PROGRAM BOOK FOR

SHORT – TERM INTERNSHIP

(ONLINE)

Name of the Student : Vasipalli Abhisri

Name of the Course : Artificial Intelligence and Machine Learning

Name of the College : Vishnu Institute of Technology

Registration Number : 21PA1A05I7

Period Of Internship : 10 weeks

Name and Address of

Intern Organization : Edu Skills Foundation under AICTE and Amazon Web Services (AWS).

Vishnu Institute of Technology

Bhimavaram, WGDT, Andhra Pradesh

VISHNU
UNIVERSAL LEARNING

May-July 2023

An Internship Report on

Artificial Intelligence and Machine Learning

Submitted in accordance with the requirement for the degree of

Bachelor of Technology

Under the Faculty Guideship of

A Revathi

Department of

Computer Science and Engineering

Vishnu Institute of Technology

Submitted by:

Vasipalli Abhisri

Reg. No: 21PA1A05I7

Department of

Computer Science and Engineering

Vishnu Institute of Technology

DECLARATION

I, vasipalli Abhisri a student of 3rd Year of Bachelor of Technology Program, Reg. No. 21PA1A05I7 of the Department of Computer Science and Engineering, Vishnu Institute of Technology College do hereby declare that I have completed the mandatory internship from May to July 2023 in AI&ML from EduSkills Foundation under AICTE and Amazon Web Services(AWS) under the Faculty Guideship of Mr. D. Shankar, Department of Computer Science and Engineering, Vishnu Institute of Technology.

Vasipalli Abhisri

Date:

Certificate

This is to certify that the **Summer Intern Project** report submitted by **Vasipalli Abhisri** on the title "**Artificial Intelligence and Machine Learning Virtual Internship**" is a record of the summer intern project work done by him during the academic year 2023-2024 in partial fulfillment of Bachelor of Technology.

Internal Examiner

Head of the Department

External Examiner

Contents

Chapter 1. Introduction to Machine Learning

This chapter serves as an introduction to machine learning (ML) and illustrates how ML fits into the broader context of artificial intelligence (AI).

1.1 What is Machine Learning?

Machine learning is a subset of AI, which is a broad branch of computer science for building machines that can do human tasks. Deep learning itself a subdomain of machine learning.
Machine learning is the scientific study of algorithms and statistical models to perform a task by using inference instead of instructions.

Tom Mitchell, a pioneer of machine learning, wrote this definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Deep Learning

The capabilities of AI and ML have advanced significantly with the development of deep learning. The operation of the human brain served as the basis for the theory underlying deep learning. Although the technology is different, an artificial neural network (ANN) is modelled after the organic neurons in the brain. Artificial neurons have a single output and one or more inputs. Based on a transformation of the inputs, these neurons fire (or activate their outputs). These synthetic neurons are arranged in layers with connections between the layers to form brain networks. A network typically consists of hidden, input, and output layers.

1.2 Business problems solved using machine learning

Machine learning has three main types.
The first type is Supervised Learning, where a model uses known inputs and outputs to generalize future outputs.
The second type is Unsupervised learning, where the model does not know inputs or outputs - it finds patterns in the data without help.
The third type is Reinforcement learning, where the model interacts with its environment and learns to take actions that maximize rewards.
It is important to know the different ML types because the type can guide you toward selecting algorithms that make sense for solving your business problem.

Supervised Learning:

Supervised learning is further classified into Classification and Regression problems.

One of the main use cases of supervised learning:
Computer vision (CV) is a large field that consists mostly of classification problems.
Computer vision enables machines to identify people, places, and things in images with accuracy at (or above) human levels, and with greater speed and efficiency. It is often built with deep learning models.

Unsupervised Learning:

The common sub-categories of Unsupervised Learning are Clustering and Dimensionality reduction.

One of the main use cases of unsupervised learning:

Natural language processing (NLP) is another area of machine learning with increasing use.
NLP is used in many applications such as:

- Chat or call centre bots – Automated systems for getting your bank balance or ordering food from a restaurant.
- Translation tools – Converting text between languages, or applications that can translate menus in real time.
- Voice-to-text translations – Converting spoken words into text. Can be used to power automatic subtitles.
- Sentiment analysis – Enables you to analyse the sentiment of comments in reviews of products, music, and movies. These sentiments can be used to give the movie an audience rating.

Reinforcement Learning:

Another kind of machine learning that has been gaining in popularity recently is reinforcement learning. Unlike other machine learning, reinforcement learning continuously improves its model by mining feedback from previous iterations. In reinforcement learning, an agent continuously learns, through trial and error, as it interacts in an environment.

One of the main use cases of reinforcement learning:
Self-driving vehicles bring together several machine-and deep-learning algorithms and models to solve the problem of driving from A to B. A major task is continuously detecting the environment and forecasting changes. This task involves object detection, which is the localization and prediction of movement of the detected object. The outputs of these findings act as inputs to other systems that make decisions on what to do with the vehicle's various controls.

Some use cases involve self-driving vehicles that require real-time responses to the environment. For example, if a previously hidden pedestrian walks out from behind an obstacle, the vehicle brakes must be applied immediately. Such actions cannot have any latency or room for error.

1.3 Machine Learning Process

The machine learning pipeline process can guide through the process of training and evaluating a model.

The iterative process can be broken into three broad steps –

- Data processing
- Model training
- Model evaluation

When you train your model, recognize the danger of overfitting or underfitting the model.

Overfitting – Model performs well on training data but it does not perform well on evaluation data.
Underfitting – Model performs poorly on the training data.

After model is retained and satisfied with the results, the model is deployed to deliver the best possible predictions.

