CSP 554 - BIG DATA TECHNOLOGY

Intermediate Draft

MACHINE LEARNING USING SAGE-MAKER

Group Members:

Aakef Waris (A20420535)
Amandeep Singh Oberoi (A20466752)
Amit Nikam (A20470263)
Rutul Mehta (A20476293)

Project Overview:

The core problem that is being solved is sentiment analysis through classification. Given an amazon review, our target model will be able to classify whether or not it is a good or bad review. The ground truth comes from the user's star rating. If the user gives 4 or 5 stars their review will be deemed as positive or a good review, and if they give one or 2 stars then it will be a bad review.

Approach:

The general approach to this problem will be to pull in data from an existing S3 bucket into an AWS sagemaker notebook. From here, the data can be visualized, cleaned and prepared to train an array of candidate models. The candidate models will be trained on the cloud in a sagemaker notebook, which will all be then scored using a variety of evaluation metrics to see what model performs best under different criteria.

Libraries Needed: NLTK, sklearn, numpy, pandas, matplotlib, seaborn

Data Collection/Exploration:

The data being used to train the sentiment analysis model is a public dataset stored on AWS S3:

https://s3.console.aws.amazon.com/s3/buckets/amazon-reviews-ml?region=us-west-2&tab=objects

Data Visualization:

For visualization of the data and the outcomes, we would be using *Matplotlib and seaborn Library* which is a python library specifically designed for making plots and graphs. We would be saving our plots and graphs in a PNG or other similar format and then would export the visualizations to either a S3 bucket or to the cloud based linux system.

Data Preprocessing:

We would be pre-processing the data to remove any unwanted and useless data. Additionally in this part, we would be feature engineering and transforming the raw data into a format that would be meaningful and useful to create models upon. To do so we would be using the nltk library, which is Natural Language ToolKit library which comes with multiple modules like stopwords detection, stemming the words, lemmatizing the words and so on. It also included modules to guess the sentiment of the text and assign scores. Other than that, we would be using Regular Expressions (re module in python) to do the low level text / language filtering. We might use additional modules if required.

Benchmark models

We'll look at all of **sklearn** out of the box models to build a baseline for the project. We looked into the following algorithms:

- **Gaussian Naive Bayes (GaussianNB)**

  For binary (two-class) and multi-class classification problems, Naive Bayes is a classification algorithm. When stated with binary or categorical input values, the technique is the easiest to grasp. Naive Bayes can be extended to real-valued attributes by assuming a Gaussian distribution, which is the most frequent assumption. Other functions can be used to estimate the data distribution, but the Gaussian (or Normal) distribution is the most straightforward to work with because it just requires you to estimate the mean and standard deviation from your training data.

- **Decision Trees**

  For classification, Decision Trees (DTs) are a non-parametric supervised learning method. The goal is to learn simple decision rules from data attributes to develop a model that predicts the value of a target variable. A tree is an approximation to a piecewise constant. In the example below, decision trees use a series of if-then-else decision rules to estimate a sine curve using data. The decision criteria become more complex as the tree grows deeper, and the model becomes more accurate.

- **Ensemble Methods (Bagging, AdaBoost, Random Forest, Gradient Boosting)**

  Bagging, often known as Bootstrap Aggregation, is a strong, simple, and effective ensemble approach. The approach employs the bootstrap to sample several copies of a training set, i.e. sampling with replacement, and it may be used with any form of classification or regression model.

  Random forest is a classification algorithm that uses numerous decision trees to classify data. When creating each individual tree, it employs bagging and feature randomization in order to generate an uncorrelated forest of trees whose committee prediction is more accurate than that of any one tree.

  Boosting is a meta-algorithm that can be thought of as a method of model averaging. It's the most popular ensemble method, as well as one of the most effective learning concepts. This method was created with classification in mind. The original boosting algorithm created a strong learner by combining three weak learners.

