|  |  |
| --- | --- |
| **CSCI 4364/6364 Machine Learning**  **Fall 2023**  Section 81 |  |

ML Semester Project Proposal Template[[1]](#footnote-1)

Please make a copy of the template, review and delete the instructions and the examples in gray and populate each of the fields in regular black font.

**Team Name:** Pick a witty name for your team that describes that problem you’re solving.

**Project Team Members (1-5 members):**

|  |  |
| --- | --- |
| **Name** | **GWID** |
| **Toghrul Tahirov** |  |
| **Yusif Mukhtarov** |  |
| **Aytan** |  |
| **Shamil** |  |
| **Tural Mehtiyev** |  |

**Document Version History**

|  |  |
| --- | --- |
| **Date** | **Description** |
| 10/22/2023 | Initial Draft Submitted **Deadline 10/17/2023** |
|  | Instructor Feedback 1 |
|  | Edits and updates from Feedback 1 |
|  | Approval from instructor |
|  | Final Report Submitted **Deadline 12/14/2023** |

|  |
| --- |
| **Project Guidelines (30% of Grade)**  The objective of the ML project is to gain hands-on experience in solving an interesting problem using Machine Learning. You may work individually or as a team (up to five).  The problem should include supervised, semi-supervised, or reinforcement learning methods.  If you are already pursuing research in ML, you may choose a particular aspect related to your research.  You may focus on a specific method or algorithm and evaluate its performance against baseline datasets (i.e., new activation function, architecture, distance measure, optimizer, etc.). Alternatively, you may apply ML methods on a specific real-world application (e.g., real-time trends analysis of streaming media).  At the end of the course, teams will present a 10-minute overview with demo to the class, and submit a final report and source code.  I am happy to address any specific questions during office hours or by appointment. |

# 1 Analysis

Add foundational background information about your project. It should be sufficiently detailed to give the reader understanding of what problem you’re trying to solve and how your approach is different from others who have tried something similar.

## 1.1 Problem Description:

Describe the context of the problem, why it’s important, and how ML might solve this problem.

**Example:**

*During Covid lockdowns and Work From Home (WFH), we observed a 30% increase in the number of chat message, and employees are increasingly struggling with keeping up with conversations across multiple chat rooms. We would like to use text summarization to summarize the dialogs to help users manage their chat conversations.*

***Problem Description****: Can abstractive text summarizers be used for chat conversation summaries?*

## 1.2 Performance Criteria:

State a few top-level project requirements your solution should fulfill (limit to no more than 5 or 6):

*Example:*

* *Summaries should contain the essential facts of the conversations.*
* *Summaries should be grammatically correct and free of offensive language.*
* *Summaries should be generated for chat session with three or more messages that occur in a window of five minutes.*
* *Summaries should be no more than 100 words.*
* *…*

## 1.3 Related Work:

Here, you should describe the domain broadly, and how others have attempted the same or similar problem. Generally, you’ll want to list papers from peer-reviewed conferences (NeurIPS, ICML, ICLR, etc.) or journals (JMLR, AI, etc.). You may use blogs and on-line sources as secondary related work. If you are trying to solve a problem for industry, you may also refer to the work other teams have done.

You’ll want to highlight and define the **Machine Learning Task**.

**Example:**

*Text summarization aims at generating accurate and concise summaries from input document(s). In contrast to extractive summarization which merely copies informative fragments from the input, abstractive summarization may generate novel words. A good abstractive summary covers principal information in the input and is linguistically fluent (Zhang, Zhao, Saleh, & Liu, 2020). …*

## 1.2 Project Objective:

The project objective is a statement of what the project is to achieve, by which success or failure of the project can be evaluated.

***Example:***

*Create a chat summarization model that can provide accurate, and syntactically correct summaries for at least 80% of chat conversations.*

# 2 Hypothesis

# 2.1 Methodology

# Forecasting the future, especially 48 hours ahead, is inherently challenging. While predicting the future with certainty is impossible, modeling and estimating it based on past patterns is feasible. We seek to leverage the seasonality and stationary inherent to the data to make accurate forecasts. Below, we provide a list of both the Statistical Machine Learning and Deep Learning models chosen for this task, alongside the rationale for their inclusion.

