Department of Computer Science
School of Systems and Technology
University of Management and Technology

CS446 Machine Learning
Spring 2023

Project-2
Apply Machine Learning to a Real-world Problem

Total Marks: 100

Contribution to Final Assessment: 10%

Submission Date: May 31, 2023 @5pm

---

1 Objectives

In this course project-2, we will serve the following objectives:

- Learn to extract the features from audio (speech) recordings or images
- Understand the role of feature extraction in machine learning
- Learn to use the different algorithms covered in the course to the problems of

  Project 1: Sentiment Analysis from Audio Recordings

  Project 2: Surface Defect Detection and Classification from Images

  Project 3: Reproduce Results of a Research Paper

  Project 4: Vehicle Detection and Classification from images

  Project 5: Predicting the Results of English Premier League (EPL)

- Learn to evaluate and compare the performance of different algorithms
- Learn to document, report and present the solution of a machine learning problem

We assume that the execution of the project will help the retention of the material and significantly enhance the depth of your understanding.

We require you to work in a group of (maximum) three students (every student in the group will receive same score).

Work on only one project.

2 Project 1 - Sentiment Analysis from Audio Recordings

2.1 Motivation

Sentiment analysis from audio recordings is the process of using natural language processing techniques to analyze the emotional tone of spoken language in audio files. This can be done using machine learning algorithms that identify patterns and features in the audio, such as intonation, pitch, and tempo, to infer the underlying sentiment of the speaker. Sentiment analysis from audio recordings has a wide range of applications, including in customer service, where it can be used to monitor customer sentiment and satisfaction during phone calls, and in social media monitoring, where it can help to identify trends and sentiment around specific topics. It can also be used in the entertainment industry, where sentiment analysis can help to gauge audience reactions to films, TV shows, and other forms of media.

2.2 Project Overview

In this project, you will work on the development of a classifier to identify the emotion/sentiment for each recorded audio. You will first remove noise from the audio files and then extract the features of the given audio signal and use these features for classification.

2.3 Data Set Description

CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified).

The actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral and Sad) and four different emotion levels (Low, Medium, High and Unspecified).

You can access the dataset from here: link https://github.com/Now33d/SentimentAnalysisDataset

2.4 Project Requirements

Before processing the audio files, you are required to select audio files from the dataset. You must create training, validation and testing sets that are best suited to train and evaluate your classifier(s). Then remove the noise from the selected audio files using low pass filtering.

We require you to review the literature and identify a list of features that can be used for synthetic speech attribution. For your convenience, we have listed below the features that can be potentially used for the problem under consideration.

- Fourier transform
- Mel frequency cepstral coefficients (MFCCs): Features that describe the overall shape of a spectral envelope of the signal.

- Melspectrogram : A spectrogram where the signal frequencies are converted to Mel scale.
- Chromagram: Also known as chroma features, this is a representation for audio or speech signal in which the entire signal spectrum is projected onto 12 bins known as pitch classes.
- Bicoherence: Mean, variance, skewness and kutrosis of magnitude and phase of bicoherence.
- Spectral centroid: Location of center of mass of the spectrum.
- Spectral bandwidth: Difference between the highest and the lowest frequency in the spectrum.
- Spectral contrast: The decibel difference between peaks and valleys in the spectrum.

2.5 Classifiers

Followed by feature extraction, we require you to evaluate the performance of following different classifiers.

- K-Nearest Neighbors
- Logistic Regression
- Naive Bayes
- Support Vector Machines
- Neural Network

You are allowed to use Scikit-learn implementations of the algorithms.

3 Project 2 - Surface Defect Detection and Classification from Images

3.1 Motivation

In recent years, thanks to advances in hardware technologies and the equally growing need for real-time quality inspection of production processes, there is a demand for artificially intelligent solutions geared towards improving the robustness of machine vision systems. Defect detection on surface using machine vision involves the use of computer algorithms and image processing techniques to automatically detect and classify defects on the surface of a material. This technology can be applied in various industries, such as manufacturing, quality control, and automotive, where the detection of surface defects is crucial for ensuring the safety and functionality of the product. The machine vision system captures images of the material's surface and analyzes them to identify anomalies such as scratches, dents, and cracks. By automating the defect detection process, machine vision can significantly improve production efficiency and reduce the likelihood of human error. It can also save time and resources compared to manual inspection, which is more time-consuming and prone to errors. Additionally, defect detection using machine vision can provide valuable data that can be used to optimize production processes and prevent future defects from occurring.

