INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKURDI, PUNE

Documentation On

"*Personal Voice Assistant using machine learning*"

PG-EDBDA FEB 2020

Submitted By:-

Group No: G-14
Janet Francis-1521
Sharon Yadav-1538

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

Content

TOPICS	Page no
ABSTRACT	
ACKNOWLEDGEMENT	
HISTORY	
PROBLEM STATEMENT	
PURPOSE	
PRODUCT GOALS AND OBJECTIVES	
PRODUCT DESCRIPTION	
SCOPE	
TECHNOLOGIES	
WHY PYTHON?	
SVM	
FEATURES OF PERSONAL ASSISTANT	
FUTURE PROSPECTIVE	
CONCLUSION	
REFERENCES	

ACKNOWLEDGEMENT

We had a great experience working on this project and we got to learn a new skill through this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to the teachers and especially Mr Rahul Pund and Mr Manish for their guidance and constant supervision as well as providing necessary information regarding the project and also for their support in completing the project.

We would like to express our gratitude towards our parents and friends for their kind cooperation and encouragement which help us in the completion of the project.

ABSTRACT

The project aims to develop a personal-assistant for python-based systems. Personal assistant draws its inspiration from virtual assistants like Cortana for Windows, and Siri for iOS.

It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands . Users can interact with the assistant either through voice commands .As a personal assistant, It assists the end-user with day-to-day activities like general human conversation, searching queries in google, and document in local system and image classification on local system. The user statements commands are analysed with the help of machine learning to give an optimal solution

HISTORY

Radio Rex was the first voice activated toy released in 1911. It was a dog that would come out of its house when its name is called.

In 1952 Bell Labs presented “Audrey”, the Automatic Digit Recognition machine. It occupied a six- foot-high relay rack, consumed substantial power, had streams of cables and exhibited the myriad maintenance problems associated with complex vacuum-tube circuitry. It could recognize the fundamental units of speech, phonemes. It was limited to accurate recognition of digits spoken by designated talkers. It could therefore be used for voice dialing, but in most cases push-button dialing was cheaper and faster, rather than speaking the consecutive digits.

Another early tool which was enabled to perform digital speech recognition was the IBM Shoebox voice-activated calculator, presented to the general public during the 1962 Seattle World's Fair after its initial market launch in 1961. This early computer, developed almost 20 years before the introduction of the first IBM Personal Computer in 1981, was able to recognize 16 spoken words and the digits 0 to 9.

The first natural language processing computer program or the chatbot ELIZA was developed by MIT professor Joseph Weizenbaum in the 1960s. It was created to "demonstrate that the communication between man and machine was superficial". ELIZA used pattern matching and substitution methodology into scripted responses to simulate conversation, which gave an illusion of understanding on the part of the program.

PROBLEM STATEMENT

We are all well aware about Cortana, Siri, Google Assistant and many other virtual assistants which are designed to aid the tasks of users in Windows, Android and iOS platforms. But to our surprise, there's no such virtual assistant available for opening document from local machine and doing image classification between images and objects.

PURPOSE

This Software aims at developing a personal assistant for python-based systems. The main purpose of the software is to perform the tasks of the user at certain commands, provided in either of the ways, speech . It will ease most of the work of the user as a complete task can be done on a single command. Personal assistant draws its inspiration from Virtual assistants like Cortana for Windows and Siri for iOS. Users can interact with the assistant either through voice commands or keyboard input.

PRODUCT GOALS AND OBJECTIVES

Currently, the project aims to provide the Windows Users with a Virtual Assistant that would not only aid in their daily routine tasks like searching the web, extracting weather data, vocabulary help and many others but also help in automation of various activities.

In the long run, we aim to develop a complete server assistant, by automating the entire server management process - deployment, backups, auto-scaling,logging, monitoring and make it smart enough to act as a replacement for a general server administrator.

PRODUCT DESCRIPTION

As a personal assistant, Personal assistant assists the end-user with day-to-day activities like general human conversation, searching queries in various search engines like Google, youtube , stackoverflow, image classification, retrieving documents. The user statements/commands are analysed with the help of machine learning to give an optimal solution.

SCOPE

Presently, Personal assistant is being developed as an automation tool and virtual assistant.
Among the Various roles played by voice assitant are:
1. Search Engine with voice interactions
2. Opening document as per voice command
3. Image classification

TECHNOLOGIES USED

1)Python

2)Machine learning

Libraries used

- Numpy
- Matplotlib
- Os
- pyttsx3
- speech_recognition
- datetime
- webbrowser
- walk
- Wikipedia
- datetime

Why python?

AI projects differ from traditional software projects. The differences lie in the technology stack, the skills required for an AI-based project, and the necessity of deep research. To implement your AI aspirations, you should use a programming language that is stable, flexible, and has tools available. Python offers all of this, which is why we see lots of Python AI projects today.

From development to deployment and maintenance, Python helps developers be productive and confident about the software they're building. Benefit that make Python the best fit for machine learning and AI-based projects include simplicity and consistency, access to great libraries and frameworks for AI and machine learning (ML), flexibility, platform independence, and a wide community. These add to the overall popularity of the language.

Simple and consistent

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

Additionally, Python is appealing to many developers as it's easy to learn. Python code is understandable by humans, which makes it easier to build models for machine learning.

