AI-LABS

AI-Tech UK - OPEN-SOURCE TOOLS, TECHNOLOGIES, AND PLATFORMS

GET STARTED WITH OUR HANDS-ON AI LABS

Abstract

Instructions for a hackathon centred around problems using an open solution platform

20th April 2023

Table of Contents

Agenda

Hackathon -Hands on AI Day – Thursday 20th April 2023

9:30 – 10:00 - Registration

10:00 – 10:50 – Setting the scene

10:50 – 11:00 - Grouping & Assessment

11:00 – 12:00 – Breakout for Ideation

12:00 - Checkpoint

12:30 – 1:00 – Lunch

1:00 – 3:00 – Resume Hackathon

3:00 - Checkpoint

4:00 – 5:00 - Show & Tell

5:00 - Pitch & Assessment

Before you start

Think about what you hope to get out of this hackathon. Do you want to focus on just one technology and get the best model you possibly can, or do you want to understand all the technologies? Either is perfectly valid.
Remember learning is a journey not a destination, you can get a working model in a short period of time but what will you have learnt if you just stop there?

Overview of Problem Uses

There are four areas of AI that can be explored as part of the AI Tech UK's – open-source Hackathon offering of coding in Python. These are:

01 Data Exploration
02 Machine Learning Algorithms and Applications
03 Computer Vision
04 Natural Language Processing

Each area will have a series of problems with associated notebooks and data files. For each area a structured approach is provided detailing the objective, dataset, algorithms and models and accompanying libraries, useful links, success criteria and in some cases what to do next (a question, a discussion, use of your own data, refer to a specific link/lab/task, or simply reading further resources of your choice).

01 Data Exploration

Introduction

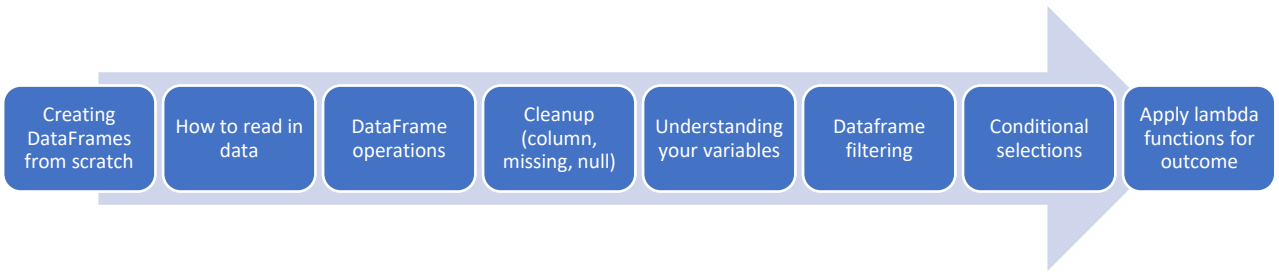
Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest.
There are a series of problems for data exploration starting off with some very simple data, wot working with files to and working through problems that have large dataset and more data issues. These include:

1. Simple Data Exploration Part I and Part II
2. Data exploration – Football Dataset
3. Data exploration – Automobile
4. Data exploration – Supermarket Analysis

1 Simple Data Exploration Part I and Part II

Objective

In data exploration- Part 1 and Part II we focus on the form of the data, and how to get it ready so that you can start cleaning it, transforming, and analysing it.
In Part 1 we work on the core components of pandas that are series and dataframes. You will use the Part1 Notebook file to complete the tasks of the Part II – starting from step 3 onwards – dataframe operations.



Approach

For example, say you want to explore a dataset stored in a CSV on your computer. Pandas will extract the data from that CSV into a DataFrame — a table, basically — then let you do things like:

- Calculate statistics and answer questions about the data, like
    - What's the average, median, max, or min of each column?
    - Does column A correlate with column B?
    - What does the distribution of data in column C look like?
- Clean the data by doing things like removing missing values and filtering rows or columns by some criteria
- Visualize the data with help from Matplotlib. Plot bars, lines, histograms, bubbles, and more.
- Store the cleaned, transformed data back into a CSV, other file, or database.

Dataset

For Part I you will use a dictionary to store your simple data of purchases of fruit. In Part II you will use the same dictionary data but from Purchases.csv - a simple Excel csv file with 3 columns and 4 rows. This is the same data in Part 1 but stored in a file, but you will read from this file. You will work with the IMDB-Movie-Data.csv file to carry out the dataframe operations, clean the data carry out some analysis based on conditional sections and applying functions.

Libraries

Library used is Pandas – a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool.

Algorithms and models

As this is a simple data exploration exercise – you will use carry out a range of dataframe operation using a range of methods to clean the data and then to filter against conditions using a lambda python function.

Success criteria.

