Applied Data Analysis — Assignment 2

Instructors: Dr.Salavati TA: Peyman Naseri

In the first assignment, you practiced **EDA**, **Preprocessing**, and **Feature Engineering**.

Now, in this homework, we will focus on **Regression** and **Classification Modeling** using the same dataset (or a new one if preferred).

Please make sure your evaluation is meaningful and aligned with the characteristics of your dataset. Different problems require different metrics, and blindly reporting standard metrics (such as MSE or accuracy) is not enough. Think carefully about what performance measure best reflects the real objective of your task. If needed, you may look at similar Kaggle projects, benchmarks, or even use LLM tools to get suggestions about the most appropriate metrics for your specific dataset. However, you must justify your final choice of evaluation metrics in your notebook.

Similarly, if you believe that a different classical model (beyond the ones explicitly listed in this assignment) is more appropriate for your problem, you are encouraged to implement and evaluate it. In that case, clearly explain why this model is a good fit for your dataset and compare its performance with the baseline models.

Dataset Selection

- You may continue working on the same dataset used in Homework 1 or choose a new one.
- The dataset should have **enough numerical and categorical features** to apply regression and classification techniques.
- You can find great options on Kaggle or use a dataset from local or industrial domains.

Submission Guidelines

Your results must be presented in a **well-documented Jupyter Notebook (.ipynb)**.
Follow best coding practices and keep your notebook clean, modular, and reproducible.

How to Submit

1. Use the **same GitHub repository** that you created for Homework 1.
2. You can also share your notebook via **Google Colab** (make sure link access is open).
3. In your final submission, include:
    - GitHub link (to your updated repository)
    - Colab link (if applicable)
    - The actual `.ipynb` file
4. You may structure your code **modularly** (e.g., split parts into `.py` files and import them) instead of putting everything inside one notebook.

After submission, a **short in-person session** will be scheduled for you to **explain and review your assignment**.

Late Submission Policy

- A **10% penalty** will be applied for each late day.

Collaboration Policy

All homeworks must be done **individually**.

Evaluation Criteria

You will be graded **qualitatively** based on the following:

- The analysis solves or meaningfully addresses the problem
- The notebook is clear, readable, and well-commented
- Explanations are concise, insightful, and easy to follow

Bonus point

If you go beyond the basic requirements and add something valuable or creative, you can earn **extra credit**. The goal of bonus points is to reward genuine insight, effort, and clarity — not just extra code.

For example:

- Having a clear and informative **README.md** file in your GitHub repository that briefly explains your models, results, and structure.
- Creating **interactive model comparison visualizations** (e.g., ROC and Precision-Recall curves, confusion matrices, feature importance plots).
- Implementing a **summary table** or **dashboard** comparing regression and classification models based on different metrics.
- Conducting **error analysis** identifies where your models perform poorly and explains why.
- Adding a brief **interpretation section** that explains what each model is learning (e.g., coefficient interpretation for linear models, decision paths for trees).
- Using **cross-validation** to show the stability of your results.
- Comparing models not only by accuracy but also by **training time, complexity, or interpretability**.
- Visualizing **decision boundaries** for 2D toy datasets to show conceptual understanding.

However, unnecessary complexity, heavy libraries, or overly fancy visuals will not earn extra points. Overengineering does not earn extra credit.
Focus on producing a notebook that is **executable, insightful, and educational** rather than overloaded or messy.

Generative AI Policy

The use of tools such as **ChatGPT, Claude, Gemini**, or other similar AI assistants is **allowed and encouraged** but **use them wisely**.

- Try to solve each problem yourself first.

- Then, you may use AI tools to check, improve, or compare your results.
- Remember: what matters most is **understanding** the code you submit, not who wrote it.
- Be cautious, large models often “hallucinate” or produce inaccurate results.

Important: It is recommended that you use the course’s Ai teaching assistant before the deadline and upload your answers, approximate scores, and suggestions for improving your implementations.

Ai ADS Assistant(چاکر شما)

The goal is to help you become confident in solving real data science problems independently.

Homework Components

1. Regression Methods

On the dataset of your choice, practice the following regression algorithms:

- Linear Regression
- Kernel Regression
- Ridge Regression
- LASSO Regression
- (Optional) Any other model you see fit. For example:
    - Polynomial Regression
    - Bayesian Ridge Regression
    - Elastic Net
    - Locally Weighted Regression
    - Decision Tree Regression
    - SVR
    - ARIMA Regression (for time series forecasting)
    - SARIMA Regression (for time series forecasting)

Evaluate model performance on your test set using:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- Mean Absolute Percentage Error (MAPE)

- **R^2 Score**
- (Optional) Any other evaluation metric you see fit. For example:
    - Huber Loss
    - Mean Squared Logarithmic Error (MSLE)
    - Median Absolute Error
    - Root Mean Squared Logarithmic Error (RMSLE)

Discussion Question:

1. Choose the best regression metric for your dataset and justify.
2. Explain when each regression model is preferable.
3. Briefly explain **the kernel trick** in a few sentences and how it can help achieve better regression results.