Many programmers say that Python is more intuitive than other programming languages. Others point out the many frameworks, libraries, and extensions that simplify the implementation of different functionalities. It's generally accepted that Python is suitable for collaborative implementation when multiple developers are involved. Since Python is a general-purpose language, it can do a set of complex machine learning tasks and enable you to build prototypes quickly that allow you to test your product for machine learning purposes.

Extensive selection of libraries and frameworks

Implementing AI and ML algorithms can be tricky and requires a lot of time. It's vital to have a well-structured and well-tested environment to enable developers to come up with the best coding solutions.

To reduce development time, programmers turn to a number of Python frameworks and libraries. A software library is pre-written code that developers use to solve common programming tasks. Python, with its rich technology stack, has an extensive set of libraries for artificial intelligence and machine learning. Here are some of them:

- Keras, TensorFlow, and Scikit-learn for machine learning
- NumPy for high-performance scientific computing and data analysis
- SciPy for advanced computing
- Pandas for general-purpose data analysis
- Seaborn for data visualization

Scikit-learn features various classification, regression, and clustering algorithms, including support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is designed to work with the Python numerical and scientific libraries NumPy and SciPy.

With these solutions, you can develop your product faster. Your development team won't have to reinvent the wheel and can use an existing library to implement necessary features.

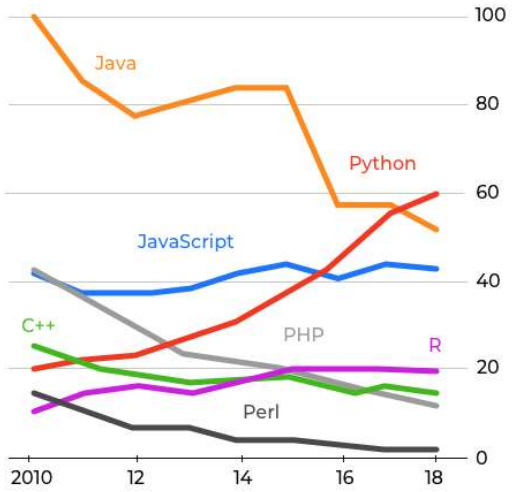
What is Python good for?

Here's a table of common AI use cases and technologies that are best suited for them. We recommend using these:

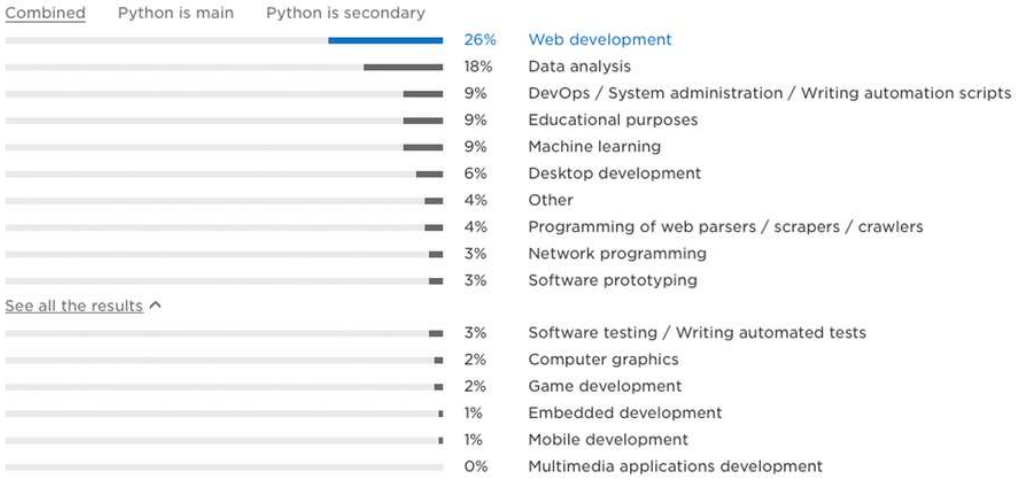
Platform independence
Platform independence refers to a programming language or framework allowing developers to implement things on one machine and use them on another machine without any (or with only minimal) changes. One key to Python's popularity is that it's a platform independent language. Python is supported by many platforms including Linux, Windows, and macOS. Python code can be used to create standalone executable programs for most common operating systems, which means that Python software can be easily distributed and used on those operating systems without a Python interpreter.

What's more, developers usually use services such as Google or Amazon for their computing needs. However, you can often find companies and data scientists who use their own machines with powerful Graphics Processing Units (GPUs) to train their ML models. And the fact that Python is platform independent makes this training a lot cheaper and easier.

Great community and popularity
In the Developer Survey 2018 by Stack Overflow, Python was among the top 10 most popular programming languages, which ultimately means that you can find and hire a development company with the necessary skill set to build your AI-based project.
If you look closely at the image below, you'll see that Python is the language that people Google more than any other.



In the Python Developers Survey 2017, we observe that Python is commonly used for web development. At first glance, web development prevails, accounting for over 26% of the use cases shown in the image below. However, if you combine data science and machine learning, they make up a stunning 27%.



Python as the best language for AI development

Spam filters, recommendation systems, search engines, personal assistants, and fraud detection systems are all made possible by AI and machine learning, and there are definitely more things to come. Product owners want to build apps that perform well. This requires coming up with algorithms that process information intelligently, making software act like a human.

What is image classification?

- Supervised classification
- Unsupervised classification
- Example

Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. Depending on the interaction between the analyst and the computer during classification, there are two types of classification: supervised and unsupervised.

