CS535/EE514: Machine Learning (Fall 2023)
Project

Introduction

The goal of this project is to allow students to apply the knowledge and techniques acquired throughout the course to tackle real-world problems, while paving the way to explore and utilize the tools and methods of **your choosing**.

You will be given the ability to choose from the following topics:

Project 1: Content Moderation and Toxicity Classification
Project 2: Text Similarity and Duplicate Detection
Project 3: Tabular Regression
Project 4: Time Series
Project 5: Research Paper Implementation

For each of the topics, you will be provided a starting dataset and be given a few prompts as starting points for carrying out experiments. As an example, if you had to perform Sentiment Classification, some starting points would be: (1) Naive Bayes, (2) RNNs and Embeddings, (3) Pretrained Transformers like BERT, and (4) LLMs like GPT-3.5. You will be instructed on how many experiments to carry out in the respective project sections - just be sure to have a decent variety (i.e. using BERT-small, BERT-medium, BERT-large don't count as distinct approaches).

You are allowed to:

1. Use any tools and approaches of your choosing, even if it is outside the scope of the class (be creative!). If you want to use APIs, consult the course staff first.
2. Take inspiration from other approaches on the Internet, whether that be articles, research papers, or notebooks on community platforms.

You are not allowed to:

1. Interact across groups.
2. Deviate from the fixed project topics.
3. Employ virtually the *same* approach multiple times: it is your responsibility to consult with the TAs regarding what's permissible and what isn't.

Your project will be graded out of 100 points. To account for any difficulty spikes across projects, we will be performing relative grading on a project basis.

The final deadline for this project is **11:55pm Friday, 1st December 2023**.

Project 1: Content Moderation and Toxicity Classification

Motivation:
In an era where communication happens across digital platforms at an unprecedented scale, ensuring a safe and respectful online environment is a pressing concern. The explosive growth of user-generated content, from social media posts to comments on discussion forums, brings both tremendous opportunities for interaction and the challenge of dealing with harmful and toxic content. Content moderation plays a pivotal role in maintaining the quality and safety of these platforms, and Machine Learning can help automate this responsibility.

Overview:
The goal of this project is to build a model that can help in detecting and flagging content that is (potentially) harmful. This means defining every step of the pipeline: feature extraction, preprocessing, model building, training, and evaluation.
You need to carry out three (3) distinct experiments.

Datasets:
The Jigsaw Toxic Comment Classification Dataset is a good starting point to train a model that can also provide insights into six possible types of comment toxicity: *mild toxicity*, *severe toxicity*, *obscenity*, *threats*, *insults*, *identity hate*.

Propositions:
Some routes to explore, in order of sophistication:

1. Naive Bayes (or any other shallow classifier like Logistic Regression)
2. Sequence Models like RNNs (also making use of Embeddings)
3. Pretrained Encoder-Transformers like BERT and DeBERTa
4. Encoder-Decoder Transformers like T5 and BART
5. Large Language Models like GPT-3.5 and LLaMA-2

Since text can be encoded in many ways, with each model, it is important to research what the inputs should be. For example, one can explore using a simple TF-IDF encoding representation for text to a Logistic Regression model, or use Embeddings derived from pretrained Sentence Transformers. Some other models, especially Transformers, are rigid in how they expect inputs so be careful when preprocessing your data accordingly.

Some tools that can be critical for your project:

- NLTK
- Sentence Transformers
- HuggingFace Models

Submission and Grading Criteria:
You will have to submit the following:

1. Notebooks of each of your 3 approaches (3 approaches x 20 points each = 60 points)
2. Project Report containing details on your problem setup, methodology, findings, evaluation, and best model (20 points)
3. 5 minute video presentation (20 points)

Note that

- Your notebooks will not be graded solely on code and getting done with a task, but also on how well you were able to explain what you were doing, why you were doing it, and if it was better or worse relative to your previous techniques (i.e. a proper quantitative evaluation).
- Your project report should not be more more than 3-4 pages. It should be a standalone demonstration of everything that was done in the project.
- The video presentation should be accompanied with slides, and it is not necessary for everyone to present.