Feature extraction and learning is an essential part of machine vision system these days. Although most surface defect detection or localization techniques out there employ self-adaptive neural networks, nevertheless, it is important to establish an acute understanding of the classical approach to isolate an object in an image or a video stream. This project will give you a hands-on comprehension of the way in which a feature identification pipeline is designed and use the features for classification tasks.

3.2 Project Overview

In this project, you will work on the development of a classifier to detect defects in an image. This can be interpreted as a multi-class classification problem in which we have three classes: no defect, hole and mark. You will extract features of the object from the given data-set to train your classifier. And use the trained classifier to search for defects in images of a reel of paper.

3.3 Data Set Description

In the data-set, there are 1500 images of a reel of paper. The images belong to three classes,

- defect-free paper
- paper with a hole
- paper with a pen mark on it

You can access the dataset from here: link
https://drive.google.com/drive/folders/13wio89bUATpnMWuEFYZHN0lYIn9-n23E
You are encouraged to explore data augmentation to get more training data (optional).

3.4 Feature Extraction and Object Localization

For your convenience, we have listed below the features that can be potentially used for the problem under consideration. We require you to review the literature and identify more features that can be used for defect detection and localization.

- Colour Histogram
- Spatial Binning
- Histogram of Oriented Gradients

3.5 Classifiers

Followed by feature extraction, we require you to evaluate the performance of the following different classifiers.

- Support Vector Machines
- Naive Bayes
- Logistic Regression
- Neural Network

You will use these trained classifiers to perform two tasks.

(i) three-class classification: classify individual images into the three classes i.e., paper, paper with hole, or paper with pen mark.

(ii) defect localization: in the images with defects, you are required to find the location of the defect and draw a bounding box around it.

You are allowed to use Scikit-learn (or any other library you prefer) implementations of the algorithms.

4 Project 3 - Reproduce Research Paper Results

4.1 Overview

In this project, we require you to reproduce the results of either of the following research papers.

Paper 1: Learning to Learn without Gradient Descent by Gradient Descent, ICML 2016

Overview: The aim of this project is to reproduce the results of meta-learning algorithm proposed in the paper that learns to learn by optimizing an inner loss function using a separate meta-learner.

Dataset: You can use the MNIST dataset, which consists of 60,000 28x28 grayscale images of handwritten digits.

Methodology:

- Reproduce the Baseline Model: Start by reproducing a baseline model, such as a simple convolutional neural network, trained on the MNIST dataset.
- Implement the Meta-Learning Algorithm: Implement the meta-learning algorithm proposed in the paper, which involves training a separate meta-learner to optimize the parameters of the inner learner, which is the baseline model trained on the MNIST dataset.
- Train and Evaluate the Models: Train the baseline model and the model with the meta-learning algorithm and evaluate their performance on the test set of the MNIST dataset.
- Compare and Analyze the Results: Compare the performance of the baseline model and the model with the meta-learning algorithm and analyze the results to understand the impact of the meta-learning algorithm on the performance of the model.

Paper 2: Variational Autoencoder for Semi-Supervised Text Classification, AAAI, 2019

Overview: The aim of this project is to reproduce the results of the paper that proposes a variational autoencoder-based approach for semi-supervised text classification, which uses a small amount of labeled data and a large amount of unlabeled data to improve the performance of text classification.

Dataset: You can use the 20 Newsgroups dataset, which consists of 20,000 newsgroup documents across 20 different categories.

Methodology:

- Implement the Baseline Model: Start by implementing a baseline model, such as a simple bag-of-words model, trained on the labeled data of the 20 Newsgroups dataset.
- Implement the Variational Autoencoder: Implement the variational autoencoder-based approach proposed in the paper, which uses the labeled and unlabeled data

to train a deep generative model that can generate latent representations of the input documents.