With the ArcGIS Spatial Analyst extension, there is a full suite of tools in the Multivariate toolset to perform supervised and unsupervised classification (see An overview of the Multivariate toolset). The classification process is a multi-step workflow, therefore, the Image Classification toolbar has been developed to provided an integrated environment to perform classifications with the tools. Not only does the toolbar help with the workflow for performing unsupervised and supervised classification, it also contains additional functionality for analyzing input data, creating training samples and signature files, and determining the quality of the training samples and signature files. The recommended way to perform classification and multivariate analysis is through the Image Classification toolbar.

Supervised classification

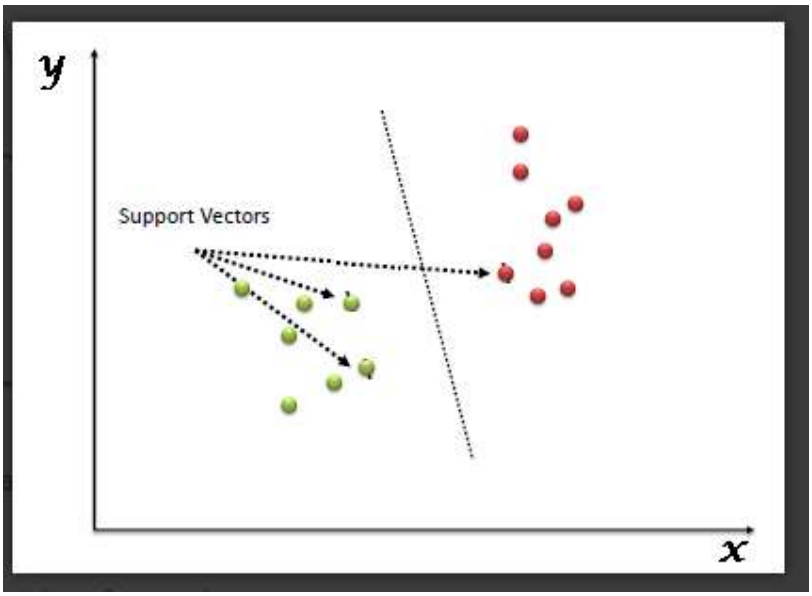
Supervised classification uses the spectral signatures obtained from training samples to classify an image. With the assistance of the Image Classification toolbar, you can easily create training samples to represent the classes you want to extract. You can also easily create a signature file from the training samples, which is then used by the multivariate classification tools to classify the image.

Unsupervised classification

Unsupervised classification finds spectral classes (or clusters) in a multiband image without the analyst's intervention. The Image Classification toolbar aids in unsupervised classification by providing access to the tools to create the clusters, capability to analyze the quality of the clusters, and access to classification tool.

What is Support Vector Machine?

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



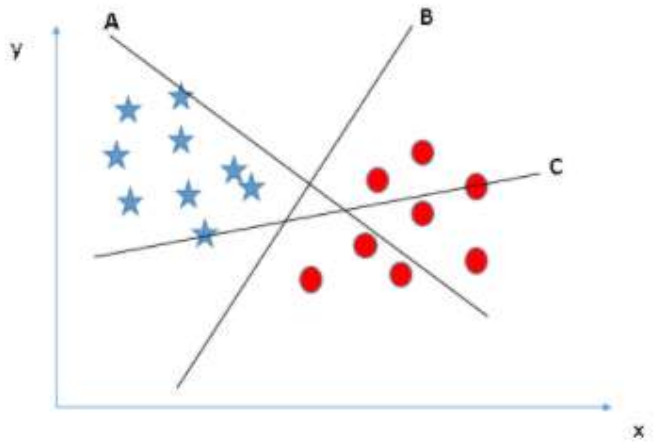
Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

How does it work?

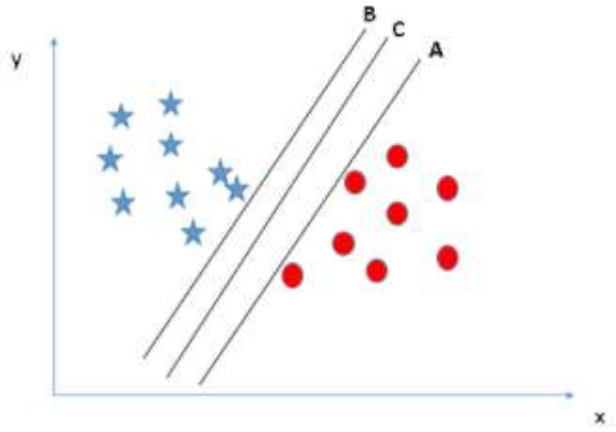
Above, we got accustomed to the process of segregating the two classes with a hyper-plane. Now the burning question is "How can we identify the right hyper-plane?". Don't worry, it's not as hard as you think!

Let's understand:

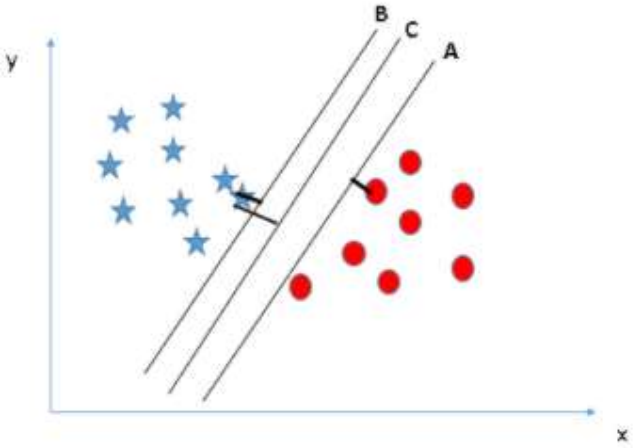
- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.



- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.
- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

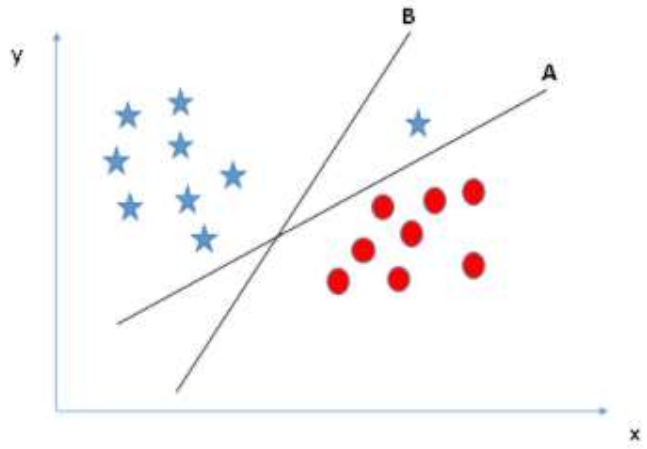


Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as Margin**.** Let’s look at the below snapshot:



- Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane



Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

Why we use SVM for image classification

Support vector machines have one built-in "layer" that helps with having an interpretation of the data - the kernel. You could even use output from some other image classifier, including a neural network, as the kernel. E.g. you could measure how far apart two images are in "classifier space" from a trained neural network (maybe one trained against different targets from the ones you want to classify with the SVM)

There is also an implied "layer" in image feature construction. Common image features are histograms of pixel values, histograms of pixel differences (edge detection), bags of "visual words" and there are many more possible transformations and statistics.

Typically if you are using a classifier other than a deep neural network (which is supposed to discover the features automatically), then you will pre-process the image into a set of features. That pre-processing covers the majority of an approach being "good at images".

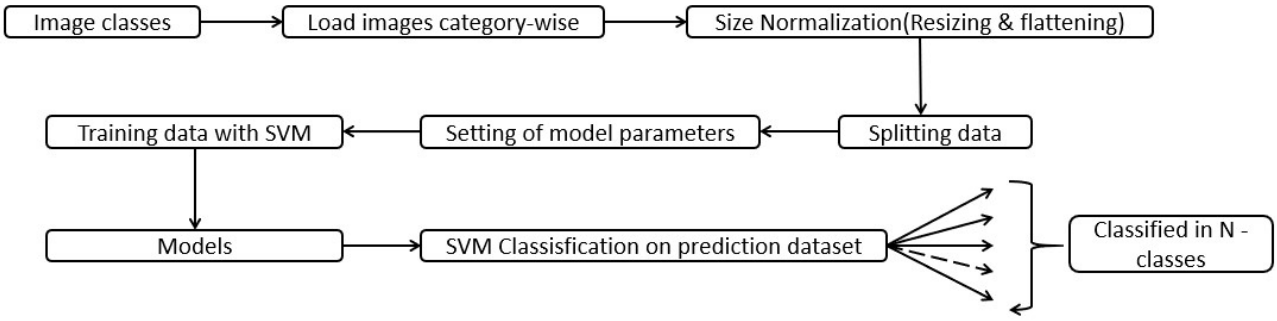
To add some stuffs to what Neil Slater said:

- First I'm not sure you got (as I read your question) the fact that SVMs are quite good, as long as you use them with a proper kernel. Which means that you must find a way of transforming your images such that the resulting input data are linearly separable in the sens of your formula. If the SVM algorithm is very simple, using kernel is nontrivial.
- Then the best approach nowadays for image classification is deep neural network. Not because they are magic but mostly because of the use of convolutional layers. Let say that for 10 000 neurons in a network, 100 will do what SVM do: classification. The rest of it just act as a kernel. In the case of images, since features to recognize are intricately local and global, the use of successive convolutional layers can decorticate your image to retrieve the specifications you need (in your example, you could recognize brushstroke as well as global shapes).

To sum up:

- Shallow learning (SVM for example) work great but you have to design your own kernel, or treat your data before using them.
- Deep learning (convolutional networks for example) do the work for you, but are harder to set up and to interpret.

Flow of image classification :