1. To be able to create a dataframe from imported data from a file
2. To be able to carry out a range of dataframe operations to get a summary of the data
3. To understand the role of variables in working with the data to analyse it further.
4. To be able to filter or query data like a database but using python functions to retrieve results

Useful links

- The ultimate guide to cleaning data

- Data exploration tools and techniques
- Comprehensive guide to data exploration

What to do next

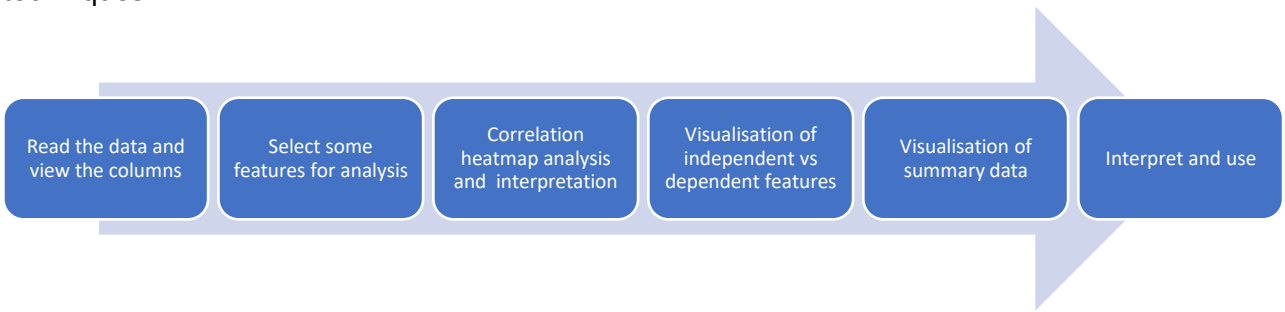
To review the useful links and explore data in problem domains with the other labs where their business rules to be followed to explore and make sense of the data.

2 Data exploration – Football Dataset

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis and to check # assumptions with the help of summary statistics and graphical representations. We will use a range of libraries and techniques to explore data and consider the efficiency of getting the results. In this lab, some of the steps earlier will be grouped together.

Objective

To explore football match data to make financial decisions.
As football is a global multi-billionaire business making decisions on who should play for what the club, and matches is based on the players behaviour during the match. This data is based on text – video analysis will provide further explorative analysis and support decision making and find some reason why a player has the score they have. Here we will introduce a range of visualisation techniques.



Approach

Here descriptive summaries are created to understand the football data and focus on specific variables to analyse their relationship or correlation with other variables. Visualizations are used to present the data at various interim stages and results,.

Dataset

'Footballmatchdata' is a dataset with 89 columns of 18206 rows of data. This has a lot of data. It has information about football players, their personal and financial information, player behaviour and overall performance scoring and salary equivalence. As football is a global multi-billionaire business making decisions on who should play for what the club, matches and players

Libraries

Libraries used are:

- Pandas – a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool.
- NumPy - to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.
- Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structures. Seaborn helps you explore and understand your data.
- For efficiency **Pandas profiling** is an open source **Python** module with which we can quickly do an exploratory data analysis with just a few lines of code.

Algorithms and models

The technique of correlation is used to compare variables, to ascertain any relationship. Here Seaborn correlation heatmaps present the same information in a visually appealing way. What more: they show in a glance which variables are correlated, to what degree, in which direction, and alerts us to potential multicollinearity problems.
Heatmap have various arguments one of which is the colour palette customise to show stronger correlation (normally darker) and weaker correlation are lighter shades (but this can vary). For example, you will see a score of 0.94 identifying a strong correlation between dribbling and ball control which is a known football fact, but the data also states this.
Use of visualization techniques such as cat plot, box plots and joint plots to view independent vs dependent variables. For example, to visualise the overall score of a player (independent) and their age (dependent) or the player position (dependent).

Success criteria

1. To be able to generate descriptive analytics and selective summaries of the data.
2. To utilise the correlation technique to create a starting point to select features for further analysis
3. To create visualization of various combinations of independent variables (sometimes summarised) vs dependent variable to discover patterns to spot anomalies and to create a plan for further analysis.

Useful links

Data Visualization in Python with matplotlib, Seaborn, and Bokeh

What to do next

To explore data based on problem domains and specific business rules using more techniques and specific algorithms before training the data.

3 Data exploration – Automobile Dataset

Objective

In this data exploration and visualization lab the goal is data discovery, from an automobile dataset which is used for several business functions from procurement to transactions to customer support. Here more refined statical techniques are used to understand the data, to help normalise it for further use in creating and training a model.

Approach

This utilises a pipeline of tasks to support the data exploration and preparation for the next stage of creating of models.

Dataset

The Auto MPG sample data set is a collection of 398 automobile records from 1970 to 1982. This not a large dataset but used for data discovery.
It contains attributes like car name, MPG, number of cylinders, horsepower, and weight.
This is a great sample data set to explore and visualize using python.

Libraries

- OS – a library to support working with files and their storage folders.
- Pandas – a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool.
- NumPy - to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.
- Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structures. Seaborn helps you explore and understand your data.
- For efficiency **Pandas-profiling** is an open-source **Python** module with which we can quickly do an exploratory data analysis with just a few lines of code.
- SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics, and many other classes of problems.