- Train and Evaluate the Models: Train the baseline model and the model with the variational autoencoder and evaluate their performance on the test set of the 20 Newsgroups dataset.
- Compare and Analyze the Results: Compare the performance of the baseline model and the model with the variational autoencoder and analyze the results to understand the impact of the variational autoencoder on the performance of the model, as well as its ability to leverage the unlabeled data for semi-supervised learning.

5 Project 4 - Vehicle Detection and Classification from Images

5.1 Motivation

In recent years, thanks to advances in hardware technologies and the equally growing need for traffic monitoring, there is a demand for artificially intelligent solutions geared towards improving surveillance of traffic in urban areas. Vehicle detection is a crucial step in these solutions, which is what you will be exploring in this project.

Feature extraction and learning is an essential part of object detection. Although most object detection techniques out there employ self-adaptive neural networks, nevertheless, it is important to establish an acute understanding of the classical approach to isolate an object in an image or a video stream. This project will give you a hands-on comprehension of the way in which a feature identification pipeline is designed and use the features for classification tasks.

5.2 Project Overview

In this project, you will work on the development of a classifier to detect a vehicle (car or motorbike) in a video stream. This can be interpreted as a multi-class classification problem in which we have three classes: no vehicle, car and motorbike.

You will extract features of the object from the given data-set and implement a sliding window technique to use your trained classifier to search for vehicles in images.

5.3 Data Set Description

In the data-set, there are 5000 images collected from various sources. You may add your own images to the data-set should you feel the need to do so. Get the dataset from instructor's office.

5.4 Feature Extraction and Object Localization

For your convenience, we have listed below the features that can be potentially used for the problem under consideration. We require you to review the literature and identify more features that can be used for object (vehicle) detection and localization.

- Colour Histogram
- Spatial Binning
- Histogram of Oriented Gradients

You will use these trained classifiers to localize objects (car or motorbike) in the sliding window.

5.5 Classifiers

Followed by feature extraction, we require you to evaluate the performance of following different classifiers.

- Support Vector Machines

- Naive Bayes
- Logistic Regression
- Neural Network

You are allowed to use Scikit-learn implementations of the algorithms.

6 Project-5
Predicting the Results of English Premier League (EPL)

6.1 Objectives

The objective of the project is to apply the different classification algorithms covered in the course to the problem of predicting the results of the English Premier League (EPL). This will help your retention of the material and significantly enhance the depth of your understanding. This will also develop your skills in reporting the performance of different machine learning algorithms. The project is expected to consume roughly two weeks of moderately concentrated effort. We encourage you to work in a group of (maximum) three students (every student in the group will receive same score).

6.2 Data Set Description

The data-set provided comprises of two main parts: match statistics and final stand-ings.

All data set files are in CSV format. Get dataset from Instructor's office.

6.2.1 Match Statistics

The match statistics data is organized into separate files for each year, and within each file, the rows are sorted according to match date. There are a total of 20 .csv files, one for each year from 2000-2019. The feature columns are explained in the text file provided, as shown below:

```
Attendance = Crowd Attendance
Referee = Match Referee
HS = Home Team Shots
AS = Away Team Shots
HST = Home Team Shots on Target
AST = Away Team Shots on Target
.
.
.
```

Be careful, feature columns are not uniform across all years! You will have to process them accordingly.

6.2.2 Final Standings

We also have all the final team standings for each year consolidated into a single file where each row represents a team. In this part, each attribute is simply the year of the standing. There are 43 teams and their finishing position for each year from 2000 till 2016.

6.3 Scope of Work

For the given data-set, we want to develop classifiers for the prediction the outcome of a match between two select teams, given their attributes. You will need to develop your own test and validation data accordingly.

The scope of the project includes

- formulation of the problem under consideration.
- cleaning and pre-processing the data.
- apply feature engineering (if needed).
- implement the following classifiers kNN, logistic regression, Bayes classifier, SVM, neural network for the problem under consideration. You are allowed to use Scikit-learn implementations of the algorithms.
- report the performance of different classifiers and presentation of analysis/findings.