### 2.1.1 Statistical Machine Learning Models

# ARIMA and SARIMA: Autoregressive Integrated Moving Average and Seasonal Autoregressive Integrated Moving Average are conventional methods for time series forecasting. They perform optimally on stationary data and can capture trend and seasonality. These models will essentially serve as a foundation and provide a benchmark for evaluation of advanced models.

# Statistical/Neural Prophet: This duo is an evolution from traditional ARIMA models. While the statistical variant employs a generalized additive model (GAM) to fit data, its neural counterpart utilizes a neural network with the same core philosophy.

# Gradient Boosting Algorithms: Predominantly tree-based models, they are adept at handling large datasets and a variety of feature types. The fact that these models incorporate many weak-learners ensures a reduction in overfitting, while their boosting mechanism focuses on errors, which improves the forecast accuracy.

### 2.1.2 Deep Learning Models

# **LSTM (Long Short-Term Memory)**: A type of recurrent neural network (RNN) explicitly designed for time series and sequences.

# **TCN (Temporal Convolutional Network)**: Adapts CNNs for sequential data and offers advantages over RNNs in terms of parallelism and longer memory.

# **Transformers (Temporal Fusion Transformer - TFT)**: The TFT leverages the self-attention mechanism from transformers, making it capable of focusing on different time steps of the input sequence, capturing long-term dependencies and relationships in the data.

## 2. 2. Data Description

# **Source**[: NeurIPS 2023 Citylearn Challenge](https://www.aicrowd.com/challenges/neurips-2023-citylearn-challenge/problems/forecasting-track-citylearn-challenge)

# **Number of examples**: 720 hourly samples for each of the 3 buildings and 2 neighborhoods level datasets

# **Features**: The data consists of features pertinent to building level energy loads, carbon intensity, and solar generation.

# **Labels**: Continuous labels corresponding to the below variables:

# **Building-Level Forecasts:**

# Cooling Load (kWh)

# DHW Load (kWh)

# Equipment Electric Power (kWh) (commonly known as non-shiftable load)

# Note: Each of these variables will be individually forecasted for every building.

# **Neighborhood-Level Forecasts**:

# Carbon Intensity (kgCO2e/kWh)

# Solar Generation (W/kW)

# Here, both variables will be forecasted collectively for the entire neighborhood.

# Note: I confirm that the data used for this project is neither confidential, private, nor proprietary and will be shared as part of the final delivery.

## 2. 3 Experiment

# To evaluate the performance of our models, we'll employ a rolling forecast origin methodology, ensuring our model's adaptability to new data points. Additionally, we will split the data into training and test sets to validate the model's performance on unseen data.

## 2.4 Metrics

# Mean Absolute Error (MAE): Represents the average error magnitude.

# Root Mean Squared Error (RMSE): Emphasizes larger errors.

# Mean Absolute Percentage Error (MAPE): Provides a relative error measure.

# Through rigorous testing and comparison between the models, we aim to pinpoint the most suitable approach for our 48-hour forecast.

# 3 Synthesis

Discuss the most important software, ML libraries, and other open-source tools that you plan on using in the project. How will you tune hyperparameters, etc.

Subdivide the project into phases and/or subcomponents and estimate the number of days of effort required to accomplish the task.

While its not required, I strongly encourage including a project work plan with tasks and expected dates to help track your project.

# 4 Validation

## 4.1 Results:

Summarize how you will compute the raw results for each experiment and describe visualizations and tables you intend to include in the final report.

You should also demonstrate learning as loss function curves (e.g., Tensorboard) that converge over many training iterations.

## 4.2 Conclusions:

**Formal Conclusions**: Formal conclusions determine whether your project objective was achieved and require statistical measures to quantify uncertainty and confidence. Propose and justify a statistical test/method (e.g., Mann-Whitney, Kolmogorov-Smirnov, etc.) to determine if the original research hypotheses passes or fails. For formal conclusions, you should provide confidence statistics (i.e., p-values) from your statistical tests.

**Informal Observations**: Include additional conclusions relevant to your work that may be based on rigorous analysis or may be insightful and speculative. There is no need to include any remarks about informal observations in the proposal, but please include them in the final report.

**Future Work**: Not added in the proposal, but in the final report you should include a discussion about what promising avenues your work may lead to, even if they are not immediately aligned with your original objectives. If you didn’t achieve the original objective, address what might be done with more time and effort to achieve the objective.

1. Template derived from Bock, P. (2001) Getting it Right: R&D Methods for Science and Engineering, Academic Press [↑](#footnote-ref-1)