1.4 Overview of Machine Learning Tools

The various tools used for machine learning are

Jupyter Notebook is an open-source web application that enables you to create and share documents that contain live code, equations, visualizations, and narrative text.

pandas is an open-source Python library. It is used for data handling and analysis. It represents data in a table that is similar to a spreadsheet. This table is known as a pandas DataFrame

Matplotlib is a library for creating scientific static, animated, and interactive visualizations in Python. You use it to generate plots of your data later in this course. Seaborn is another data visualization library for Python. It's built on matplotlib, and it provides a high-level interface for drawing informative statistical graphics.

NumPy is one of the fundamental scientific computing packages in Python. It contains functions for Ndimensional array objects and useful math functions such as linear algebra, Fourier transform, and random number capabilities.

scikit-learn is an open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data pre-processing, model selection and evaluation, and many other utilities. scikit-learn is built on NumPy, SciPy, and matplotlib, and it is a good package for exploring machine learning. Although you use it only to borrow a few functions in a later module, you can explore this package in greater detail after you complete this course.

1.5 Machine Learning Challenges

One can expect to encounter many challenges in machine learning like

Data:

- There is a large amount of poor-quality and inconsistent data in the world. Much of your job involves getting good data.
- Does the data represent the problem well? If you try to find credit card fraud, do you have examples to train with?

Users:

- Do you have data-science experience?
- Is staffing a team of data scientists cost-effective?

Business:

- Are the problems too complex to formulate into an ML problem?
- Can the resulting model be explained to the business? If it cannot, it might not get adopted.

Chapter-2 Implementing a Machine Learning Pipeline with Amazon SageMaker

This part introduces and describes a typical process for handling a machine learning problem. A machine learning pipeline can be applied to many machine learning problems which focuses on supervised learning, but can be adapted to other types of machine learning.

2.1 Formulating Machine Learning problem

Consider the following example.
You want to identify fraudulent credit card transactions so that you can stop the transaction before it processes. That is the problem. What is the business goal or outcome that drives this problem statement? The intended outcome is a reduction in the number of customers who end their membership to the credit card because of a fraudulent transaction.
From a business perspective, how do you define success when you have this problem and intended outcome? At this stage, you must move from qualitative statements to quantitative statements that can be easily measured. Continuing with this example, you might define success for this problem with the following metric: a successful outcome is a 10 percent reduction in the number of customers who file claims for fraudulent transactions within a 6-month period.
You have now defined the business end of your problem. Next, you start to think about the problem in terms of your machine learning (ML) model. What output do you want to see from your model? Be specific - it should be a statement that reflects what an ML model can output. An example might be: The model will output whether a credit card transaction is fraudulent or not fraudulent. Therefore, in this case, you can see that you are dealing with a binary classification problem.

In short,

- Business problems must be converted into an ML problem. Questions to ask include o Have we asked why enough times to get a solid business problem statement and know why it is important?
  - o Can you measure the outcome or impact if your solution is implemented?
- Most business problems fall into one of two categories o Classification (binary or multi): Does the target belong to a class?
  - o Regression: Can you predict a numerical value?

2.2 Collecting and Securing data

Collecting data:

Data can be obtained from several places.

- Private data is data that you (or your customers) have in various existing systems. Everything from log files to customer invoice databases can be useful, depending on the problem that you want to solve.
- Commercial data is data that a commercial entity collected and made available. Companies maintain databases that you can subscribe to. These databases include curated news stories, anonymized healthcare transactions, global business records, and location data. Supplementing your own data with commercial data can provide useful insights that you would not have otherwise.
- Open-source data comprises many different open-source datasets that range from scientific information to movie reviews. These datasets are usually available for use in research or for teaching purposes. One

can find open-source datasets hosted by AWS, Kaggle, and the UC Irvine Machine Learning Repository. Government and health organizations are other sources of data that might be useful.

Securing Data:

It is important to consider the security of your data. The real data about customer transactions or health records must be kept secure. The AWS Identity and Access Management (IAM) service controls access to resources. One must make sure that correctly secure your data within AWS to avoid data breaches.

In addition to controlling access to data, you must make sure that your data is secure. It is a good practice, and it might also be legally required for certain data types, such as financial or healthcare records.

2.3 Evaluating the Data

Analysing information involves examining it in ways that reveal the relationships, patterns, trends, etc. that can be found within it. That may mean subjecting it to statistical operations that can tell you not only what kinds of relationships seem to exist among variables, but also to what level you can trust the answers you are getting. There are two kinds of data you are apt to be working with, although not all evaluations will necessarily include both.

- Quantitative Data - refer to the information that is collected as, or can be translated into, numbers, which can then be displayed and analysed mathematically.
- Qualitative Data - are collected as descriptions, anecdotes, opinions, quotes, interpretations, etc., and are generally either not able to be reduced to numbers, or are considered more valuable or informative if left as narratives.

Need for Data Evaluation

- The data can show whether there was any significant change in the dependent variable(s) you hoped to influence.
- They can uncover factors that may be associated with changes in the dependent variable(s).
- They can show connections between or among various factors that may have an effect on the results of your evaluation.
- They can help shed light on the reasons that your work was effective or, perhaps, less effective than you had hoped.
- They can provide you with credible evidence to show stakeholders that your program is successful, or that you have uncovered, and are addressing limitations.
- Their use shows that you are serious about evaluation and about improving your work.
- They can show the field what you are learning, and thus pave the way for others to implement successful methods and approaches.

2.4 Feature Engineering

Two things can make the models more successful: feature selection and feature extraction or creation.

Feature selection is about selecting the features that are most relevant and discarding the rest. Feature selection is applied to prevent either redundancy or irrelevance in the existing features, or to get a limited number of features to prevent overfitting.

Feature extraction is about building up valuable information from raw data by reformatting, combining, and transforming primary features into new ones. This transformation continues until it yields a new set of data that can be consumed by the model to achieve the goals.