You will upload **one zip file** to Google Drive, sharing the link with us. The zip file will contain the following structure:

```
├── Gxx_Project1.zip
│ └── Gxx_NB1.ipynb
│ └── Gxx_NB2.ipynb
│ └── Gxx_NB3.ipynb
│ └── Gxx_Report.pdf
│ └── Gxx_Presentation.mp4
```

Deviations from the structure will incur an incontestable penalty.

Project 2: Text Similarity and Duplicate Detection

Motivation:
In an age of information abundance, the ability to find commonalities and identify duplications within vast volumes of textual data is more critical than ever. Whether you're managing vast databases of information, curating content, or enhancing search capabilities, the capacity to discern text similarities and detect duplicates is a superpower. Some interesting applications include: tagging duplicate posts on platforms like Stack Overflow and Quora; finding out whether two student essays are just reworded versions of one another, etc. The techniques can go far beyond simple syntactic string matching, and Machine Learning can help us with that.

Overview:
The goal of this project is to build a model that is able to classify and flag whether two pieces of content are duplicates of one another. This means defining every step of the pipeline: feature extraction, preprocessing, model building, training, and evaluation. You need to carry out three (3) distinct experiments.

Datasets:
The nature of the model is highly dependent on the dataset used: one solid starting point is to use the Quora Question Pairs Dataset, which is a collection of pairs of questions from Quora with labels indicating whether the two are duplicates or not. Another route is to use the US Patent Phrase to Phrase Matching dataset, which contains the (binned) similarity scores between short descriptions of patents.

Propositions:
Some routes to explore, in order of sophistication:

1. Naive Bayes (or any other shallow classifier like Logistic Regression)
2. Sequence Models like RNNs (also making use of Embeddings)
3. Pretrained Encoder-Transformers like BERT and DeBERTa
4. Encoder-Decoder Transformers like T5 and BART
5. Large Language Models like GPT-3.5 and LLaMA-2

Since text can be encoded in many ways, with each model, it is important to research what the inputs should be. For example, one can explore using a simple TF-IDF encoding representation for text to a Logistic Regression model, or use Embeddings derived from pretrained Sentence Transformers. Some other models, especially Transformers, are rigid in how they expect inputs so be careful when preprocessing your data accordingly.

Additional Note: Since you have to deal with pairs of text samples, you may want to experiment with aggregating representations - maybe horizontally stacking the TF-IDF representations, or taking the difference of the Embeddings from a Sentence Transformer, before passing into a Logistic Regression model.

Some tools that can be critical for your project:

- NLTK
- Sentence Transformers
- HuggingFace Models

Submission and Grading Criteria:
You will have to submit the following:

1. Notebooks of each of your 3 approaches (3 approaches x 20 points each = 60 points)
2. Project Report containing details on your problem setup, methodology, findings, evaluation, and best model (20 points)
3. 5 minute video presentation (20 points)

Note that

- Your notebooks will not be graded solely on code and getting done with a task, but also on how well you were able to explain what you were doing, why you were doing it, and if it was better or worse relative to your previous techniques (i.e. a proper quantitative evaluation).
- Your project report should not be more more than 3-4 pages. It should be a standalone demonstration of everything that was done in the project.
- The video presentation should be accompanied with slides, and it is not necessary for everyone to present.

You will upload **one zip file** to Google Drive, sharing the link with us. The zip file will contain the following structure:

```
├── Gxx_Project2.zip
│  └── Gxx_NB1.ipynb
│  └── Gxx_NB2.ipynb
│  └── Gxx_NB3.ipynb
│  └── Gxx_Report.pdf
│  └── Gxx_Presentation.mp4
```

Deviations from the structure will incur an incontestable penalty.