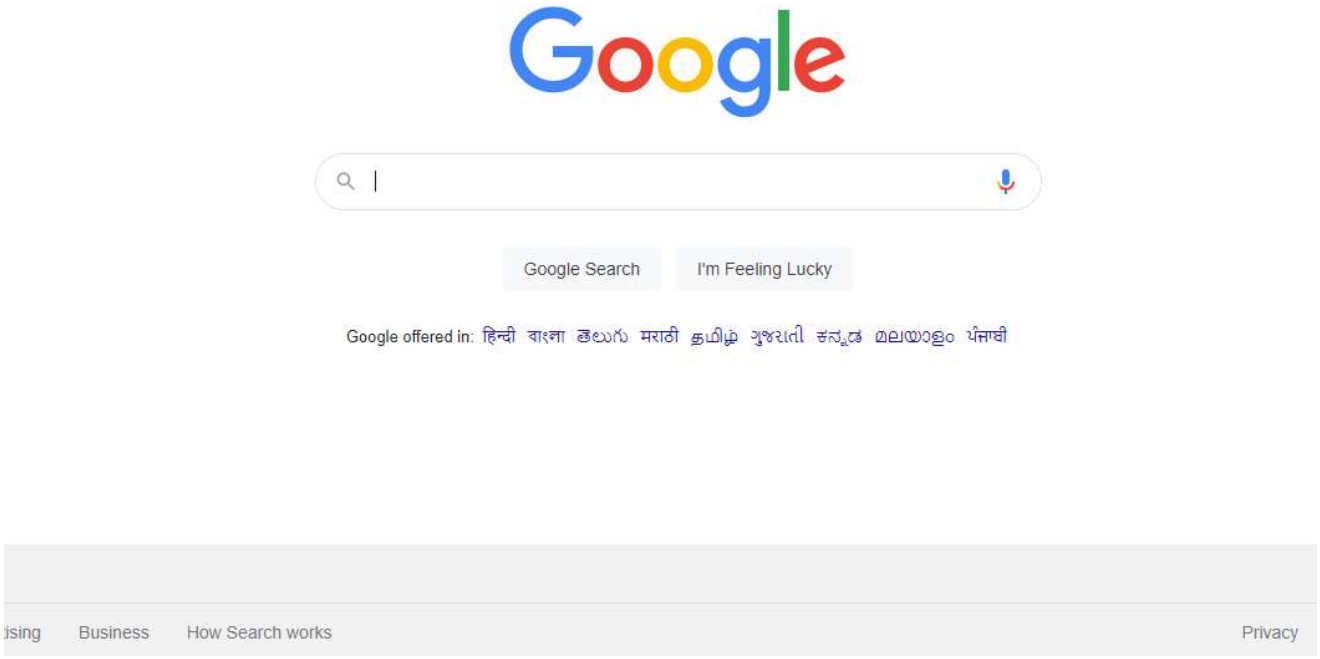
Features in Personal Assistant

1. Queries from the web:

Making queries is an essential part of one's life, and nothing changes even for a developer working on windows.We have addressed the essential part of a netizen's life by enabling our voice assistant to search the web. Personal assistant supports Google,Google displays the result according to voice queries.

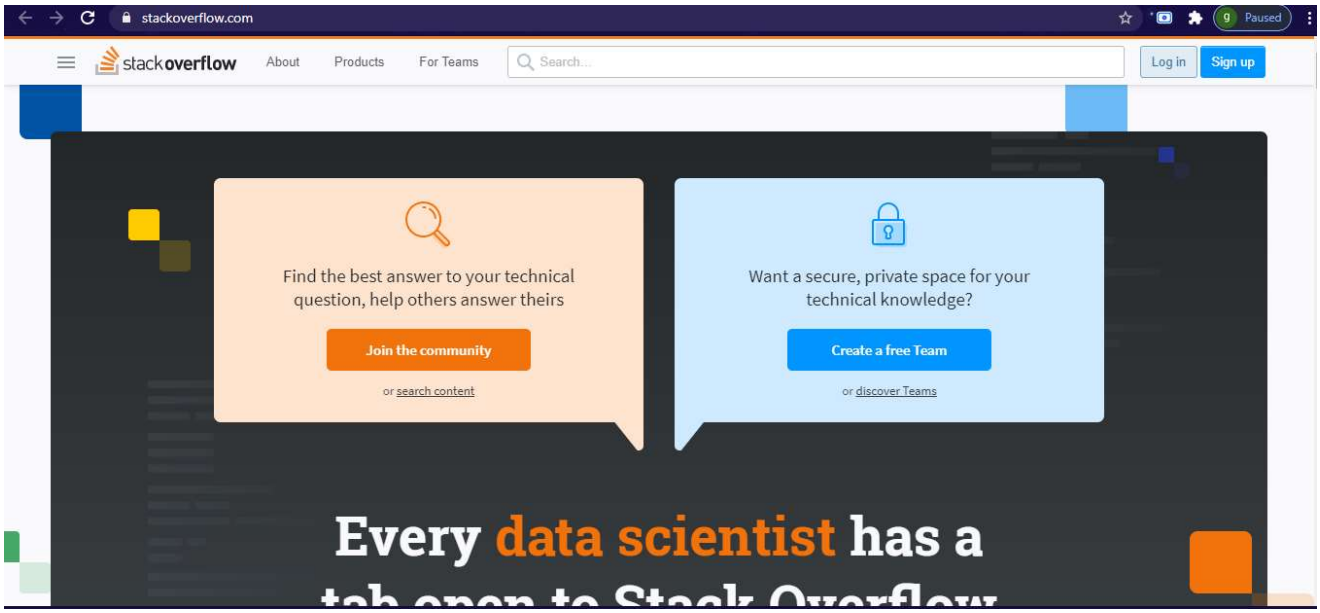
```
In [7]: runfile('D:/assistant/assistant_main.py', wdir='D:/
assistant')
Reloaded modules: module1.my_functions
DESKTOP-GCKP0MO
Listening...
Recognizing...
User said: open Google
```

OPEN GOOGLE



OPEN STACKOVERFLOW

```
stening...
cognizing...
er said: open stack overflow
```



OPEN YOUTUBE

```
Listening...
Recognizing...
User said: open YouTube
```

IT STOP'S WHEN USER SAYS STOP

```
Recognizing...
Listening...
Recognizing...
User said: stop


In [9]:
```

2.Image classification

Have you ever stumbled upon a dataset or an image and wondered if you could create a system capable of differentiating or identifying the image?