Feature extraction covers many activities that range from handling missing data to converting text data into numerical data. Although the list is not exhaustive, it should give you some idea of the data handling that is needed to get data into a useful state.

Encoding Ordinal Data

Most ML algorithms work best with numerical data. Therefore, one must make sure that all columns in the dataset contain numeric data by converting or encoding the data.

If the categorical data has order to it, you can encode the text into numerical values that capture this ordinal relationship. For data that shows maintenance costs, you might encode Low to 1, Medium to 2, High to 3, and Very High to 4.

Encoding non-ordinal data

If the categorical data does not have any order to it, then you must break the data into multiple columns. You do not want to introduce an ordinal relationship to the data that is not present. For example, suppose you assign a value of 1 to the first colour (like red) and 2 to the next value (like blue). Then, the model might interpret blue as more important than red, because it has a higher numeric value. A better way is to encode non-ordinal data into multiple columns or features.

Cleaning the data

In addition to converting string data to numerical data, you must clean your dataset for several other potential problem areas. Before you encode the string data, you must make sure that the strings are all consistent. You also must make sure that variables use a consistent scale.

Some data items might also capture more than one variable in a single value. If you want to train your ML system for both variables, you must split that variable into two variables.

Finding missing data

You might also find that you have missing data. For example, some columns in your dataset might be missing data because of a data collection error. Perhaps data was not collected on a particular feature until the data collection process was well under way. Missing data can make it difficult to accurately interpret the relationship between the related feature and the target variable.

In most cases one must use human intelligence to update missing values with something meaningful and relevant to the problem.

Finding and dealing with outliers

Outliers are points in your dataset that lie at an abnormal distance from other values. They are not always something that you want to clean up, because they can add richness to your dataset. However, they can also make it harder to make accurate predictions. The outliers affect accuracy because they skew values away from the other more normal values that are related to that feature. In addition, an outlier might indicate that the data point belongs to another column.

One of the more common ways to find univariate outliers is with a box plot. A box plot shows how far a data point is from the mean for that variable.

A scatter plot can be an effective way to see multivariate outliers.

You can handle outliers with several different approaches. They include, but are not limited to:

- Deleting the outlier: This approach might be a good choice if your outlier is based on an artificial error. Artificial error means that the outlier is not natural and was introduced because of some failure perhaps incorrectly entered data.
- Transforming the outlier: You can transform the outlier by taking the natural log of a value, which in turn reduces the variation that the extreme outlier value causes. Therefore, it reduces the outlier's influence on the overall dataset.
- Imputing a new value for the outlier: You can use the mean of the feature, for instance, and impute that value to replace the outlier value. Again, this would be a good approach if an artificial error caused the outlier.

2.5 Training a model

Some algorithms might not be able to work with training data in a DataFrame format. In general, the data that you use for ML training can be stored in various formats, depending on your use case and algorithm. Various algorithms commonly use some file formats, such as CSV. Many Amazon SageMaker algorithms support training with data in CSV format.
The target variable in your training dataset should be the first column on the left. Your features should be to the right of the target variable column.

Splitting the data

Evaluating a model with the same data that it trained on leads to overfitting. Recall that overfitting is where your model learns the particulars of a dataset too well. It essentially memorizes the training data, instead of learning the relationships between features and labels. Hence the following methods can be used for training the model,

Hold-out method is when you split your data into multiple sets. These sets are usually training data, validation data, and testing data.

k-fold cross validation technique provides good metrics to choose which model is better. K-fold cross validation randomly partitions the data into K different segments. For each segment, you use the rest of the data outside the segment for training to do a validation on that particular segment.

2.6 Hosting and using the model

After the model is trained, it is ready to be deployed. One can deploy your model in two ways.

For single predictions, deploy your model with Amazon SageMaker hosting services. Amazon SageMaker deploys multiple compute instances that run your model behind a load balanced endpoint. Applications can call the API at the endpoint to make predictions.

To get predictions for an entire dataset, use Amazon SageMaker batch transform. Instead of deploy and maintain a permanent endpoint, Amazon SageMaker spins up your model and performs the predictions for the entire dataset that you provide. It stores the results in Amazon S3 before it shuts down and terminates the commute instances. Performing batch predictions when you test the model is useful because you can quickly run your entire validation set against the model. You do not need to write any code to process and collate the individual results.

2.7 Evaluating the accuracy of the model

An important part of this phase is to choose the most appropriate metric for the business situation.
Evaluating the model helps us to understand how well the model is performing.
The various evaluation metrics that can be used for evaluating the model are:

For Classification Model,

Confusion Matrix : In a confusion matrix, you can get a high-level comparison of how the predicted classes matched up against the actual classes.
The three main metrics used to evaluate a classification model are accuracy, precision, and recall.
Accuracy is defined as the percentage of correct Prediction for the test data.

$$\text{Accuracy} = \frac{Number\ of\ correct\ Predictiions}{Total\ number\ of\ predictions}$$

Precision is defined as the fraction of relevant examples among all of the examples which were predicted to belong in certain class.

$$\text{Prediction} = \frac{TP}{TP+FP}$$

Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all the examples that are truly belong in that class.

$$\text{Recall}= \frac{TP}{TP+FN}$$

For Regression Model,

Evaluation metrics for regression models are quite different than metrics for classification models because we are now predicting in a continuous range instead of a discrete number of classes.