2. Binary Classification Methods

On the dataset of your choice, practice the following **binary classification** algorithms:

- Logistic Regression for classification
- SVM
- Kernel SVM
- K-Nearest Neighbors (KNN)
    - Tune to find the best number of neighbors (K)
- Decision Trees
    - Tune for the best maximum depth to avoid overfitting
- **Random Forests**
- (Optional) Any other model you see fit. For example:
    - Linear Discriminant Analysis (LDA)
    - Naive Bayes
    - Gaussian Mixture Model Classifier (GMM)

Evaluate model performance on your test set using the following binary metrics:

- accuracy
- Precision
- Recall
- F1-Score
- **Confusion Matrix**
- ROC_Curve
- AUC
- (Optional) Any other evaluation metric you see fit. For example:

    - Sensitivity
    - Specificity
    - PR AUC (Precision-Recall AUC)
    - Cross-Entropy
    - Mutual Information
    - KL Divergence
    - Jensen–Shannon Divergence

Discussion Question:

1. Choose the best classification metric for your dataset and justify
2. Explain **3 techniques** to regularize the training process for decision trees
3. Compare Linear SVM vs Kernel SVM

3. Multiclass Classification Methods

Perform multiclass classification (with at least 4 classes) on your dataset. Implement the following methods:

- Multiclass SVM
- Multiclass Logistic Regression
    - Using One-vs-Rest (OVR)
    - Using the multinomial approach
    - Compute **log loss** for the output
- Multiclass KNN
    - Tune for best number of neighbors (K)
- Multiclass Decision Tree
- **Boosting** Techniques:
    - **XGBoost**
    - LightGBM
    - AdaBoost
    - CatBoost
- (Optional) Any other model you see fit. For example:
    - Multiclass Random Forest
    - Kernel SVM with One-vs-One (OVO)
    - Parzen Window Classifier
    - Radius Neighbors Classifier
    - Voting Classifier (Hard/Soft voting)
    - Stacking Classifier
    - HMM (for time series forecasting)

Evaluate model performance on your test set using the following multi-class metrics:

- accuracy
- Precision for each class

- Recall for each class
- F1-Score (Macro, Weighted, and Micro-averaged)
- (Optional) Any other evaluation metric you see fit. For example:
    - Kappa
    - Segmentation metrics: IoU, Dice
    - Ranking Metrics: Top-K Accuracy, **Recall@K,** NDCG
    - Multi-Label Classification Metrics: Hamming, Jaccard Index, Coverage Error
    - G-Mean

Discussion Question:

1. Choose the best multiclass-classification metric for your dataset and justify.
2. Explain how **KNN** and **Decision Trees** can be extended to **multi-label classification** problems.
3. Suppose we have a multi-label classification problem in football, where each player can belong to some of these 4 classes:
    - Class 1: The player has played for the national team before
    - Class 2: The player has a history of heart problems
    - Class 3: The player had previous knee injuries
    - Class 4: The player has been a team captain in the past

   What accuracy metric would you use to best evaluate a classification algorithm that predicts the above classes based on data from each player, and **why**?

4. Challenging Questions (Bonus)

1. Explain bias–variance trade-off in regression models
2. When does Kernel Regression outperform Linear Regression
3. Compare L1 vs L2 regularization
    a. When does LASSO perform better?
    b. When does Ridge perform better?
    c. Why does LASSO produce sparsity?
4. Explain why MAPE is unreliable in some datasets
5. Discuss the effect of outliers on regression models
6. Explain the effect of class imbalance on binary metrics. Why is accuracy misleading in imbalanced datasets
7. Explain how the decision boundaries of your models differ fundamentally (bonus)
8. Explain effect of K in KNN
9. Overfitting in Decision Trees:
    a. Why do decision trees overfit easily?
    b. Why is max depth not enough?
    c. How pruning works?
10. Explain why Tree-based models are good feature selectors

11. Micro vs Macro vs Weighted F1
    a. When is Macro F1 a better reflection?
    b. When is Weighted F1 misleading?
    c. Why does Micro F1 favor large classes?
12. Multi-label vs Multiclass
    a. Explain the fundamental difference
        i. Output space
        ii. Loss functions
        iii. Thresholding
        iv. Metrics
    b. Why KNN and Decision Trees can be extended for multi-label classification?
13. Explain precision–recall trade-off
14. Explain ROC vs PR curve
15. If you had unlimited time and resources, how would you improve your models think:
    a. Better preprocessing
    b. Better features
    c. Better models
    d. Better metrics

Contact & Questions

If you have any questions about the assignment, feel free to ask in the **Telegram group**.

If you prefer to contact me directly, you can reach me through:

Telegram: t.me/peyman886

Email: peyman.75.naseri@gmail.com

You can usually find me in the **LLM Lab** during the **afternoons** :)

Final Note

Grading will be **qualitative**, not checklist-based.
Don't focus on ticking boxes, focus on **understanding**, **experimentation**, and **model evaluation**.
By the end of this homework, you should be able to implement, tune, and compare multiple regression and classification models independently.

Due date: Sat, Azar 15, 23:59