Dataset Explanation

For a start, the dataset is divided into two parts:

a) Match-wise stats of nearly for each season in a CSV file with the name of that season. For example, the stats of all the matches in the 2017-2018 season would be in the file with that name.
b) A CSV file containing the standings of teams across nearly 20 years of the English Premier League.

Let us look at the standings file first. It is a fairly simple file with each row representing a team, and each column representing a season, with the number corresponding to where they finished in the league from 1-20. If a cell is empty, it means that the team did not participate in the English Premier League that season. For the purposes of a match predictor, a team that consistently finishes above the other team is **more likely** to win their match. This is not a surety, but if you can find a way to use it as a feature for your classifier(s), it can potentially be useful.

The match-wise stats file is where you will get most of your data/features from. Keep in mind that you are **not** allowed to use any of the stats from the match you are trying to predict. You can only use the betting odds of that particular match. You can find the full form of the abbreviations in the file "ML EPL Dataset-Explanation" but let us go through some of the data you have at your disposal. Every match has specified which team is the home team and which team is the away team. In football, there is a "myth" that home teams generally have an advantage over the away team, so that piece of information **might** be useful for your classifier(s). For each win, a team gets 3 points, for each loss 0 points, and for each draw 1 point. Teams that have a significant point advantage over the other teams are more likely to win. In the dataset, you are not given the points of a team at a certain point in time, but you can very easily calculate them if you have the results of all the matches of the season till that point in time. Furthermore, attributes such as a high number of goals scored, and a low number of goals conceded before the match we are trying to predict might have a big impact on the outcome of the match. Another thing that can potentially impact a team's chances of winning is the form they are in. Think of it like this: if a team has won their last 5 matches, and they are facing a team that has lost the last 3 and drawn 2, which team is more likely to win? The data does not explicitly have the form of a team, but you can very easily figure that out.

Here is a mind map of how you can potentially tackle the problem at hand:

- **Engineer the features:** The data is fairly raw; you will need to do a lot of work to clean it and get meaningful features out of it. Do a bit of research regarding what can contribute to a team having an advantage over the other, and how you can generate that information from the provided data.

- **Train your model:** In layman's terms, assuming the aforementioned features are used, you are essentially telling your model that if a team is on a winning streak, having a significant advantage over the other team in terms of points, and has scored a high number of goals in the past *x* matches, it is more likely to win. Choose a window of matches and use their data to train your model. Keep in mind that if you train your model to check the form of a team over the past 5 matches, you will have to do the same in your testing phase. Consistency in the train and test data is the key.

- **Test your model:** As mentioned earlier, the test data should be in exactly the same shape as the train data. One way of approaching this could be to try and run your classifier for each season separately, since the data across seasons can be inconsistent.

We understand that this must be a lot of information to process, but if you can divide your project into the three components, it can be very easily tackled. If you have any queries, please feel free to reach out. Best of luck!

7 Expectations and Scope of Work

7.1 Scope of Work

The scope of work for Project 2 includes

Task 1: formulation of the problem under consideration

Task 2: carry out literature review to identify the features that can be used for the problem under consideration

Task 3: apply feature extraction/engineering

Task 4: apply dimensionality reduction (if needed or to evaluate the impact of reduction on the performance)

Task 5: implement the different classifiers for the problem under consideration.

Task 6: report the performance of different classifiers and presentation of analysis/findings

7.2 Submission and Assessment

There are three components of assessment in the project:

- Project report(PDF) must be based on Overleaf.com using LaTeX with shared link (40 marks)
- Project code (along with code documentation, ipynb file plus Google CoLab link) (30 marks)
- 15 minutes video presentation with YouTube link summarizing your work (20 marks)
- Submission of deliverables on time (10 marks)

We encourage you to use a template from your favorite machine learning conference (e.g., NIPS or ICML) or IEEE Access for report . Your report is expected to have the following sections:

- Abstract (executive summary)
- Introduction
- Mathematical Formulation
- Identification and Extraction of Features
- Feature Engineering e.g., dimensionality reduction (optional)
- Performance Comparison of various classification algorithms as mentioned
- Performance Evaluation (plots, tables, metrics etc.), analysis and findings
- Conclusions
- References
- Authors' Biography and Photo