Algorithms and models

Use of aggregation(mean, min, max) to group data. To create a range of custom functions to find missing and duplicate data. Use of visualization to view the data and establish data such as outliers that should be excluded using multivariate analysis. To visualise pair-to-pair relationships and correlation to analyse the data. The use of SciPy statistical methods to support the error rates and to normalise the individual features and grouped features for distribution visualisations using normality, normality, Homoscedasticity and Linearity.

Success criteria.

1. To be able to create custom functions to explore variables
2. To be able to clean the data using a range of techniques and appropriate for the business rules
3. To create visualisation of all the attributes
4. To create a correlation of features for analysis
5. Data exploration by excluding data (outliers) by multivariate analysis.
6. To explore and check/test the data using statistical functions to support distribution visualisations and error rates.

Useful links

- Missing values
- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html

What to do next

- Other data quality issues such as: Inconsistent formats, Features importance, Misleading correlation, or data dependency; Data imbalance
- 9 data quality issues than can side-line AI projects
- Challenges for Data Governance and Data Quality in a Machine Learning Ecosystem

4 Data exploration – Supermarket Sales Analysis

This involves Supermarket sales exploration and analysis from different perspectives. The results from this are used to consolidate ideas for features analysis and answer specific questions. For example, 'does customer type influences the sales'.

Objective

Approach

This a pipeline of tasks to analyse the data from each perspective and then to establish key variables for feature analysis and prediction.

Dataset

Upload the dataset – 'Supermarket- sales' which containing information on sales, products, payment, and customer perspectives. Contains 1000 rows with 17 columns.

Libraries

- OS – a library to support working with files and their storage folders.
- Pandas – a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool. NumPy - to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices. Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structures. Seaborn helps you explore and understand your data.

Algorithms and models

Here a range of techniques are used including aggregation, summaries, data manipulation based on object types, unique value selection, visualisation plots of both individual, group and pair-wise features and finally filtering data based on different level of data for example – customer type and sales.

Success criteria.

- To be able to create descriptive analytics and visualizations for each perspective with reference to the business rules and individual meta data of the variables
- To be able to create aggregations and visualisations to support specific queries
- To collate aggregate data and feature variable data to establish a query for analysis.

Useful links

Creating a Simple Recommender System in Python using Pandas

What to do next

To explore working on a classification, and clustering techniques on the data.

02 Machine Learning Algorithms and Applications

Introduction

In the era of Industry 4.0, the abundance of data from various sources presents an opportunity to develop smart applications using artificial intelligence (AI), particularly machine learning (ML). ML allows software applications to make predictions without explicit programming by analysing historical data. Recommendation engines are a popular application of ML, as well as fraud detection, spam filtering, and predictive maintenance.
There are a series of problems for machine learning and recommendations starting off with some very basic regression, to more advanced level of movie recommendation systems. These include:
Linear regression
Breast Cancer detection
Diabetes detection
Recommendation system – Movie recommendation based on user profiles

Objective

- Understand the Fundamentals of regression methods
- Explore Naive Bayes and its applications in medical data analysis
- Understand Content-Based Recommender System and Collaborative filtering

Approach

You will have the opportunity to explore different machine learning algorithms, including Naive Bayes for cancer and regression for diabetes prediction, and techniques for movie recommendation. Each task will involve a specific dataset and Python notebook that can be uploaded to Google Drive and accessed through Colab. Through guided instruction and experimentation with various parameters, you will gain practical experience in manipulating and analyzing data using machine learning techniques.

Dataset

- Breast Cancer
- Diabetes
- Movies dataset: ex8_movies.mat

Libraries

- Matplotlib
- Seaborn
- Pandas
- Numpy
- Tensorflow
- Sci-kit Learn
- Copy

- SciPy

Algorithms and models

- Naïve bayes
- Linear regression

Success criteria

- To demonstrate proficiency in using regression for numerical data analysis by accurately predicting outcomes and interpreting the results to make informed decisions.
- To proficiently clean, split, and train data for recommendation systems, ensuring data quality and accuracy, and utilizing appropriate techniques to evaluate performance.
- To effectively extract and analyse relevant features in cancer and diabetes data to gain insights into the diseases and develop accurate predictions for diagnosis and treatment.
- To thoroughly explore user profiles and movie data, utilizing correlation analysis to identify meaningful relationships, and successfully building, testing, and evaluating recommendation performance to enhance user experience.

Useful links

- Machine Learning Techniques:
- Recommendation systems and machine learning: driving personalization:

What to do next

Machine learning and recommendation systems have immense potential to transform various industries by enabling intelligent analysis and automated decision-making based on vast amounts of data. By leveraging machine learning algorithms, businesses can gain valuable insights into customer behaviour, preferences, and patterns, leading to more personalized and effective recommendations. This, in turn, can lead to increased customer satisfaction, loyalty, and revenue. In addition, machine learning can be used to automate tedious and repetitive tasks, freeing up time for employees to focus on more complex and strategic initiatives. Overall, the potential of machine learning and recommendation systems lies in their ability to extract meaningful insights from data, driving innovation and growth in a wide range of industries.

