

# Sanath Narasimhan

LinkedIn: <https://www.linkedin.com/in/sanath-narasimhan-016b2a13>  
Github: <https://github.com/sanath-narasimhan>  
Website: <https://sananara-arvabhata.netlify.com>

Contact: +1 (682) 228-9743 Email: [sanath.narasimhan@gmail.com](mailto:sanath.narasimhan@gmail.com)

## EDUCATION

**Master of Computer Science**, The University of Texas at Arlington - **3.667 GPA** (Course work: Design and Analysis of Algorithms, Data Modeling and Analytical Techniques, Data Mining, Computer Graphics, Animation, Robotics, Neural Networks, Computer Vision, and Convex Optimization)

**Bachelor of Engineering in Information Science (2013-2017)** Reva Institute of Technology and Management, Bangalore, **Affiliated to** Visvesvaraya Technological University, Belgavi - **69.34%**

## SKILLS

Proficiency in Programming Languages		
Data Science/ Machine Learning	Open CV (python)	Intermediate
	R	Intermediate
	Matlab CV	Intermediate
	Python	Advanced
Web Development	HTML	Intermediate
	Python-flask	Advanced
	JavaScript	Intermediate
Application Development	C++	Advanced
	Shell Scripting	Advanced

- Familiar with machine learning and data mining approaches. Currently studying Kubernetes and docker containerization.
- Implemented various algorithms like linear regression, logistic regression, Decision trees, SVM classifier, and Convolutional Neural Network classifier from scratch on practice datasets from Kaggle and hacker rank.
- Certified in Intelligently Extract Text & Data from Document with OCR NER from Udemy on [05/26/2022](#)

## AWARDS

- Won the best use of Algolia API at the HackUTA event, October 5,6 - 2018
- Awarded the "Lone Star" Non-Resident Tuition Waiver scholarship for Fall 2018, Spring 2019, and Summer 2019 terms By the Department of Computer Science at The University of Texas At Arlington
- Nominated for Non-Resident tuition waiver Scholarship for Fall 2019, Spring 2020, and Summer 2020 terms By Department of Computer Science at The University of Texas At Arlington

## LANGUAGES

English (proficient), Tamil (Mother's tongue), Telugu, Kannada, Hindi

## Experience as Junior Data Scientist Intern at Enquero Global LLP

September 2017 - June 2018

- I was in charge of maintaining the Data Quality metrics dashboard for our client Cisco which involved acquiring new incoming data from the Hive cluster biweekly and monthly, performing validation, and updating the data to a QlikView server.
- Worked on improving the search efficiency for the sales team using hybrid clustering for the better resolution of client branches.
- Created a proof of concept to analyze the churning of partners of Cisco in the Saas services.

## Experience as Apprentice Engineer of Data science at Altimetrik Corp

September 2020 - June 2021

- Digital Lending Platform:
  - Was responsible for developing ML models for the classification of customers in different buckets based on a predefined set of criteria from an excel, automating the process. The models used were SVM, K-means, Xgboost tree, and CNN.
  - Developed and tested functions for table extraction from pdf using libraries like Camelot, PDFMiner and pytesseract, Textract for png and jpeg images
- Paypal Document Classification POC
  - Created a simple CNN classifier for detecting 10 different types of documents, trained on a set of 100 documents collected from the internet, with an f1 score of 0.96.
  - Helped develop a novel training method that combines unsupervised clustering and supervised learning, with silhouette scores to group the data points, using raw image features from inceptionV3, text features from Tf-IDF ranking, and metadata. After the nodes are created, a CNN classifier, NLP classifier, and hybrid classifier are tested to get the optimal model.
  - The technique helps scale up the number of document types that the model can recognize with ease.

- The model was tested with a set of 1000 documents with 10 classes, giving overall accuracy of 72% with hybrid as an optional model at every node.
- The application had to be deployed in a containerized docker hub that interacted with an aws s3 bucket to receive the data.
- Dummy document generator:
  - A python app built with Tkinter to help with preprocessing images.
  - First, the use-case is letting users redact documents by allowing them to draw black rectangles over sensitive information
  - The second use-case is to tag entities within an image for NER.

## Experience as Associate Engineer of Data science at Altimetrik Corp

February 2021 - December 2021

- Paypal Client Document Management System:
  - Image Quality Analysis as a preliminary filtering system.
    - Researched various industry standards for analyzing the quality of images, mainly no reference IQA techniques like PNR, Spatial Distortion, Spectral distortion , and structural integrity for image sharpness and blur detection
    - Also developed a function that measures the variance and means of the brightness values channel in the HSV format of the given image.
  - Named Entity Recognition for data extraction:
    - Used python library faker to create functions that can generate dummy data points mimicking passports and Driver's Licenses.
    - Generated 2k records and trained custom NER BERT models with libraries like spaCy, FLAIR , and Deeppavlove.
  - Document classification Prediction endpoint:
    - Developed a python flask app that downloads a pre-trained model and uses it to predict document type with an exposed endpoint.
    - Added functionality to use PP-OCR for text extraction, sklearn F1 metric for model performance evaluation.
- January 2022 - June 2022
  - Internal Project Development: Digital Lending Platform, Assisted in end-to-end MLOps projects, and gained experience in CICD pipeline.
  - Document Augmentation System: Created an MLOps application for document redaction and augmentation. The use-case focused on allowing the user to upload an image through a React.js front end to mark the regions of interest on it.

The data is passed to a python back-end run with the Flask framework. The image along with the origin of the ROI, which is tagged using text is received through a POST request. The back-end leverages the Faker library and OpenCV to draw new Text onto the ROI augmenting the original.

## PROJECTS

### Pill-em-all:

Dataset: 53,767 unique reviews from multiple users of 2638 medicines for 709 conditions.

- A medication review app developed in Python using flask for web app development. The system is built on the dataset available on Kaggle, a set of user reviews for medications with their symptoms with conditions the medicine is usually prescribed for.
- The search engine was implemented using an inverted index, giving an average retrieval time of 0.05 seconds.
- The classifier used is based on the Naive Bayes algorithm with hyperparameter tuning of the smoothing factor, achieved on an average of 60% accuracy across all classes.
- A content-based recommender system was implemented based on the given user ratings. Based on a user's search result, they are recommended medicines within the same category (a condition associated with) that have greater ratings from other users.

### Blink as a security system:

Dataset: A minimum of 15 blinks at two different rates for each user is collected using OpenCV's face landmark detection function.

- A new approach for user authentication through blinks that involves creating two SVM classifiers of two unique blink rates for each user which are used to create a password.
- Each new user is asked to create a user ID and then blink a minimum of 15 times for each rate of blink they want to set.
- The blinks are detected using a facial landmark detection algorithm to determine the Eye Aspect Ratio (EAR), which is the ratio of the width of the eyes based on horizontal landmarks to the height from the vertical landmarks, at every frame.
- The SVM classifiers take the Eye Aspect Ratio (EAR) in a window of 15 frames and use the existing labels to learn the weights for detecting a unique blink rate.
- The average precision of the SVM classifiers for three unique users was 91.0367%. The password can have either of the blinks in any combination and sequence, we use a hash table to map the sequence of blinks to characters depending on the length of the whole password ensuring the encryption of the blink password while storing in the database.