Instructions: Please read the following instructions thoroughly

- For the entire assignment, use `Python` for your analysis. Write your code in a Jupyter Notebook named as `[your-student-ID]_hw1.ipynb` (e.g., `2025-20000_hw1.ipynb`). The use of `R` is not allowed. You are allowed to use any libraries in `Python`.
- Type up your report and save as PDF named as `[your-student-ID]_hw1.pdf`. We do not allow the submission of a photo or a scanned copy of hand-written reports.
- Please upload the two files on eTL **without zipping**. Submissions via email are not allowed. The violation of the filename or submission instruction will result in the penalty of 5 points.
- You can discuss the assignment with your classmates but each student must write up his or her own solution and write their own code. Explicitly mention your classmate(s) you discussed with or reference you used (e.g., website, Github repo) if there is any. If we detect a copied code without reference, it will be treated as a serious violation of the student code of conduct.
- We will apply a grace period of late submissions with a delay of each hour increment being discounted by 5% after the deadline (i.e., 1-minute to 1-hour delay: 95% of the graded score, 1 to 2-hour delay: 90% of the graded score, 2 to 3-hour delay: 85%, so on). Hence, if you submit after 20 hours post-deadline, you will receive 0 points. No excuses for this policy, so please make sure to submit in time.

1. [**50 pts**] In this problem, you will use the `Carseats` data set attached in the assignment (`Carseats.csv`) for linear regression.

   (a) [10 pts] Using `scikit-learn` (`sklearn`)'s `LinearRegression`, fit a multiple linear regression model to predict `Sales` using `Price`, `Urban`, and `US`.
   - Clearly state how you encoded the qualitative variables (`Urban`, `US`) and whether an intercept was included.
   - Report the fitted coefficients (including the intercept) and the training R^2 of the model.

   (b) [7 pts] Write the fitted model in equation form, being careful to handle the qualitative variables properly. (E.g., use indicator variables $\mathbf{1}\{\texttt{Urban} = \text{Yes}\}$ and $\mathbf{1}\{\texttt{US} = \text{Yes}\}$ with a clearly specified baseline.) Provide an interpretation of each coefficient in the model.

   (c) [13 pts] **Closed-form OLS and comparison.** Construct the design matrix $\mathbf{X}$ that includes an intercept column and uses *one* dummy/indicator for each qualitative predictor (to avoid the dummy-variable trap), matching your coding choices in Part (a). Implement the ordinary least squares estimator

   $$\hat{\boldsymbol{\beta}}_{\text{LS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

   Report $\hat{\boldsymbol{\beta}}_{\text{LS}}$ and compare it to the `scikit-learn` coefficients from Part (a) (provide a small table with both and their absolute differences).

(d) [10 pts] For each predictor variable $j \in \{\texttt{Price}, \texttt{Urban}, \texttt{US}\}$, test $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$ at level $\alpha = 0.05$. Indicate for which variables you can reject H_0 (i.e., for which there is evidence of association with `Sales`). *Note:* Since `scikit-learn` does not provide p-values, you may (i) compute t-statistics using the usual LS formulas, or (ii) refit the same model with `statsmodels` for inference.

(e) [10 pts] Perform a validation set approach. That is, create a random 70%/30% train/validation split (set and report a random seed), fit the linear model on the training set, and report the estimated *validation-set* R^2 and MSE. Then, perform 5-fold cross-validation and report the cross-validated R^2 and MSE (averaged across folds). Briefly discuss any differences between these two results.

2. [**50 pts**] In class, we used logistic regression to predict the probability of `default` using `income` and `balance` on the `Default` data set attached in the assignment (`Default.csv`). In this problem, you will (i) fit the model with `scikit-learn`, (ii) implement and optimize the (negative) log-likelihood yourself, and (iii) evaluate generalization via a validation split and cross-validation. Set and report a random seed before any resampling/splitting.

(a) [10 pts] Using `scikit-learn` (`sklearn`)'s `LogisticRegression`, fit a model to predict `default` from `income` and `balance`.

- Clearly state how you encoded `default` (e.g., `Yes`=1, `No`=0) and whether you standardized `income` and `balance`.
- To match the MLE (no regularization), set `penalty='none'` with a suitable solver (e.g., `lbfgs`) *or* use a very large `C` (e.g., `C=1e12`) if `penalty` must be specified. Increase `max_iter` if needed.
- Report the fitted intercept and coefficients and the *training* log-likelihood of the model.

(b) [7 pts] Write the model in equation form and interpret coefficients. Let $\eta_i = \beta_0 + \beta_1 \texttt{income}_i + \beta_2 \texttt{balance}_i$ and $p_i = \Pr(\texttt{default}_i = 1 \mid \texttt{income}_i, \texttt{balance}_i) = \frac{1}{1+e^{-\eta_i}}$. Provide an interpretation of each coefficient on the *log-odds* scale (sign and relative magnitude), and translate at least one interpretation to the *odds ratio* scale via e^{β_j}.

(c) [15 pts] **Code the likelihood and optimize (no closed form).** Construct the design matrix with an intercept column and columns for `income` and `balance`. Implement MLE using the *log-likelihood* $\log \mathcal{L}(\boldsymbol{\beta})$ for parameter $\boldsymbol{\beta}$ with $y_i \in \{0, 1\}$.

- State the *log-likelihood* $\mathcal{L}(\boldsymbol{\beta})$ for parameter $\boldsymbol{\beta}$ with $y_i \in \{0, 1\}$. What is the gradient $\nabla \log \mathcal{L}(\boldsymbol{\beta})$?
- With $\mathcal{L}(\boldsymbol{\beta})$ and $\nabla \log \mathcal{L}(\boldsymbol{\beta})$, use an off-the-shelf optimizer to compute the MLE (e.g., use `scipy.optimize.minimize` with `BFGS` or `Newton-CG`)
- Report the optimized coefficients, the maximized log-likelihood, and compare *numerically* to the `scikit-learn` estimates from Part (a) (coefficients and log-likelihood). Explain any differences (e.g., convergence tolerance, scaling).

(d) [10 pts] Create a random 70%/30% train/validation split (report the random seed). Fit the model from Part (a) on the training set (using the same preprocessing choices) and evaluate on the validation set. Report the validation *misclassification rate* (use threshold 0.5). Then, perform 5-fold cross-validation and report the cross-validated misclassification rate (averaged across folds). Briefly discuss any differences between these two results.

(e) [8 pts] In addition to variables `income` and `balance`, include a dummy variable for `student` (e.g., $\mathbf{1}\{\texttt{student} = \text{Yes}\}$; specify your baseline). Using 5-fold cross-validation, estimate and report the *average* misclassification rate. Briefly discuss whether including `student` reduces the estimated error.