03 Computer Vision

Introduction

Computer vision is a field of artificial intelligence that uses algorithms and deep learning models to analyse and extract information from visual data. It enables machines to perform various applications such as face recognition, visual attention, and image classification. Face recognition involves identifying an individual's identity using their facial features, while visual attention analyses an image or video to determine the relevant parts. Image classification uses deep neural networks to categorize images into various classes based on their content. Computer vision has significant applications in healthcare, security, retail, and entertainment industries, among others, where visual data analysis and interpretation are essential for informed decision-making and process improvement.

There are a series of problems for image and video analysis starting from basic image processing where you can manipulate a given image and add different artefacts in it. Other areas include face detection, face recognition, face emotion analysis and visual attention as listed below:
Simple digital image processing
Image classification
Face analysis – face detection, face recognition, age, gender, emotion and race prediction through face analysis
Visual attention modelling – predicting the area of interests in a given visual (image/videos)

Objective

- Understand the Fundamentals of Image and Video Processing: learn the basic concepts of digital image and video processing, including image and video formats, pixel manipulation, filtering, and feature extraction.
- Learn Image Classification Techniques: learn about image classification techniques, including traditional machine learning models and deep learning models such as Convolutional Neural Networks (CNNs).
- Perform Comparative Analysis of Face Detection Models: learn about the most commonly used face detection models, including Haar cascades, and deep learning-based models. You will perform a comparative analysis of these models, evaluating their strengths and weaknesses.
- Explore Face Recognition and Emotion Analysis: learn about various face recognition techniques, including DeepFace.
- Understand Visual Attention Modeling: learn about various visual attention modeling techniques, including saliency-based models, which can help identify the most salient regions in an image or video.

Approach

For each task, you will be providing with a dataset and a Python notebook that can be uploaded to your Google Drive and opened in Colab for testing, manipulation, and analysis. You will be guided through the notebook with step-by-step instructions and code snippets, and have the chance to experiment with different parameters and configurations to improve your results. We encourage you to go through the given codes, ask questions, and share your findings. By the end of the session, you will have a better understanding of image and video processing, face detection and recognition, emotion analysis, visual attention modelling, and how to use Google Colab and Python to implement these techniques in your own projects.

Dataset

Some generic images and videos are provided for testing purposes, but we do not use any specific dataset. You are encouraged to use your own image and video datasets to test and analyse the computer vision algorithms covered. This will give you the opportunity to work with real-world data and tailor the techniques to your specific use cases. We will provide guidance on how to load and pre-process your own datasets in the Python notebooks, and offer suggestions for additional datasets that you can explore on your own.

Libraries

Libraries that you would need to use are:

- OpenCV
- TensorFlow
- Keras
- RetinaFace
- DeepFace
- CV2
- MatPlotLib
- NumPy
- PIL

Algorithms and models

- OpenCV is a popular computer vision library that includes tools for image and video processing, feature extraction, and machine learning.
- RetinaFace is a popular face detection algorithm that uses a deep learning model to detect faces in images and videos.
- DeepFace is a deep learning-based facial recognition library that can be used to identify and verify individuals in images and videos.
- Convolutional Neural Networks (CNNs) are a popular type of neural network used for image classification tasks.

Success criteria

- To proficiently perform basic image processing tasks, showcasing a deep understanding of relevant techniques and their applications.
- To conduct comprehensive facial analysis, including face detection, recognition, and emotion analysis, utilizing appropriate methods and demonstrating expertise in the field.
- To gain a thorough understanding of the visual attention model, successfully building and testing it to identify important visual features in images.
- To extensively explore and test image classification using convolutional neural networks (CNNs).

Useful links

- Face recognition with OpenCV, Python, and deep learning
- Deepface: A comprehensive facial analysis framework:

What to do next

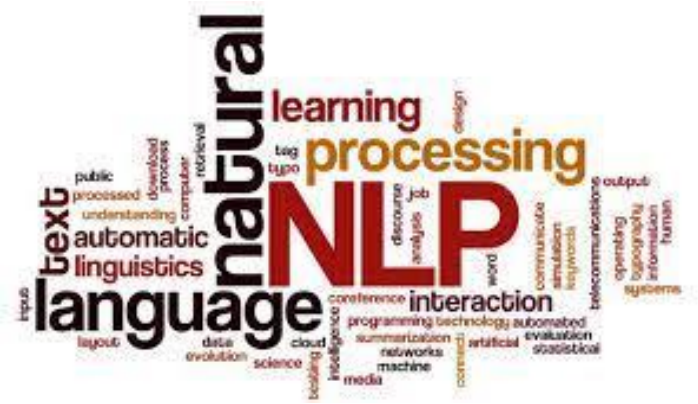
The potential of computer vision in the future is immense, especially for businesses. One can learn a lot from observing how customers navigate their stores, what items they focus on, where production lines slow down, or which inventory requires replenishment. However, it's not feasible for a human to keep track of every aspect of a business all the time. This is where computer vision comes into play.

