**Assignment 1**

**Due October 18, 2024 at 11:59pm**

**This assignment is to be done individually.**

**Important Note:** The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

**DO NOT**:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

**DO**:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment.

**Submitting Your Assignment**

The assignment must be submitted online on Coursys. You must submit a report in **PDF format**. You may typeset your assignment in LaTeX or Word, or submit neatly handwritten and scanned solutions. We will not be able to give credit to solutions that are not legible.

For the last question you are required to submit: A **Python script** (`linreg_submission.py`) containing your complete code for each of the linear regression tasks. You do not need to include the actual Python code in your report, but please provide a clear discussion of your results.

1. **Linear Algebra**

   a) Let $U$ be a subspace of $\mathbb{R}^5$ defined by:

$$U = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \in \mathbb{R}^5 : x_1 = 3x_2 \text{ and } x_4 = 2x_5 \right\}$$

   Find an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$.

   b) Prove that the set $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_2 + \mathbf{u}_3\}$ is linearly independent.

2. **SVD and Eigendecomposition Rotation Matrices and Practical Applications:**

   Consider the matrix $A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 1 \end{bmatrix}$.

   a) Define the matrix $B = AA^\top$ and compute the matrix $B$.

   b) Singular Values from Eigenvalues:
   The Eigendecomposition (EVD) of matrix $B$ is given as:

$$B = U\Lambda U^\top$$

   where $U$ is the matrix of eigenvectors, and $\Lambda$ is the diagonal matrix of eigenvalues.

   - Explain how the eigenvalues of $B$ relate to the singular values of $A$.
   - Compute the singular value matrix $\Sigma$ for $A$, given that the eigenvalues of $B = AA^\top$ are 25, 3, and 0.

   c) Consider an SVD of a matrix $D$ as follows:

$$D = U\Sigma V^\top = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}^\top$$

   A matrix $R_\theta \in \mathbb{R}^{2\times 2}$ is a 2D rotation matrix if it has the following form:

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

   where $\theta \in \mathbb{R}$. Geometrically speaking, $R_\theta \mathbf{v}$ rotates $\mathbf{v}$ counterclockwise by angle $\theta$, for any $\mathbf{v} \in \mathbb{R}^2$, as shown in Figure 1.

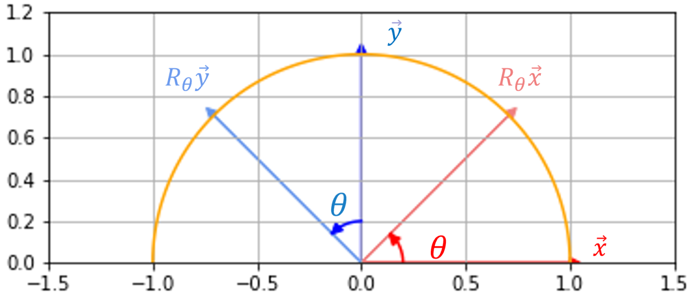

Figure 1: In this case, $\mathbf{x} = (1, 0)$ and $\mathbf{y} = (0, 1)$ are both rotated by $\theta = \frac{\pi}{4}$.

Show that $U$ and $V^\top$ are both rotation matrices and find their corresponding rotational angles $\theta_U$ and $\theta_{V^\top}$.

3. **Taylor Expansion** Given $\vec{x} = \left[x_1, x_2, x_3\right]^\top$, consider a nonlinear function $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$ as follows:

$$f(\vec{x}) = 5x_1^2 + 3x_2^2 + 2x_3^2 + 4x_1x_2 - 2x_1x_3 + 6x_2x_3$$

   a) Compute the **Gradient** and **Hessian matrix** of $f$.
   b) Find the **second order Taylor Expansion** at the point $\vec{x}_0 = [0, 0, 0]^\top$.
   c) State whether $f$ is convex, concave, or neither convex nor concave. Prove your claim.