The concept of image classification will help us with that. Image Classification is one of the hottest applications of computer vision and a must-know concept for anyone wanting to land a role in this field.

Our personal assistant is designed to do that also we will see a very simple but highly used application that is Image Classification. Not only will we see how to make a simple and efficient model classify the data but also learn how to implement a pre-trained model and compare the performance of the two.

Accuracy of transportation model

```
datadir = 'C:/Users/Sharon/Desktop/newimages/images'
categories = ['aeroplane','car','cycle']
```

```
Started
   Resizing and Flattening completed...
   SPLITTING DATA...
  SVC Model data...
   Modelling part completed...
Predixted value : [1 1 1 0 0 2 2 2 1 0 1 2 2 1 1 0 1 1 0 2 2 0 1
2 0 2 0 1 2 2 0 2 2 0 0 0 2
 0 1 1 1 2 1 0 2 1 2 0 0 1 2 1 0 0 0 0 2 1 2 1 0 0 2 2 2 1 1 0 2
2 2 0 0 0
 1 2 1 1 1 2 1 2 1 0 0 0 2 1 1 1 2 2 2 1 0 1 2 0 2 1 1]
Original value :  [1 1 1 0 0 2 2 2 1 0 1 2 2 1 1 2 1 1 0 2 2 0 1
2 1 2 0 1 2 2 0 2 2 0 0 0 2
 0 1 1 1 0 1 1 2 1 2 0 0 1 2 1 0 0 0 0 2 1 2 1 0 0 2 2 2 1 1 2 0
2 2 0 0 0
 1 2 1 1 1 2 1 2 1 0 0 0 2 1 1 1 2 2 2 1 0 1 2 0 2 1 1]
0.9405940594059405
[[27  2  2]
 [ 0 35  0]
 [ 2  0 33]]
```

Accuracy of fruits model

```
datadir = 'C:/Users/Sharon/Desktop/newimages/images'
categories = ['apple','grapes','kiwi','papaya']
```

```
Started
   Resizing and Flattening completed...
   SPLITTING DATA...
  SVC Model data...
   Modelling part completed...
Predixted value : [2 0 1 1 1 2 0 1 2 2 1 0 0 1 3 0 0 2 0 3 0 3 3
1 0 0 1 1 0 1 1 1 2 3 3 1 2
 2 0 3 0 3 3 0 2 3 0 2 2 2 1 2 2 2 0 2 1 2 2 2 1 0 2 2 3 0 1 2 3
1 1 3 2 0
 3 0 2 2 0 2 0 0 3 3 2 1 0 0 1 0 0 3 2 0 1 3 1 2 1 3 3 0 2 0 2 2
0 1 3 3 3
 2 3 0 0 0 1 3 1 1 3 1 1 2 2 2 0 0 3 0 0 3 0 3 0]
Original value :  [2 0 1 1 1 2 0 1 2 2 1 2 0 1 3 0 0 2 0 3 0 3 0
1 2 2 1 1 0 1 1 1 2 3 3 1 2
 2 0 3 0 3 3 0 2 3 0 2 2 2 1 2 2 2 0 2 1 2 2 2 1 0 2 1 3 0 1 1 3
1 1 3 2 0
 3 0 2 2 0 2 0 0 3 3 2 1 0 0 1 0 0 2 2 0 1 3 1 2 1 3 3 0 2 0 2 2
0 1 0 3 3
 2 3 2 0 0 1 3 1 1 3 1 1 1 1 2 0 0 3 0 0 2 0 3 0]
0.9111111111111111
[[36  0  4  0]
 [ 0 30  0  0]
 [ 0  4 32  0]
 [ 2  0  2 25]]
```

Accuracy of animal class (horse and squirrel)

```
datadir = 'E:\\NEWWWWWWWWWWWWWWWWW\\IACSD\\project work\\images_dataset'
categories = ['horse','squirrel']
```

```
In [1]: runfile('E:/NEWWWWWWWWWWWWWWWWW/IACSD/project work/models.py', wdir='E:/NEWWWWWWWWWWWWWWWWW
2021-03-30 22:00:35.971493
Started
   Resizing and Flattening completed...
   SPLITTING DATA...
  SVC Model data...
   Modelling part completed...
Predixted value : [1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1 0
 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 1]
Original value :  [1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1 0
 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1]
0.9846153846153847
[[34  1]
 [ 0 30]]
2021-03-30 22:09:46.372629
```