Project 3: Tabular Regression

Motivation:
Tabular Regression is a timeless cornerstone in the area of Data Analysis and Machine Learning. Being able to iterate and experiment with all sorts of different techniques on structured data is a gateway to mastering this craft - usually it's not as simple as throwing the same 5 techniques at every dataset, but being adaptable to what unique things can be exploited with the problem at hand. What you learn here applies to a sea of domains, and the applications are invaluable.

Overview:
The goal of this project is to experiment and come up with the best model you can, for predicting the prices of Bulldozers given a rich feature set (described in the dataset below). Having to work on tabular data: preprocessing (cleaning your data, normalizing, encoding categorical features etc.), feature selection, model selection, a proper evaluation, are all necessary components.

Datasets:
You will be working with the dataset from the Blue Book for Bulldozers Competition. You are not allowed to work with anything else, or augment the existing dataset with new data.
You need to carry out five (5) distinct experiments.

Propositions:
There's a lot of room to mix and match techniques across model building and preprocessing, so each experiment you carry out can be a unique combination of each.

- In regards to data preprocessing, you can look into different techniques for dealing with missing values if there are any, how you can best normalize your data for your model, how you can best encode specific categorical variables etc.).
- Another rich area for exploration is feature engineering: does combining different features or transforming them in a certain way lead to better results? Is it worthwhile to see whether certain features are redundant and are hurting model performance? Is there something unique about the dataset (like datetime features) that can be used to extract valuable features for your model to use?
- The fun part is building a model- you can experiment with any model you want. If you want an easy way out, use a KNN model that doesn't require much data preprocessing to function normally, or go with something more robust like an SVM or Neural Network that is like a "hammer to every nail" type of model. Or go really fancy and make your own Ensemble of these models, maybe stacking them, or bagging their predictions.

Just be sure there's enough variety in these 5 experiments to make them distinct from one another.

Submission and Grading Criteria:
You will have to submit the following:

1. Notebooks of each of your 5 approaches (5 approaches x 12 points each = 60 points)
2. Project Report containing details on your problem setup, methodology, findings, evaluation, and best model (20 points)
3. 5 minute video presentation (20 points)

Note that

- Your notebooks will not be graded solely on code and getting done with a task, but also on how well you were able to explain what you were doing, why you were doing it, and if it was better or worse relative to your previous techniques (i.e. a proper quantitative evaluation).
- Your project report should not be more more than 3-4 pages. It should be a standalone demonstration of everything that was done in the project.
- The video presentation should be accompanied with slides, and it is not necessary for everyone to present.

You will upload **one zip file** to Google Drive, sharing the link with us. The zip file will contain the following structure:

```
├── Gxx_Project3.zip
│ └── Gxx_NB1.ipynb
│ └── Gxx_NB2.ipynb
│ └── Gxx_NB3.ipynb
│ └── Gxx_NB4.ipynb
│ └── Gxx_NB5.ipynb
│ └── Gxx_Report.pdf
│ └── Gxx_Presentation.mp4
```

Deviations from the structure will incur an incontestable penalty.

Project 4: Tabular Time Series

Motivation:
Time Series prediction stands at the intersection of history and foresight. It's the art and science of gleaning insights from past events to make accurate predictions about the future. The techniques applied to one Time Series prediction task can be generalized across a huge variety of other problems, and blends well into a cornucopia of relevant applications in both the industry and academic research. Machine Learning will let you tackle this predictive modeling task with lots of tools in your belt.

Overview:
The goal of this project is to build a model that is able to forecast electricity prices (for the dataset discussed below). Having to work on tabular data: preprocessing (cleaning your data, normalizing, encoding categorical features etc.), feature selection, model selection, a proper evaluation, are all necessary components.

Datasets:
You will be working with the Panama Electricity Load Forecasting Dataset. You are not allowed to work with anything else, or augment the existing dataset with new data.
You need to carry out five (5) distinct experiments.

