets can contain historical or representation bias. If certain groups are under-represented, a model may generalize poorly for them, leading to unfair outcomes. Before deployment, I would (i) audit performance across demographic slices, (ii) consider re-balancing or re-weighting, and (iii) document system behavior and limitations.

**Brief Comparison & Notes**

* **Validation schemes differ:** SVM was reported with **training accuracy**, while kNN used **LOOCV**. They’re not directly comparable; SVM accuracy would typically drop under cross-validation.
* **Optional exploration:** Nonlinear SVMs (RBF/polynomial) produced higher training accuracy in a quick check (e.g., poly d=3,C=10≈0.979), which may reflect overfitting without cross-validation.

**REFERENCES**

https://web.archive.org/web/20200121091131/http://www.statsoft.com/Textbook/k-Nearest-Neighbors.