- Mean Absolute Error (MAE) : MAE is fundamental and most used evaluation metric for regression problems. Here the difference between the actual and predicted values is calculated. This error can either be positive or negative but we are concerned about the magnitude and is given by

$$\text{MAE} =\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(x_i = x)$$

- Mean Squared Error (MSE) : MSE is a prevalent evaluation metric for regression problems. It is similar to the mean absolute Error, but the Error is squared here
- $$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i = \overline{y_i})^2$$

- Root Mean Squared Error (RMSE) : RMSE is the most famous evaluation metric for the regression model. The overall calculation of RMSE is similar to MSE; the final value is square-rooted as we calculated the square of errors in MSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - yj)^2}$$

R – Squared (□) : R-squared explains to what extent one variable's variance explains the second variable's variance. It is also known as the Coefficient of Determination.

$$R = \frac{\sum_{i=1}^{N}(y_i - y_\iota)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}.$$

2.8 Hyperparameter and Model Tuning

Hyperparameters can be thought of as the knobs that tune the machine learning algorithm to improve its performance.

Hyperparameters have a few different categories.

□ The first kind is model hyperparameters, which help define the model itself.

□ The second kind is optimizer hyperparameters. These hyperparameters are related to how the model learns the patterns that are based on data, and they are used for a neural-network model. These types of hyperparameters include optimizers like gradient descent and stochastic gradient descent.

□ The third kind is data hyperparameters, which relate to the attributes of the data itself. They include attributes that define different data augmentation techniques, such as cropping or resizing for imagerelated problems.

Hyperparameter tuning might not necessarily improve your model. It’s an advanced tool for building machine solutions. As such, it should be considered part of the process of using the scientific method. When you build complex ML systems like deep learning neural networks, it is impractical to explore all the possible combinations.

Tuning best practices:

1. Don’t adjust every parameter.
2. Limit your range of values to what’s most effective.
3. Run one training job at a time instead of multiple jobs in parallel.
4. In distributed training jobs, make sure that the objective metric that you want is the one that is reported back.

Chapter-3 Forecasting

3.1 Overview of Forecasting

Forecasting is an important area of machine learning. It is important because so many opportunities for predicting future outcomes are based on historical data. Many of these opportunities involve a time component. Although the time component adds more information, it also makes time series problems more difficult to handle than other types of predictions.

The time series data falls into two broad categories.

- The first type is univariate, which means that it has only one variable.
- The second type is multivariate, which means that it has more than one variable.

In addition to these two categories, most time series datasets also follow one of the following patterns:

- Trend – A pattern that shows the values as they increase, decrease, or stay the same over time.
- Seasonal – A repeating pattern that is based on the seasons in a year.
- Cyclical – Some other form of a repeating pattern.
- Irregular – Changes in the data over time that appear to be random or that have no discernible pattern.

3.2 Processing Time Series data

Time series data is captured in chronological sequence over a defined period of time.
Introducing time into a machine learning model has a positive impact because the model can derive meaning from change in the data points over time. Time series data tends to be correlated, which means that a dependency exists between data points.

Because you have a regression problem - and because regression assumes independence of data points - you must develop a method for handling data dependence. The purpose of this method is to increase the validity of the predictions.

Handling missing data
A common occurrence in real-world forecasting problems is missing values in the raw data. Missing values makes it harder for a model to generate a forecast. The primary example in retail is an out-of-stock situation in demand forecasting. If an item goes out of stock, the sales for the day will zero. If the forecast is generated based on those zero sales values, the forecast will be incorrect.

Missing values can be marked as missing for various reasons. Missing values can occur because of no transaction, or possibly because of measurement errors. Maybe a service that monitored certain data was not working correctly, or the measurement could not occur correctly.
The missing data can be calculated in several ways:

- Forward fill - Uses the last known value for the missing value.
- Moving average - Uses the average of the last known values to calculate the missing value.

- Backward fill - Uses the next known value after the missing value. Be aware that it is a potential danger to use the future to calculate the past, which is bad in forecasting. This practice is known as lookahead, and it should be avoided.
  Interpolation - Essentially uses an equation to calculate the missing value.

Downsampling
Downsample means moving from a more finely grained time to a less finely grained time.
When you downsample, you must decide how to combine the values. In the case of sales data, summing the quantity makes the most sense. If the data is temperature, you might want to find the average. Understanding your data helps you decide what the best course of action is.

Upsampling
The inverse of downsampling is upsampling. The problem with upsampling is that it's difficult to achieve in most cases. Suppose that you wanted
to upsample your sales data from daily sales to hourly sales. Unless you have some other data source
to reference, you wouldn't be able to change from daily to hourly sales.

Smoothing
Removing these outliers and anomalies is known as smoothing.
You might consider smoothing for the following reasons.

1. Data preparation - Removing error values and outliers
2. Visualization - Reducing noise in a plot

Seasonality
Seasonality in data is any kind of repeating observation where the frequency of the observation is stable.

Stationarity, trends and autocorrelation
It is important to know how stable a system is. The level of stability, or stationarity, can tell you how much you should expect the system's past behaviour to inform future behaviour. A system with low stability is not good for predicting the future.

Often, you will want to determine the trend for a time series. However, adjusting the series for the trend can make it difficult to compare the series with another series that was also adjusted for trend. The trends might dominate the values in the series, which can lead you to overestimate of the correlation between the two series.

Autocorrelation is one of the special problems that you face with time series data. As you saw in other machine learning problems, the goal of building an ML model is to separate the signal from the noise. Autocorrelation is a form of noise because separate observations are not independent of each other.

A time series with autocorrelation might overstate the accuracy of the model that is produced. Some of the algorithms that you see in this module can help correct for autocorrelation.

Chapter-4 Computer Vision

Computer vision is an exciting space in machine learning. The advances in computing power and algorithms over the last 10 years have led to an increase in capabilities and easier access to computer vision technologies.

4.1 Introduction to Computer Vision (CV)

Computer vision enables machines to identify people, places, and things in images with accuracy at or above human levels, with greater speed and efficiency. Often built with deep learning models, computer vision automates the extraction, analysis, classification, and understanding of useful information from a single image or a sequence of images. The image data can take many forms, such as single images, video sequences, views from multiple cameras, or three - dimensional data.