AI and computer vision-based solutions have the potential to be deployed by enterprises today and can be adapted for future use cases. With computer vision, companies of all sizes can leverage AI on edge devices like cameras, edge servers, or even in the cloud. The applications of computer vision are vast, including biometrics such as face and gait recognition, visual surveillance, medical imaging analysis, and in-store customer behaviour analysis.

By utilizing computer vision, businesses can automate their processes, optimize their operations, and enhance their customer experience. Computer vision's potential to uncover valuable insights and trends in data analysis will continue to increase, leading to more effective decision-making and better overall performance.

04 Natural Language Processing

Introduction



Natural Language Processing is about how we interact with computers and human language to perform useful tasks. NLP is a branch of artificial intelligence (AI).
NLP involves techniques, trends and technologies deployed in a range of powerful business cases and applications. The global pandemic has brought the future forward by five years, due to AI adoption, investment, and latest NLP language models.

NLP is performed on text collections (corpora, plural of corpus)

- Tweets
- Facebook Posts
- Conversations
- Movie Reviews
- Documents, Books and many more

To use a range of NLP libraries, techniques, and datasets for specific use cases such as sentiment analysis, machine translation and chatbot development (in lab 3) and appreciate its limitation

Nuances of meaning make natural language understanding difficult

- Text's meaning can be influenced by context and reader's "world view"
- Text is highly contextual, ambiguous, and irregular

Range of NLP use cases:
1 NLP – language functions
2 NLP – document comparison
3 NLP – simple chatbot
4 NLP – spam detection

1 NLP – language functions

Objective

Approach

NLP involves a pipeline of common tasks. Depending on the specific task and goal various other tasks will be performed to the textual data and different approaches to the representation of the data, which will impact the analysis and results.

Dataset

Range of textual data stored in data structures and files such as 'RomeoAndJuliet.txt' and similar documents for comparison.

Libraries

- NLTK(Natural Language Toolkit) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. NLTK has been called "a wonderful tool for teaching and working in, computational linguistics using Python," and "an amazing library to play with natural language."
- TextBlob - A library build on top of NLTK - for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
- WordCloud - package helps us to know the frequency of a word in textual content using visualization
- Spacy - a free, open-source Python library that provides advanced capabilities to conduct natural language processing (NLP) on large volumes of text at high speed. It helps you build models and production applications that can underpin document analysis, chatbot capabilities, and all other forms of text analysis.
- Imageio is a Python library that provides an easy interface to read and write a wide range of image data, including animated images, volumetric data, and scientific formats.
- NumPy is a library for
- SkLearn library for predictive data analysis – built on NumPy, SciPy, and MatPlotlib – we will use cosine similarity and TDIFvectorise
- **WordNet** is an English word database created by Princeton University **TextBlob** uses NLTK's WordNet interface to look up word definitions, and get **synonyms** and **antonyms**

Algorithms and models

Following on from the approach – there are various NLP techniques and language models used. In this case simple one – but on the other end of the spectrum there are large language model used for training and testing data in real life applications.
Some examples include: tokenisation, string methods and comparisons; part of speech tagging to determine part of speech to determine meaning; extracting noun phrases, stop word elimination; Sentiment Analysis with TextBlob's Default Sentiment Analyzer; language detection and translation, inflection: pluralisation and singularization; speech checking and correction, normalisation: stemming and lemmatization; word frequencies, getting Definitions, Synonyms and Antonyms from WordNet; deleting top words, Ngram (sequence of n text terms); Visualizing Word Frequencies with Bar charts and Word Clouds; Getting the top 20 words; Named Entity Recognition (NER) with SpaCy to determine the text is about; similarity detection accuracy model (simple ~ 40MB /medium ~ 91 MB /large sized ~788MN) and cosine similarity algorithm

