Final Project

CS-UY 4563 - Introduction to Machine Learning

Overview

- **Partner with one student** and select a machine learning problem of your choice.
- **Apply the machine learning techniques** you've learned during the course to your chosen problem.
- **Present your project** to the class at the semester's end.

Submission Requirements on Gradescope

Submit the following on **Gradescope** by the evening before the first presentation (exact date to be announced):

- **Presentation slides.**
- **Project write-up** (PDF format).
- **Project code** as a Jupyter Notebook. If necessary, a GitHub link is acceptable.
- If using a custom dataset, upload it to Gradescope (or provide a GitHub link, if necessary).

Project Guidelines

Write-Up Requirements

Your project write-up should include the following:

1. **Introduction:** Describe your data set and the problem you aim to solve.
2. **Perform some unsupervised analysis:**
   - Explore pattern or structure in the data using clustering and dimensionality (e.g PCA).
   - **Visualize the training data[1]:**
     - Plot individual features to understand their distribution (e.g., histograms or density plots).
     - Plot individual features and their relationship with the target variable.
     - Create a correlation matrix to analyze relationships between features.
   - Discuss any interesting structure is present in the data. If you don't find any interesting structure, describe what you tried.
3. **Supervised analysis:** Train at least three distinct learning models[2] discussed in the class (such as Linear Regression, Logistic Regression, SVM, Neural Networks, CNN).[3]

   For implementation, you may:

   - Use *your own implementation* from homework or developed independently.
   - Use libraries such as *Keras*, *scikit-learn*, or *TensorFlow*.

   For each model,[4] you must:

   - Try different *feature transformations*. You should have at least three transformations. For example, try the polynomial, PCA, or radial-basis function kernel.

     For neural networks, different *architectures* (e.g., neural networks with varying numbers of layers) can also be considered forms of *feature transformations*, as they learn complex representations of the input data.
   - Use different *regularization techniques*. You should have at least 6 different regularization values per model

[1] Do not look at the validation or test data.

[2] You can turn a regression task into a classification task by binning, or for the same dataset, select a different feature as the target for your model. Or you can use SVR.

[3] If you wish to use a model not discussed in class, you must discuss it with me first, or you will not receive any points for that model.

[4] Even if you get a very high accuracy, perform these transformations to see what happens.

4. **Table of Results:**
    - Provide a table with *training* accuracy and *validation* metrics for every model. Include results for the different parameter settings (e.g., different regularization values).
        - For classification include metrics such as precision/recall.
        - For regression modes, report metrics like MSE, R^2. For example, suppose you're using Ridge Regression and manipulating the value of λ. In that case, your table should contain the training and validation accuracy for every lambda value you used.
    - Plot and analyze how performance metrics (like accuracy, precision, recall, MSE) change with different feature transformations, hyperparameters (e.g.regularization settings, learning rate).
5. **Analytical Discussion:**
    - Analyze the experimental results and explain key findings. Provide a chart of your key findings.
    - Highlight the impact of feature transformations, regularization, and other hyperparameters on the model's performance. Refer to the graphs provide in earlier sections to support your analysis. Focus on interpreting:
        - Whether the models overfit or underfit the data.
        - How bias and variance affect performance, and which parameter choices helped achieve better generalization.

Presentation Guidelines

- You and your partner will give a six-minute presentation to the class.
- Presentations will be held during the last 2 or 3 class periods and during the final exam period for this class. You will be assigned a day for your presentation. If we run out of time the day you are to present your project, you will present the next day reserved for presentations.
- **Attendance during all presentations is required.** A part of your project grade will be based on your attendance for everyone else's presentation.

Important Notes on Academic Integrity

- Your submission will undergo plagiarism checks.
- If we suspect you of cheating, you will receive 0 for your final project grade. See the syllabus for additional penalties that may be applied.

Dataset Resources

Below are some resources where you can search for datasets. As a rough guideline, your dataset should have **at least 200 training examples** and **at least 10 features**. You are free to use these resources, look elsewhere, or *create your own dataset*.

- `https://www.kaggle.com/competitions`
- `https://www.openml.org/`
- `https://paperswithcode.com/datasets`
- `https://registry.opendata.aws/`
- `https://dataportals.org/`
- `https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research`
- `https://www.reddit.com/r/datasets/`
- `https://www.quora.com/Where-can-I-find-large-datasets-open-to-the-public`

Modifications

- If you have a project idea that doesn't satisfy all the requirements mentioned above, please inform me, and we can discuss its viability as your final project.
- If you use techniques not covered in class, you must demonstrate your understanding of these ideas.

Brightspace Submissions Guidelines

- **Dataset and Partner:** Submit the link to your chosen dataset and your partner's name by October 30th.
- **Final Submissions:** Upload your presentation slides, project write-up, and code to Gradescope by the evening before the first scheduled presentation. The exact date will be announced once the total number of projects is confirmed. (I expect the due date to be December 4th or December 9th.)

Potential Challenges and Resources

As you work with your dataset, you may encounter specific challenges that require additional techniques or tools. Below are some topics and resources that might be useful. Please explore these topics further through online research.

- **Feature Reduction:** Consider using PCA (which will be covered in class). PCA is especially useful when working with SVMs, as they can be slow with high-dimensional data.

  If you choose to use SelectKBest from scikit-learn, you must understand why it works before you use it.

- **Creating Synthetic Examples:** When using SMOTE or other methods to generate synthetic data, ensure that only real data is used in the validation and test sets.

  - If using synthetic data, make sure your validation set and test set mirrors the true class proportions from the original dataset. A balanced test set for naturally unbalanced data can give misleading impressions of your model's real-world performance. For more details, see: Handling Imbalanced Classes

- **Working with Time Series Data:** For insights on working with time series data, visit: NIST Handbook on Time Series
- **Handling Missing Feature Values:**
  - See Lecture 16 at Stanford STATS 306B
  - Techniques to Handle Missing Data Values
  - How to Handle Missing Data in Python
  - Statistical Imputation for Missing Data
- **Multiclass Classification:**
  - Understanding Softmax in Multiclass Classification
  - Precision and Recall for Multiclass Metrics
- **Optimizers for Neural Networks:** You may use Adam or other optimizers for training neural networks.
- **Centering Image Data with Bounding Boxes:** If you are working with *image data*, you are allowed to use *bounding boxes* to center the objects in your images. You can use libraries like *OpenCV ('cv2')*.

Tips

Don't forget to scale your data as part of preprocessing. Be sure to document any modifications you made, including the *scaling or normalization techniques* you applied.

The following resource might be helpful. Please stick to topics we discussed in class or those mentioned above: CS229: Practical Machine Learning Advice