Classification in machine learning is used to decide which category or categories that a picture or object belongs to. This process is no different than any other classification problem for machine learning.

When you have multiple classes, this is known as a multi-class classification problem. When you have only two classes, this is known as a binary classification problem.

Object detection provides the categories of the image and the location of the objects in the image. The location is provided by a set of coordinates for a box that surrounds the image, which is known as the bounding box. Bounding boxes for object detection typically provide top, left, width, and height coordinates that surround the images. You can use these coordinates in your applications. When objects are detected in an image, a confidence number is usually associated with that object. This percentage indicates how probable it is that the object belongs to a specific class. This confidence level is important when you want to determine an action that is based on object detection, especially in applications that use facial detection.

Object segmentation is similar to object detection, but you go into more detail to get fine boundaries for each detected object. It is a fine-grained inference for predicting each pixel in the image. Applications that require object segmentation include autonomous vehicles and advanced computer-human interactions.

4.2 Image and Video Analysis

Amazon Rekognition is a computer vision service based on deep learning. You can use it to add image and video analysis to your applications.

Amazon Rekognition enables you to perform the following types of analysis:

- Searchable image and video libraries - Amazon Rekognition makes images and stored videos searchable so that you can discover the objects and scenes that appear in them.
- Face-based user verification - Amazon Rekognition enables your applications to confirm user identities by comparing their live image with a reference image.
- Sentiment and demographic analysis - Amazon Rekognition interprets emotional expressions, such as happy, sad, or surprise. It can also interpret demographic information from facial images, such as gender.

- Unsafe content detection - Amazon Rekognition can detect inappropriate content in images and in stored videos
  Text detection - Amazon Rekognition Text in Image enables you to recognize and extract text content from images.

You need to check if the applications you build using Amazon Rekognition would fall under any regulatory restrictions as defined within your field or country. Security and compliance for Amazon Rekognition is a shared responsibility between AWS and the customer.

Amazon Rekognition is designed to integrate into your applications via the API and SDKs. API operations are provided for detecting labels, faces, recognizing celebrities, and detecting unsafe images. To perform a prediction, you must provide the service either an image object in Amazon S3, or upload a byte steam of an image. Images can be JPEG or PNG formats.
Amazon Rekognition processes the image, performs the prediction, and returns a JSON object with the results.

Image Analysis
When Amazon Rekognition detects a human face, it captures a bounding box that indicates where the face was found in the video. It also can detect attributes such as position of the eyes, nose, and mouth. It can detect emotion, quality of the detection, and any landmarks that might appear.
All these items will have an associated confidence score. A higher score indicates that the model has greater confidence about the detection.
Gender is inferred from the image. It is not inferred from identity. Similarly, emotion is also determined based on the image and it might not reflect the subject’s actual emotional state.

Video Analysis
You can use Amazon Rekognition Video to detect and recognize faces in streaming video. A typical use case is when you want to detect a known face in a video stream. Amazon Rekognition Video uses Amazon Kinesis Video Streams to receive and process a video stream. The analysis results are output from Amazon Rekognition Video to a Kinesis data stream and are then read by your client application. Amazon Rekognition Video provides a stream processor (CreateStreamProcessor) that you can use to start and manage the analysis of streaming video.
To use Amazon Rekognition Video with streaming video, your application must implement the following resources:

- A Kinesis video stream for sending streaming video to Amazon Rekognition Video.
- An Amazon Rekognition Video stream processor to manage the analysis of the streaming video.
- A Kinesis data stream consumer to read the analysis results that Amazon Rekognition Video sends to the Kinesis data stream.

To find a face, you must create a collection. This process is the same as creating a collection when you work with images.

Amazon Rekognition Video places a JSON frame record for each analysed frame into the Kinesis output stream. Amazon Rekognition Video doesn't analyse every frame that's passed to it through the Kinesis video stream. A frame record that's sent to a Kinesis data stream contains information about which Kinesis video stream fragment the frame is in, where the frame is in the fragment, and faces that are recognized in the frame. It also includes status information for the stream processor.

4.3 Preparing custom datasets for Computer Vision

One challenge of using a pre-built model is that it will only find images that it was trained to find. Though Amazon Rekognition was trained with tens of millions of images, it can’t detect objects that it wasn’t trained on.
As with other machine learning processes, you must train Amazon Rekognition to recognize scenes and objects that are in a domain. Thus, you need a training dataset and a test dataset that contains labelled images. Amazon Rekognition Custom Labels can be helpful for these tasks. You can use Amazon Rekognition Custom Labels to find objects and scenes that are unique to your business needs. For example, you can use it to classify images (image-level predictions) or detect images (object-level or bounding-box-level predictions

Many machine learning problems today can be solved by training existing models. Training a computer vision algorithm to recognize images requires a large input dataset, which is impractical for most organizations. You can use an existing model or a managed service like Amazon Rekognition Custom Labels to:

- Simplify data labelling - Amazon Rekognition Custom Labels provides a UI for labelling images, including defining bounding boxes.
- Provide automated machine learning - Amazon Rekognition Custom Labels includes automated machine learning capabilities that handle the ML process for you. When you provide training images, Amazon Rekognition Custom Labels can automatically load and inspect the data, select the correct machine learning algorithms, train a model, and provide model performance metrics.
- Provide simplified model evaluation, inference, and feedback - You evaluate your custom model’s performance on your test set. For every image in the test set, you can see the side-by-side comparison of the model’s prediction versus the label that it assigned. You can also review detailed performance metrics. You can start using your model immediately for image analysis, or you can iterate and retrain new versions with more images to improve performance.