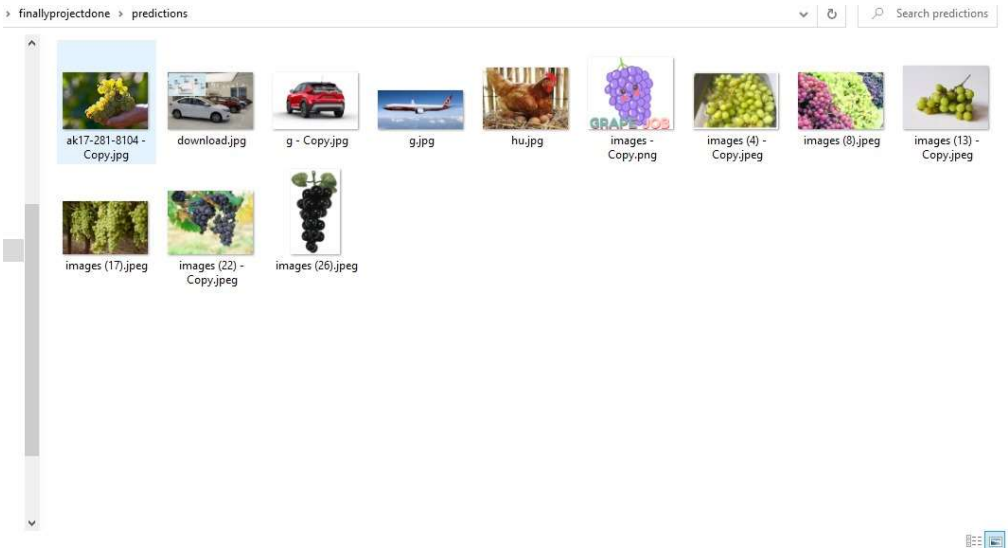
Testing models

```
In [2]: runfile('D:/finallyprojectdone/assistant_main.py', wdir='D:/finallyprojectdone')
Reloaded modules: my_functions
Listening...
Recognizing...
User said: show pictures

waiting for reply:
recognising reply :
User replied: grapes
```

```
recognising reply :
User replied: grapes

D:/finallyprojectdone/predictions/ak17-281-8104 - Copy.jpg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/download.jpg
predicted output : kiwi
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/g - Copy.jpg
predicted output : kiwi
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/g.jpg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/hu.jpg
predicted output : papaya
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/images (13) - Copy.jpeg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/images (17).jpeg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/images (22) - Copy.jpeg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
D:/finallyprojectdone/predictions/images (26).jpeg
predicted output : grapes
-----------------------------------------------------------------------------------------------------------
```

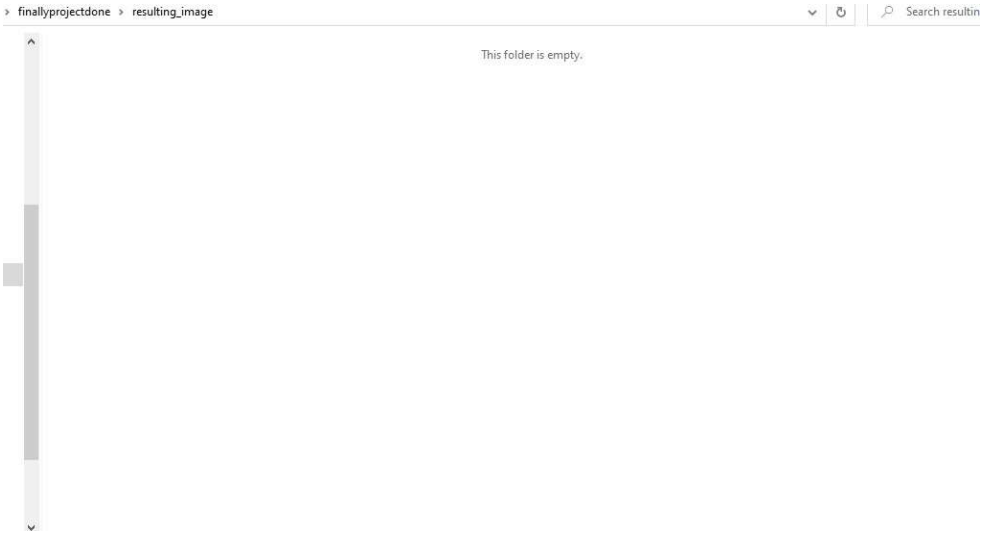
IPython console | History

Predicted on images of below folder :