4. **Convexity** For any $\vec{x}, \vec{y} \in \mathbb{R}^n$ and any $t \in [0, 1]$, a function $f$ is said to be convex if it satisfies any of these conditions:

   - $f(t\vec{x} + (1-t)\vec{y}) \leq tf(\vec{x}) + (1-t)f(\vec{y})$
   - If $f$ is differentiable: $f(\vec{y}) \geq f(\vec{x}) + (\nabla f(\vec{x}))^\top (\vec{y} - \vec{x})$
   - If $f$ is twice differentiable: $\mathrm{H}f(\vec{x}) \succeq 0$

   a) Given $x \in \mathbb{R}$ and only using the definition of convex functions given above, prove that the rectified linear unit function, $\mathrm{ReLU}(x) := \max(x, 0)$, is convex.

5. **Linear Regresssion – House Price Prediction with Polynomial Features and Ridge Regression**

You are working on predicting house prices in a real estate market using a dataset that consists of 500 examples, each with multiple features: number of rooms, house age, area size, etc. The target variable is the house price. You decide to apply linear regression and Ridge regression (a regularized form of linear regression) to build predictive models. Your goal is to assess the generalization of these models using cross-validation and to experiment with different polynomial degrees and regularization strengths to improve the model's performance.

**Data Loading**: Use the following code to load the Boston Housing dataset from GitHub and initialize the DataFrame. For the model, only use the variables rm (number of rooms) and lstat (lower status population, percentage), and the target variable medv (median house value in $1,000s).

```
import pandas as pd

# Load Boston Housing Data from GitHub
url = "https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.csv"
df = pd.read_csv(url)

# Features: ’rm’ (number of rooms) and ’lstat’ (lower status population, percentage)
X = df[[’rm’, ’lstat’]]
y = df[’medv’]  # Target: ’medv’ (median house value in $1,000s)
```

**Implementation Hints**: For managing your dataset, use `pandas` dataframes and for the models and training tools, utilize `scikit-learn`. Plotting should be done with the built-in functions or using `matplotlib`. You can refer to the official scikit-learn documentation for functions like train-test split, cross-validation, linear regression, and Ridge regression.

For coding environments:

- You can work in a Jupyter notebook in `Google Colab`, and export it as a `.py` script for final submission.
- If you are already using `VS Code`, consider using `#%%` cell separators in your Python script, allowing you to run parts of your script like Jupyter notebook cells.

Ensure your script includes the code for each task, such as MSE computation or the best parameter choices, and attach the result outputs to your assignment report in PDF format.

### a. Train-Test Split and Cross-Validation for Linear Regression with Polynomial Features

Split the dataset into a **training set (80%)** and a **test set (20%)**. Use 5-fold cross-validation on the **training set** to evaluate the performance of linear regression models with polynomial features. For each polynomial degree (from 1 to 5), compute the average mean squared error (MSE) over the five folds and report your results.

**Hint**: Use `PolynomialFeatures` from *scikit-learn* to create polynomial features of different degrees. When using the `cross_val_score` function, set the `scoring` parameter to `neg_mean_squared_error`.

### b. Cross-Validation for Ridge Regression with Polynomial Features

Ridge regression introduces a regularization term controlled by a hyperparameter $\alpha$. Perform 5-fold cross-validation on the **training set** with Ridge regression, using polynomial features (degrees 1 to 5) and different values of $\alpha$ (e.g., 0.1, 1, 10, 100). Use grid search to find the best $\alpha$ and the optimal degree, and report both the values that minimize the cross-validation error along with the corresponding average MSE.

**Hint**: Use `GridSearchCV` from *scikit-learn* to automate the search for the best regularization parameter $\alpha$.

### c. Generalization and Test Set Performance

Now that the best polynomial degree and regularization strength have been identified from part (b), train both the linear regression model and the Ridge regression model on the **training set**, and evaluate their MSE on the **test set**. Compare the performance of the two models, and discuss which model generalizes better to unseen data and why.

**Hint**: Use `mean_squared_error` from *scikit-learn* to evaluate the models on the test set.

### d. Theoretical Considerations

Explain why the regularization in Ridge regression helps prevent overfitting, especially when using polynomial features. How does the choice of the regularization parameter $\alpha$ and the polynomial degree influence the model? What might happen if $\alpha$ is too small or too large, or if the degree is too high?