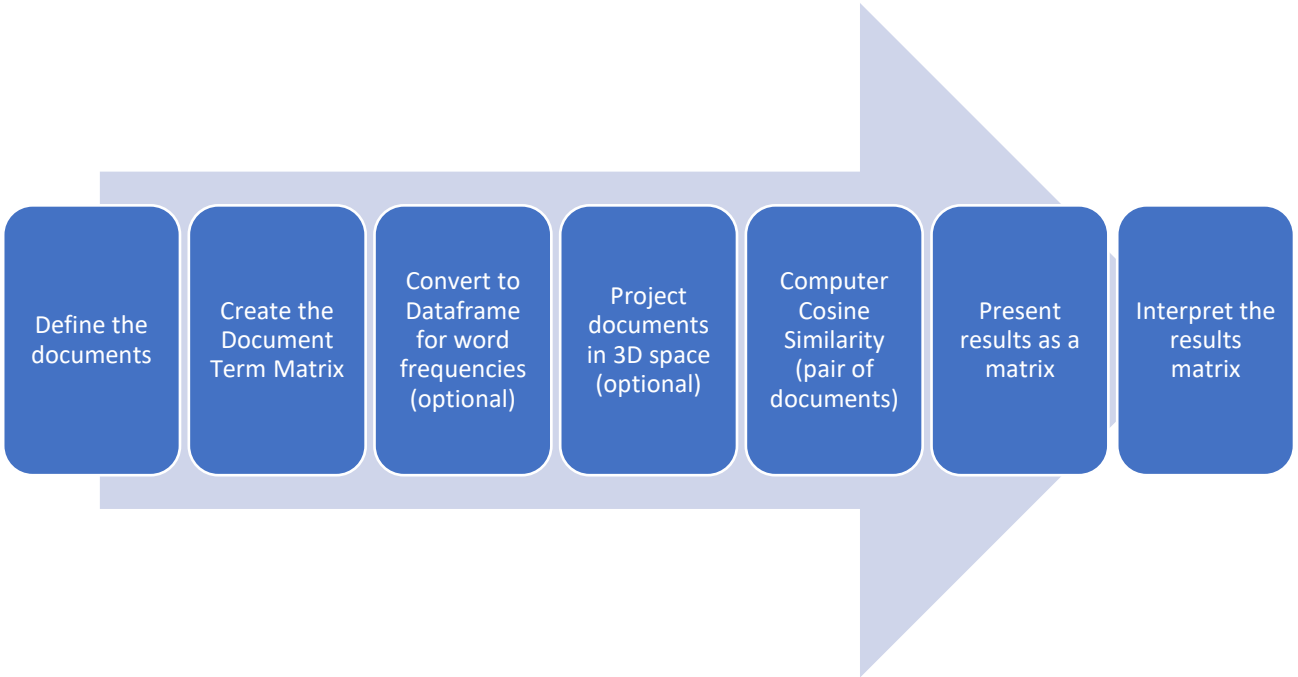
Success criteria.

- To be able to use a range of NLP libraries, techniques and datasets for specific use cases and explore data and make some decisions on the text
- To appreciate some limitations of accuracy, understanding, meaning and context

2 NLP – document comparison

Objective

A common NLP task is comparing documents for similarity and differences. For example, checking an assignment for Plagiarism. Here we will look at some simple documents but use a powerful algorithm used in most document comparisons.



Approach

A document set will have the documents read, defined, and saved in a dataframe for both presenting the word frequencies and to apply the cosine similarity to the pairwise document of the document set to create a matrix presenting the similarities. 1 denotes a 100% match of the same document. The higher the value under 1 the higher the similarity.

Dataset

Here the textual data is saved in strings. This can be replaced by text files that are read into a dictionary of strings. This is then vectorised as ion the lab.

Libraries

Use of Scikit Learn Library and feature extraction for the countVectorizer package, and use of metrics.pairwise and cosine similarly package. Pandas library is used to create a dataframe to compute the word frequencies.

Algorithms and models

Feature extraction of text is computed by the count vectorisation model which removes the stop words. This transformed to a sparse matrix to see the word frequencies of each document. Cosine similarity is a metric [ALGORITHMN] used to determine how similar the documents are irrespective of their size. 3 pairs of different documents are compared using the cosine similarity

Success criteria

1. To prepare documents for comparison and create the document term matrix
2. To create a dataframe to present word frequencies
3. To apply cosine similarity to the pairwise dataframe and then interpret the results matrix

Useful links

- Comparing Documents With Similarity Metrics - https://towardsdatascience.com/comparing-documents-with-similarity-metrics-e486bc678a7d
- Top 6 Ways To Implement Text Similarity In Python: NLTK, Scikit-learn, BERT, RoBERTa, FastText and PyTorch - https://spotintelligence.com/2022/12/19/text-similarity-python/

What to do next

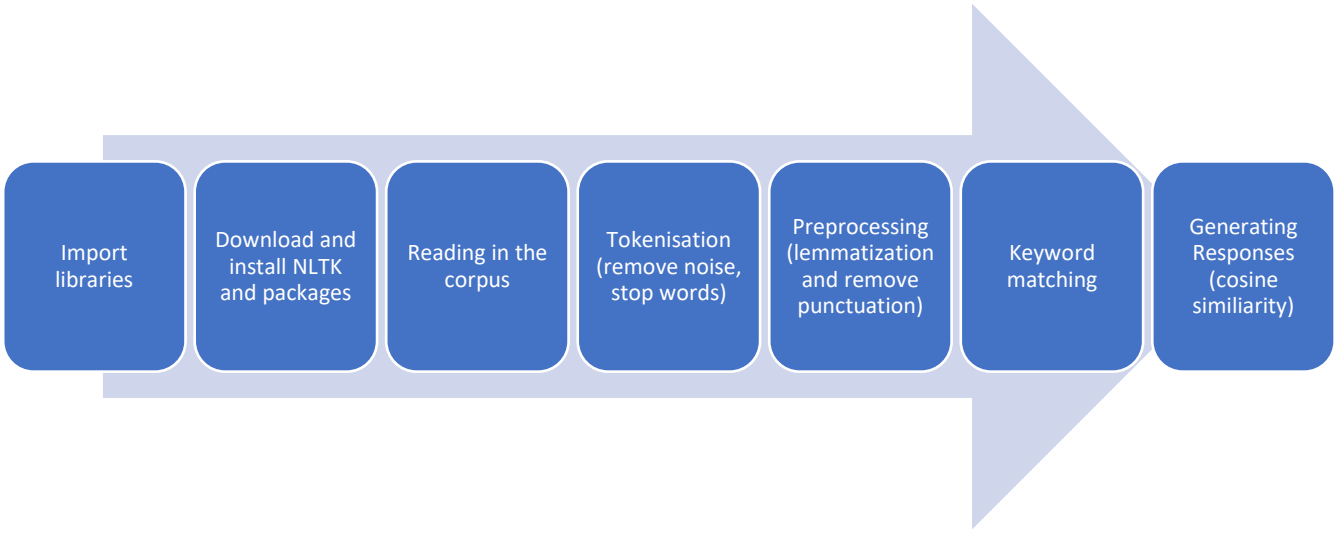
To change the code so that you have real text files for comparison.
Practice recreating the code from the useful links