  AdaBoost algorithm employs very short (one-level) decision trees as weak learners, which are introduced to the ensemble in a sequential manner. Each model in the series after that seeks to correct the predictions made by the model preceding it. This is accomplished by balancing the training dataset to focus more on training cases where previous models failed to predict correctly.

- **K-Nearest Neighbours**

  The K-Nearest Neighbour method is based on the Supervised Learning approach and is one of the most basic Machine Learning algorithms. The method assumes that the new case/data and existing cases are comparable and places the new case in the category that is most similar to the existing categories. It saves all of the available data and classifies a new data point based on its resemblance to the existing data. Also known as a non parametric algorithm and a lazy learner.

- **Stochastic Gradient Descent**

  SGD (Stochastic Gradient Descent) is a straightforward yet effective optimization approach for determining the values of parameters/coefficients of functions that minimize a cost function. In other words, it's utilized to learn discriminative linear classifiers using convex loss functions like SVM and Logistic regression. (SGD) classifier is a simple SGD learning process that supports multiple loss functions and classification penalties.

- **Support Vector Machines**

  Support Vector Machines (SVMs) are a classification technique. It can handle both continuous and categorical data with ease. To differentiate various classes, SVM creates a hyperplane in multidimensional space. SVM iteratively creates the best hyperplane, which is then utilized to minimize an error. The goal of SVM is to identify a maximum marginal hyperplane (MMH) that splits the dataset into classes as evenly as possible.

- **Logistic Regression**

  Logistic regression is a supervised learning approach used to handle binary "classification" problems. Because logistic regression is a simple yet powerful classification technique, it is frequently employed for binary classification applications. Customer churn, spam email, website or ad click predictions are just a few of the problems that logistic regression can solve. It's even employed as a neural network layer activation function.

  The logistic function, commonly known as the sigmoid function, is the foundation of logistic regression. It takes any real-valued integer and translates it to a value between 0 and 1.

Evaluation:

We would be using the following evaluation metrics to understand the performance of our models.

Confusion matrix: It is a tabular summary of the number of correct and incorrect predictions made by a classifier. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score. Sklearn’s confusion_matrix module in this metrics sub-library is really useful to do so. In fact it also provides precision_recall_fscore_service, another module in the same sub-library that will help us get Precision, Recall and F-scores for our models.

Accuracy: Accuracy_measure module of the sklearn’s metrics sub-library will be used to evaluate the accuracy of the model. It takes the True labels and the Predicted Labels as arguments and calculates the accuracy of the model. Even though an F-score would be a good enough measure, accuracy is the simplest one to use and understand.

Cross validations: Cross-validation, sometimes called rotation estimation or out-of-sample testing, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set. This will help us ensure that the model that we are selecting is actually performing well in every case and just once by accident.

Member’s participation:

Aakef	Visualize data(unprocessed and processed) as well as evaluation metrics to draw insights and generate reports that would help us get meaningful insights.
Amandeep	Model development and training using the preprocessed data & hypertuning models aimed at data classification, in order to increase performance and produce better outcomes.

Amit	Data retrieval and uploading to the cloud. Data Preprocessing to remove anomalies and unwanted cases. Feature engineering to create bag-of-words, tf-idf, n-grams and additions features that would be used in training the model.
Rutul	Maintaining clusters on AWS, S3 buckets and similar services. Setting up IAM accounts for the team. Handling billing, cost and cutoff for the services.

Reference:

T. K. Shivaprasad and J. Shetty, "Sentiment analysis of product reviews: A review," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017, pp. 298-301, doi: 10.1109/ICICCT.2017.7975207.

Jagdale, Rajkumar S., Vishal S. Shirsat, and Sachin N. Deshmukh. "Sentiment analysis on product reviews using machine learning techniques." In Cognitive Informatics and Soft Computing, pp. 639-647. Springer, Singapore, 2019.

Mengle, Saket SR, and Maximo Gurmendez. Mastering machine learning on Aws: advanced machine learning in Python using SageMaker, Apache Spark, and TensorFlow. Packt Publishing Ltd, 2019.

https://scikit-learn.org/stable/supervised_learning.html