Propositions:
Time Series problems give you a lot of opportunity to play around. Most experiments in the context of Time Series revolve around Feature Engineering, so here are some pointers for starters:

1. Using only the features to predict the target at the current step using a model like KNNs, Linear Regression or Decision Trees (similar to a normal regression problem).
2. Creating a fixed window of previous values of the target as features (*lagged features* to be specific) to the model (like those above), putting aside everything else.
3. Exploiting the sequential nature of the task and modeling with the same feature set as (2), but with Sequence Models like RNNs and LSTMs.
4. Blending (1) and (2) to get the lagged features *and* other features as input to the model.
5. Using a model designed specifically for Time Series like Prophet or ARIMA.

You can read up on articles specific to feature engineering for Time Series, like this and this.

Submission and Grading Criteria:
You will have to submit the following:

1. Notebooks of each of your 5 approaches (5 approaches x 12 points each = 60 points)
2. Project Report containing details on your problem setup, methodology, findings, evaluation, and best model (20 points)
3. 5 minute video presentation (20 points)

Note that

- Your notebooks will not be graded solely on code and getting done with a task, but also on how well you were able to explain what you were doing, why you were doing it, and if it was better or worse relative to your previous techniques (i.e. a proper quantitative evaluation).
- Your project report should not be more more than 3-4 pages. It should be a standalone demonstration of everything that was done in the project.
- The video presentation should be accompanied with slides, and it is not necessary for everyone to present.

You will upload **one zip file** to Google Drive, sharing the link with us. The zip file will contain the following structure:

```
├── Gxx_Project4.zip
│ └── Gxx_NB1.ipynb
│ └── Gxx_NB2.ipynb
│ └── Gxx_NB3.ipynb
│ └── Gxx_NB4.ipynb
│ └── Gxx_NB5.ipynb
│ └── Gxx_Report.pdf
│ └── Gxx_Presentation.mp4
```

Deviations from the structure will incur an incontestable penalty.

Project 5: Research Paper Implementation

Motivation:
Implementing research papers is a very useful skill in the field of Machine Learning. Oftentimes, simply being able to understand the main takeaways of the paper is not enough - the real pressure test is being able to implement it in code and replicate the results, before being able to think upon making improvements. This project is an opportunity to develop an important skill, and learn a very useful technique along the way.

Overview:
The goal of this project is to replicate the techniques used in, and the results from, the paper "Entity Embeddings of Categorical Variables" by Guo and Berkahn. Following this, you will *try* to beat the scores they have reported, using your own techniques (only one attempt is required).
Note that you are **not allowed to use TensorFlow or Keras** in this project, and that all your work will be in notebooks with comments and Markdown explanations behind everything that's happening (and relating to the paper content), otherwise you will not be awarded points for it.

Datasets:
Since you have to follow the paper's techniques and methodology, do not stray from using the dataset the authors have used.

Propositions:
In replicating results, there's not much room to deviate from the core methodology. Still, in trying to beat the scores, feel free to use any models you want or change up the preprocessing techniques and hyperparameters.

Submission and Grading Criteria:
You will have to submit the following:

1. Notebook of the implementation (NB1 - 40 points)
2. Notebook of your group's attempt to beat the score (NB2 - 20 points)
3. Project Report containing details on your problem setup, methodology, findings, evaluation, and best model (20 points)
4. 5 minute video presentation (20 points)

Note that

- Your notebooks will not be graded solely on code and getting done with a task, but also on how well you were able to explain what you were doing, why you were doing it, and if it was better or worse relative to your previous techniques.
- Your project report should not be more more than 3-4 pages. It should be a standalone demonstration of everything that was done in the project.
- The video presentation should be accompanied with slides, and it is not necessary for everyone to present.

You will upload **one zip file** to Google Drive, sharing the link with us. The zip file will contain the following structure:

```
├── Gxx_Project5.zip
│  └── Gxx_NB1.ipynb
│  └── Gxx_NB2.ipynb
│  └── Gxx_Report.pdf
│  └── Gxx_Presentation.mp4
```

Deviations from the structure will incur an incontestable penalty.