After you start using your model, you can track your predictions, correct any mistakes, and use the feedback data to retrain new model versions and improve performance.

The following are the steps to be followed for Custom Labelling:
Step 1 - Collect images that contain the objects or scenes you want to find.
Step 2 - Upload and label images from your computer or Amazon S3, or import an Amazon SageMaker Ground Truth .manifest file for already labelled images.
Step 3 - Create a dataset to evaluate your model’s performance, select an existing dataset, or split your training dataset for testing.
Step 4 - Train your custom model by using your training datasets. The best ML techniques will automatically be selected.
Step 5 - Evaluate your model performance on your test dataset. Improve your model by adding images to the training dataset.
Step 6 - Use your customer model to analyse images with an API operation.

Chapter-5 Natural Language Processing

5.1 Overview of Natural Language Processing (NLP)

NLP is a broad term for a general set of business or computational problems that you can solve with machine learning (ML). NLP systems predate ML. Two examples are speech-to-text on your old cell phone and screen readers. Many NLP systems now use some form of machine learning. NLP considers the hierarchical structure of language. Words are at the lowest layer of the hierarchy. A group of words make a phrase. The next level up consists of phrases, which make a sentence, and ultimately, sentences convey ideas.

Challenges for NLP:

- Discovering the structure of the text - One of the first tasks of any NLP application is to break the text into meaningful units, such as words, phrases, and sentences.
- Labelling data - After the system converts the text to data, the next challenge is to apply labels that represent the various parts of speech. Every language requires a different labelling scheme to match the language's grammar.
- Representing context - Because word meaning depends on context, any NLP system needs a way to represent context. It is a big challenge because of the large number of contexts. Converting context into a form that computers can understand is difficult.
- Applying grammar - Although grammar defines a structure for language, the application of grammar is nearly infinite. Dealing with the variation in how humans use language is a major challenge for NLP systems. Addressing this challenge is where machine learning can have a big impact.

You can apply the ML development pipeline that you have seen throughout this course when you develop an NLP solution. The first task is to formulate a problem, and then collect and label data.

For NLP, collecting data consists of breaking the text into meaningful subsets and labelling the sets. Feature engineering is a large part of NLP applications. This process gets more complicated when you have irregular or unstructured text. For example, if you build an application to classify documents, you must be able to distinguish between words with common terms but different meanings. Labelling data in the NLP domain is sometimes also called tagging. In the labelling process, you must assign individual text strings to different part of speech. You can use specialized tools to help with NLP labelling.

5.2 Pre-processing

The first task for an NLP application is to convert the text to data so that it can be analysed. You convert text by removing words that are not needed for the analysis from the input text.
In the example "This is Sample text", the words This and is are removed to leave the phrase sample text.

Lemmatization

After you remove these stop words, you can normalize text by converting similar words into a common form. For example, the words run, runner, ran, and running are all different forms of the word run. You can normalize all instances of these words within a block of text by using processes of stemming and lemmatization.

Standardization :

After you normalize the text, you can standardize it by removing words that are not in the dictionary that you use for analysis. Examples include acronyms, slang, and special characters.

The Natural Language Toolkit (NLTK) Python library provides functions that you can use to remove stop words, normalize, and standardize text.

5.3 Creating tokens and feature engineering

One of the first steps for creating an NLP system is to convert the text into a data collection, such as a DataFrame. All the NLP libraries provide functions to assist with this type of conversion. This example shows how to use the word_tokenize function that's provided in the NLTK library.

After you clean up your text and load it into a DataFrame, you can apply one of the NLP models to create features.

Common models include:

- Bag of words - This simple model captures the frequency of words in a document. For each word in the document, a key is created, with a value that is the number of occurrences within that document.
- Term frequency and inverse document frequency (TF-IDF) - Term frequency is a count of how many times a word appears in a document. Inverse document frequency is the number of times a word occurs in a group of documents. These two values are used

together to calculate a weight for the words. Words that frequently appear in many documents have a lower weight.

Text Analysis Categories

Text analysis has three broad categories:

- Classifying text - This category of analysis is similar to other classification systems that you have seen in this course. Text provides the input to a process that extracts features, which are then sent though an ML algorithm. This algorithm interacts with a classifier model to infer the classification. You can use the NLTK Python library to create a classification system.
- Discovering similarities - Text matching has many applications. For example, auto-correct, spell check, and grammar check are all based on text matching. The edit distance (also known as the Lowenstein distance) algorithm is frequently used.
- Deriving relationships - You can derive relationships between different words or phrases in the text by using a process called coreference resolution. Several NLP systems provide Python libraries for deriving relationships.

Derive meaning by entity extraction

The process of extracting entities is known as named entity recognition (NER). A NER model has the following components:

- Identify noun phrases by using dependency charts and part of speech tagging.
- Classify phrases by using a classification algorithm, such as Word2Vec.
- Disambiguate entities by using a knowledge graph.

After the named entities are extracted, you can use a knowledge graph to extract meaning. A knowledge graph combines subject matter expertise with machine learning to derive meaning.