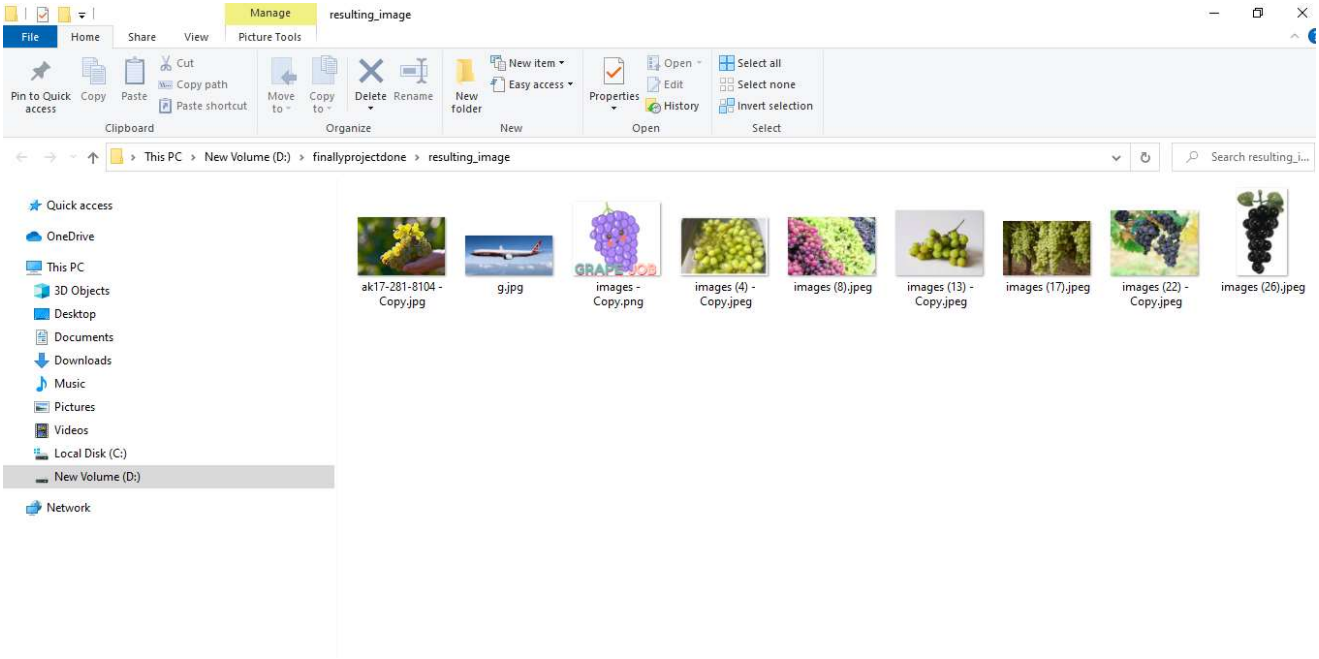


Result Stored in resulting_images folder :

Before prediction :



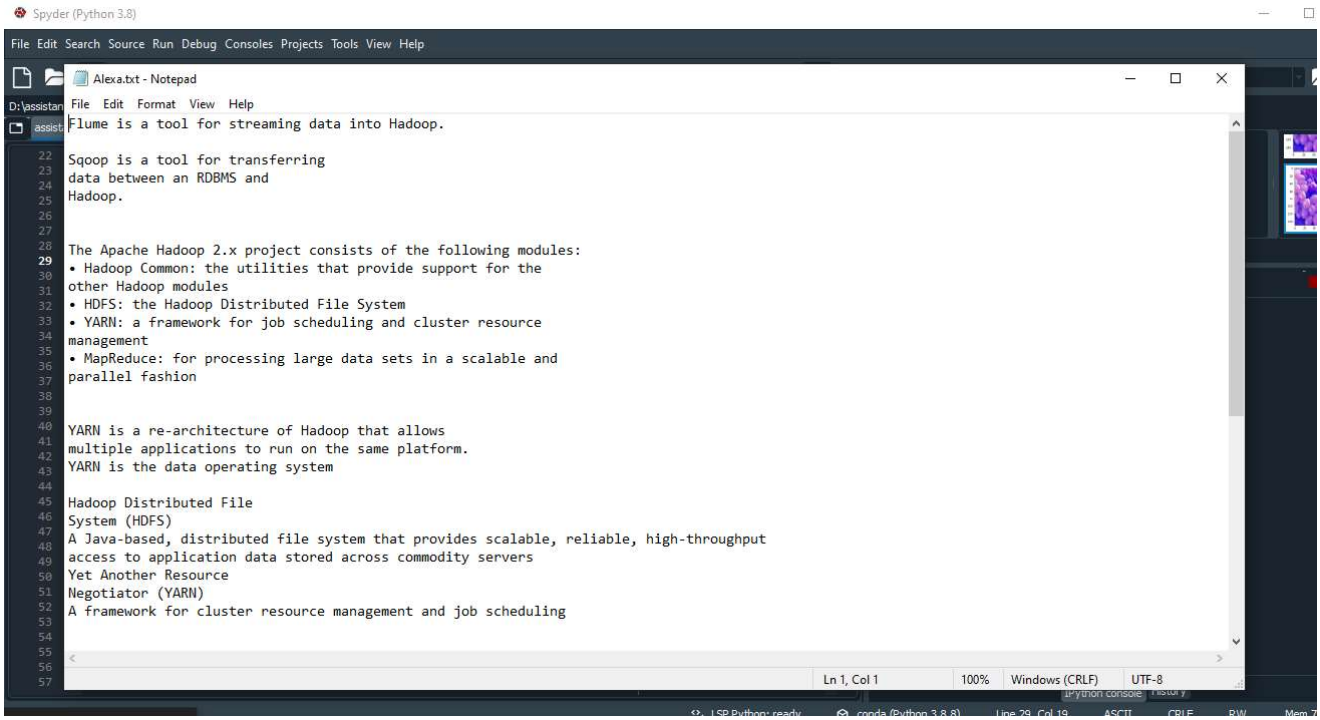
After Prediction : (grapes classified)



3.Open document

Our personal assistant is designed for searching any document or any type of file example jpg from local machine by taking the voice command from the user.It doesn't matter what extension it have

```
In [8]: runfile('D:/assistant/assistant_main.py', wdir='D:/
assistant')
Reloaded modules: module1.my_functions
DESKTOP-GCKP0MO
Listening...
Recognizing...
User said: open document

waiting for reply:
recognising reply :
User replied: d

waiting for reply:
recognising reply :
User replied: Alexa
```



FUTURE PROSPECTIVE

We plan to Integrate Personal Assistant with mobile using react native, to provide a synchronized experience between the two connected devices.Further, in the long run, Personal assistant is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with personal assistant.

CONCLUSION

Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web, retrieving document and image classification . The future plans include integrating Jarvis with mobile using React Native to provide a synchronised experience between the two connected devices. Further, in the long run, Personal assistant is planned to feature auto deployment supporting elastic beanstalk, backup files, and all operations which a general Server Administrator does. The functionality would be seamless enough to replace the Server Administrator with Personal Assistant.

References:

- https://scikit-learn.org/stable/modules/svm.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://scikit-learn.org/stable/
- https://towardsdatascience.com/svm-support-vector-machine-for-classification-710a009f6873
- https://www.analyticsvidhya.com/blog/2020/10/create-image-classification-model-python-keras/
- https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/