3 NLP – simple chatbot

You are familiar with chatbots from Google Alexa, Sira and even live chat and more recently ChatGPT (very complex but effective). A simple chatbot is built from scratch using Python. This has no cognitive skills but a good way to get into NLP and get to know about chatbots and understand its limitations and where it needs improvement.

History of chatbots dates to 1966 when a computer program called ELIZA was invented by Weizenbaum. It imitated the language of a psychotherapist from only 200 lines of code. You can still converse with it here: Eliza.

Objective



Approach

Creating a chatbot is based on a textual domain data and questions are asked of the We define a function response which searches the user's utterance for one or more known keywords and returns one of several possible responses. If it doesn't find the input matching any of the keywords, it returns a response:" I am sorry! I don't understand you" chatbot with a natural language generative response. The questions must be structured correctly for adequate keyword matching responses. To generate a response from our bot for input questions, the concept of document similarity will be used. We define a function response which searches the user's utterance for one or more known keywords and returns one of several possible responses. If it doesn't find the input matching any of the keywords, it returns a response:" I am sorry! I don't understand you". You can continue asking question and if you want to exit, type Bye!"

Dataset

A ‘chatbot.txt’ file is read. The text is an extract from a Wikipedia page on chatbots.

Libraries

As stated in Lab 1 – NLP uses range of standard libraries such as the NLTK, and its packages. Here we invoke WordNet Lemmitizer for popular words. Here it invokes io, string, NumPy, random, scikit-learn feature extraction for the TfidVectoriser package and scikit-learn metric pairwise for the cosine similarity package.

Algorithms and models

Common NLP tasks of tokenisation and pre-processing with the WordNet Lemmatizer are used create the lemma or base form of the word – so it is easy to match words. Keyword matching is limited to the dictionary of greeting-inputs and greeting responses – but can be extended for more intents.
The main work is the generating of the response using two different methods bag-of-words (words considered but not the order of them) and Term Frequency-Inverse Document Frequency, or TF-IDF approach to rescale the frequency words by how often they appear in all documents. TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

In any vectorisation work, linear algebra and matrix manipulation is performed. To generate the response the concept of document similarity is used. We define a function response which searches the user’s utterance for one or more known keywords and returns one of several possible responses. If it doesn’t find the input matching any of the keywords, it returns a response:” I am sorry! I don’t understand you”

Success criteria.

1. To be able to install NLTK, necessary libraries and read in the corpus into a raw string for questioning
2. To be able to tokenise and pre-process the raw string into lemma tokens
3. To be able to apply cosine similarity to generate a response for the question posed using simple keyword matching

Useful links

- How To Build Your Own Chatbot Using Deep Learning - https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281
- 14 most popular platforms to build a chatbot https://marutitech.com/14-powerful-chatbot-platforms/
- 26 Best Real Life Chatbot Examples – well-known brands) https://www.tidio.com/blog/chatbot-examples/