ACTIVITY LOG

Day	Brief description of daily activity	Learning Outcome	Person in-charge Signature
Day – 1,2,3	Introduction to Machine Learning	Recognize how machine learning and deep learning are part of artificial intelligence	
Day – 4,5,6	Business problems solved using Machine Learning	Able to identify how machine learning can be used to solve a business problem	
Day – 7,8,9	Machine Learning terminology and process	Able to describe artificial intelligence and machine learning terminology	
Day – 10,11,12	Overview of Machine Learning tools	Able to list the tools available to data scientists	
Day – 13,14	Challenges in Machine Learning	Able to identify when to use Machine Learning instead of traditional software development methods	

WEEKLY REPORT

Week 1 & 2

Objective of Activity done:

Is to recognize how machine learning and deep learning are part of Artificial intelligence, to describe artificial and machine learning terminology, identify how machine learning can be used to solve a business problem, describe the machine learning process, list the tools available to data scientists, identify when to use machine learning instead of traditional software development methods

Detailed Report:

- A subset of the larger discipline of computer science known as artificial intelligence is machine learning.
- Describe various terms used in the machine learning and artificial intelligence.
- Formulating a business problem into Supervised, Unsupervised or Reinforcement problem.
- Understand the classification of machine learning problems into Classification and Regression problems.
- Understand how machine learning can be applied for classifying objects in images and videos through Computer Vision
- Have a glance at various tools used to perform machine learning like Jupyter notebook and pandas, NumPy, sklearn, matplotlib... libraries of python.

ACTIVITY LOG

Day	Brief description of daily activity	Learning Outcome	Person in-charge Signature
Day – 1,2,3	Formulation of problem	Able to formulate a business problem	
Day – 4,5,6	Obtain and secure data	Able to store data in Amazon S3 or in Amazon Elastic File System	
Day – 7,8,9	Evaluating the obtained data	Able to build a Jupyter notebook and outline the process of data evaluation	
Day – 10,11,12	Training the model	Able to train the model using various machine learning tools	
Day – 13,14	Tuning	Tune the model with the optimal parameters	

WEEKLY REPORT

Week 3 & 4

Objective of Activity done:

Is to collect and secure data, evaluate the collected data and pre-process it, train the machine learning model with the obtained data, evaluate the accuracy of the model, host the model for deployment and tune the model with optimal parameters for better predictions.

Detailed Report:

- Turn a business requirement into a machine learning problem.
- Real world data is not something that is to be kept public. Hence it is securely stored in Amazon S3 or in

  Amazon Elastic File System with IAM policy.
- Evaluating data helps in understanding about the collected data and makes data pre-processing easy.
- A correlation matrix conveys both the strong and weak linear relationships among numerical variables.
- Feature engineering helps to recognize the main features that have an impact on the predictions. It is done by feature selection and feature extraction.
- Since most machine learning models cannot deal with missing values, the columns are either imputed with some meaningful value like the mean of highly corelated feature or the column is dropped.
- Outliers make it hard to make accurate predictions. The outliers affect accuracy because they skew values away from the other more normal values that are related to that feature.

ACTIVITY LOG

Day	Brief description of daily activity	Learning Outcome	Person in-charge Signature
Day – 1,2,3	Overview of forecasting	Describe the business problems solved by using Amazon Forecast	
Day – 4,5,6	Processing time series data	Able to describe the challenges of working with time series data	
Day – 7,8,9	Using Amazon Forecast	Able to list the steps that are required to create a forecast by using Amazon Forecast	
Day – 10,11,12	Training the model	Able to split the data for training the model	
Day – 13,14	Evaluating the model	Able to evaluate the model using various metrics	

WEEKLY REPORT

Week 5 & 6

Objective of Activity done:

Is to able to describe the business problems solved by using Amazon Forest, describe the challenges of working with time series data, list the steps that are required to create a forecast by using Amazon Forecast, use Amazon Forecast to make a prediction.

Detailed Report:

- Machine learning (ML) in demand forecasting makes it possible to avoid traditional challenges associated with planning such as long delivery lead times, high transport costs, high inventory and waste levels, and incorrect decision making due to inaccurate forecasts.
- ML forecasting algorithms often use techniques that involve more complex features and predictive methods, but the objective of ML forecasting methods is the same as that of traditional methods – to improve the accuracy of forecasts while minimizing a loss function.
- One of the reasons was that most of the use cases involved forecasting low-frequency series with monthly, quarterly, or yearly granularity.
- To obtain data at different frequencies either downsampling or upsampling is performed.
- Seasonality in data is any kind of repeating observation where the frequency of the observation is stable.

ACTIVITY LOG

Day	Brief description of daily activity	Learning Outcome	Person in-charge Signature
Day – 1 - 6	Overview of Computer Vision	Understand the basics of computer vision	
Day – 7 - 10	Analyse images and videos	Able to understand the processing of finding objects in a frame	
Day – 11,12	Preparing Custom datasets for computer vision	Able to list the steps required to preparer custom dataset	
Day – 13,14	Training the model	Able to use Amazon Rekogniton to perform facial detection	

WEEKLY REPORT

Week 7 & 8

Objective of Activity done:

Is to describe the use cases for computer vision, describe the Amazon managed machine learning services available for image and video analysis, list the steps required to prepare a custom dataset for object detection, describe how Amazon SageMaker Ground Truth can be used to prepare a custom dataset, use Amazon Rekognition to perform facial detection.

Detailed Report:

- Computer vision is the automated extraction of information from digital images.
- Public safety, home security, authentication, content management, autonomous driving, medical imaging, manufacturing process control are some of the applications of Computer Vision.
- Computer vision is same as traditional machine learning techniques except it deals with images and videos.
- Image analysis includes object classification, detection and segmentation.
- Video analysis includes instance tracking, action recognition and motion estimation.
- Amazon Rekognition is a computer vision service that is based on deep learning.
- Amazon Rekognition provides image and video detection of faces, sentiment, text, unsafe content and library search.