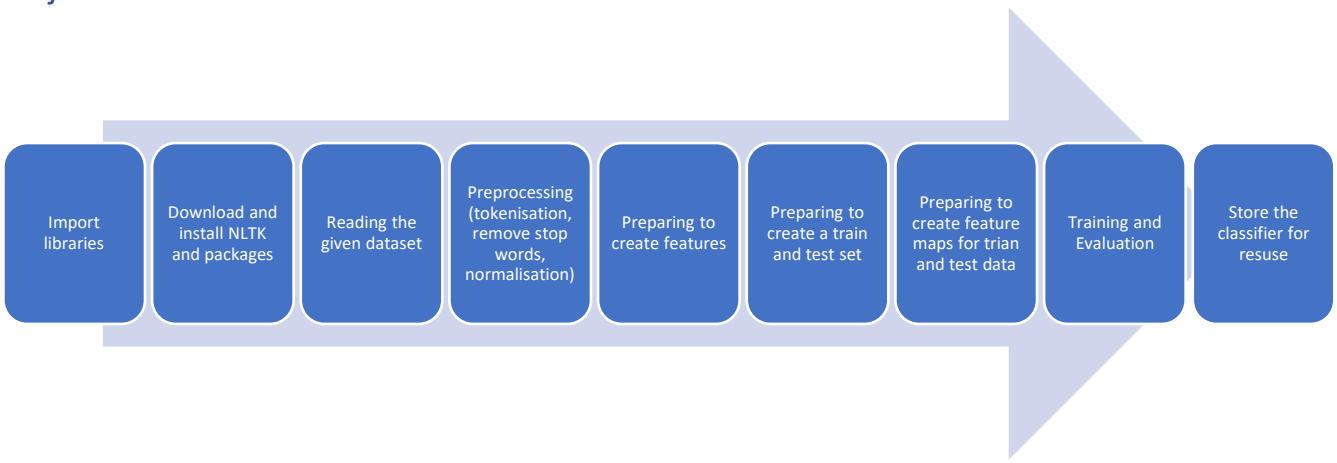
What to do next

To review the link on how to build a chatbot using Deep Learning. What is the difference between our solution and this one. What more needs to be done? What are the challenges of development?

4 NLP – Spam Detection (Email)

This NLP use case detects spam found in e-mails and placed it into a spam folder as opposed to 'ham' folder with acceptable e-mails.

Objective



Approach

This SPAM-HAM detector for e-mails is a recommendation system, based on probabilities of prior knowledge and likelihood of what is a spam e-mail or a ham e-mail. The fundamental Naive Bayes assumption is that each feature makes an independent equal contribution to the outcome. A classifier is created that is evaluated using performance measures of accuracy, precision, and recall.

Dataset

A 'spam message.txt' dataset is used for both training and test purposes. A classifier model is generated as a pickle file for reuse.

Libraries

A range of NLTK packages are downloaded and installed to consider the sentence tokenising, stopword removal, Wordnet knowledge, normalisation of the text via the PorterStemmer, and WordNet Lemmatizer. Also, NLTK will provide methods to classify and apply features to both the training and test dataset. Other libraries include random (to select the train and test messages) and Pandas to read the file into a dataframe and then to create a dataset.

Algorithms and models

The spamClassifer is trained using the Naive Bayes algorithm. This is one of the crucial algorithms in supervised machine learning that helps with classification problems. It is derived from Bayes' probability theory and is used for text classification, where you train high-dimensional datasets. Here we create a binary classifier to make predictions.

Success criteria

1. To be able to read and pre-process the messages ready for creating features
2. To be able to create and train and test set
3. To be able to create features maps for the train and test data
4. To be able to train and evaluate a SpamClassifier

Useful links

- Naives Bayes - https://scikit-learn.org/stable/modules/naive_bayes.html
- Other examples - https://www.geeksforgeeks.org/naive-bayes-classifiers/
- Naïve Bayes Classification- https://www.datacamp.com/tutorial/naive-bayes-scikit-learn
- Classification and its use cases in Machine Learning
    - Recreate either one of the labs from the useful links

What to do next for NLP

Additional mostly free and open-source NLP libraries and APIs:

- Gensim—Similarity detection and topic modelling
- Google Cloud Natural Language API—Cloud-based API for NLP tasks such as named entity recognition, sentiment analysis, parts-of-speech analysis, and visualization, determining content categories and more
- Microsoft Linguistic Analysis API
- Bing sentiment analysis—Microsoft's Bing search engine now uses sentiment in its search results
- PyTorch NLP—Deep learning library for NLP
- Stanford CoreNLP—A Java NLP library, which also provides a Python wrapper. Includes corefererence resolution, which finds all references to the same thing.
- Apache OpenNLP—Another Java-based NLP library for common tasks, including coreference resolution. Python wrappers are available.
- PyNLPl (pineapple)—Python NLP library provides a range of NLP capabilities
- KoNLPy—Korean language NLP
- Latest BERT language models

Machine Learning and Deep Learning Natural Language Applications

- Answering natural language questions—For example, our publisher Pearson Education, has a partnership with IBM Watson that uses Watson as a virtual tutor. Students ask Watson natural language questions and get answers.
- Summarizing documents—analysing documents and producing short summaries (abstracts) that can, for example, be included with search results and can help you decide what to read.

- Speech synthesis (speech-to-text), speech recognition (text-to-speech), inter-language text-to-text translation.
- Collaborative filtering—used to implement recommender systems (“if you liked this movie, you might also like…”).
- Text classification—e.g., classifying news articles by categories, such as world news, national news, local news, sports, business, entertainment, etc.
- Topic modelling—finding the topics discussed in documents.
- Sarcasm detection—often used with sentiment analysis.
- Closed captioning—automatically adding text captions to video.
- Speech to sign language and vice versa—to enable a conversation with a hearing-impaired person.
- Lip reader technology—for people who can’t speak, convert lip movement to text or speech to enable conversation.