ACTIVITY LOG

Day	Brief description of daily activity	Learning Outcome	Person in-charge Signature
Day – 1,2,3	Overview of Natural Language processing	Describe the NLP use cases	
Day – 4,5,6	NLP challenges	Able to describe the challenges of working with NLP	
Day – 7,8,9	Pre-processing text	Able to perform stop word removal, normalize and standardize text	
Day – 10,11,12	Performing feature engineering	Able to create tokens and develop features	
Day – 13,14	Capturing content and deriving meaning	Able to classify text, discover similarities and derive relationships	

WEEKLY REPORT

Week 9 & 10

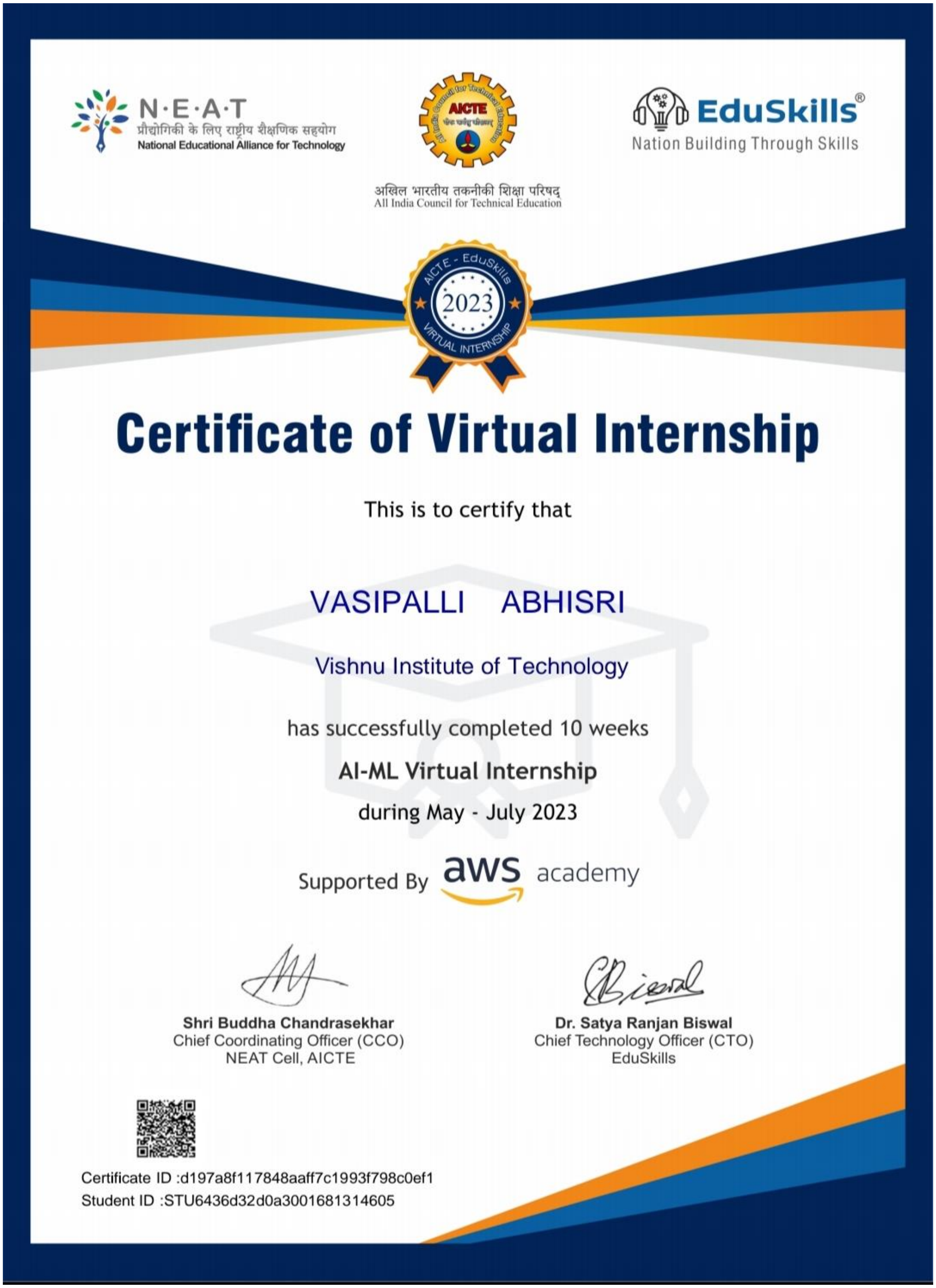
Objective of Activity done:

Is to able to describe the various use cases of Natural Language Processing, describe the challenges working performing NLP, perform removal of stop words, normalize similar text and standardize unrecognized text, convert text into tokens and develop features by applying a model, classify text into categories, discover similarities and derive the relationships among them.

Detailed Report:

- NLP development maps directly to the ML development process.
- NLP is difficult because of the imprecise nature of human language.
- Some of the main use cases for NLP are search query analysis, human-machine interaction, and market or social research.
- Amazon Transcribe can automatically convert spoken language to text.
- Amazon Polly can convert written text to spoken language.
- Amazon Translate can create real-time translation between languages.
- Amazon Comprehend automates many of the NLP use cases that are reviewed in this module.
- Amazon Lex can create a human-like interface to your applications.

Certification



N·E·A·T
प्रौद्योगिकी के लिए राष्ट्रीय शैक्षणिक सहयोग
National Educational Alliance for Technology

AICTE

अखिल भारतीय तकनीकी शिक्षा परिषद
All India Council for Technical Education

EduSkills®
Nation Building Through Skills

AICTE - EduSkills
2023
VIRTUAL INTERNSHIP

Certificate of Virtual Internship

This is to certify that

VASIPALLI ABHISRI

Vishnu Institute of Technology

has successfully completed 10 weeks

AI-ML Virtual Internship

during May - July 2023

Supported By aws academy

Shri Buddha Chandrasekhar
Chief Coordinating Officer (CCO)
NEAT Cell, AICTE

Dr. Satya Ranjan Biswal
Chief Technology Officer (CTO)
EduSkills

Certificate ID :d197a8f117848aaff7c1993f798c0ef1
Student ID :STU6436d32d0a3001681314605