Nikhil Kashyap Shankar

Machine Learning Engineer

+91 9731029490

https://github.com/NikhilSKashyap

s.nikhilkashyap@gmail.com

PROFILE

Machine Learning Engineer with two years of experience in an AI consultancy startup. Expertise includes building various statistical models like the random forest, lightGBM, ARIMA, KNN, K-means, and building deep learning models like RNN, CNN, LSTMs, GANs, Reinforcement Learning. Skilled in Machine Learning, Statistics, Problem Solving, and Programming.

EDUCATION

NIE Institute of Technology, Computer Science & Engineering — Bachelor of Engineering (2018)

EXPERIENCE

Freelance Data Scientist, Mysuru: Feb 2020 - Present

Presented free webinars in diverse topics under Machine Learning. Consulted for a company on customer churn prediction. Designed Data Science curriculum, '11 projects to Data Science', for beginners on Github. (https://github.com/NikhilSKashyap/11-Projects-to-DataScience)

Machine Learning Engineer-2, Mysuru Consulting Group, Mysuru: Feb 2018–Jan 2020

Donned the hat of various roles, including Data Scientist, Machine Learning Engineer, DevOps Engineer, Technical Architect, Project Lead, Frontend Developer, during multiple projects. Built customized statistical and deep learning models for the clients. Architect and documented the process and code through UML diagrams.

Data Analytics for Managerial Applications in R, Remote Internship: Dec 2017

Interned under Prof.Sameer Mathur, IIM Lucknow. Solved 3 Harvard Business Case Studies during the internship.

SKILLS

Programming Languages	Python, C, C++
Data Storage Platforms and Database Management Systems	MySQL, Postgres, MongoDB
Big Data tools	PySpark, Kafka
Editors and Notebooks & Visualisation tools	Vim, Emacs, Atom, VS Code, Jupyter Notebook
Cloud Platforms	GCP, AWS
Resource Management tools	Docker, Kubernetes
Machine Learning and Deep Learning Frameworks	Scikit Learn, Tensorflow, Keras, Pytorch
CI/CD	Git, CircleCI
Web Deployment and APIs	Flask, Django, Tensorflow serving

INTERESTS

Bayesian Machine Learning, KBAI, Graph Neural Networks, Graph Database - Neo4j, Probabilistic Graphical Models

PROJECTS

Smart ETF portfolio management:

Built an unsupervised time series model for stock prediction for the S&P 500 ETF. Used a Deep Q learning technique to adjust the weights of bags of stocks, which yielded 18% growth in a year. Built API endpoints around the application for the frontend to access. Implemented redundant three-tier architecture and hosted the model on GCP.

Tech Stack: Tensorflow, Flask, GCP

Price Recommendation Engine for a Telco company:

Data Engineered the given big data, which had duplicates and simmered down by 5x times using various database normal forms. Converted their R codebase to production-ready Python code, built a recommendation engine using LightGBM, and wrote regression tests for the same. Hosted the software in their cloud base.

Tech Stack: Python, R, Scikit learn

Intelligent Process Automation for a Semiconductor Company:

Automated a process of reading the various values from a digital copy of semiconductor design architecture and populating the values in excel. Built intelligence for the testing process, which was taking two weeks per design manually. Initially built a simple UI in the Django web framework. Used docker to build OS-level virtualization to deliver software packages in containers.

Tech stack: OpenCV, Tesseract, Dask, Numba, Pandas, Numpy, Django, Docker

Demand Forecasting for an Incense Manufacturing Company:

Collected, cleaned, analyzed, and interpreted an extensive unstructured data of various products of the company. During this, we were able to come up with different insights from the data through visualization. Built various models for forecasting, including simple linear regression, ARIMA, and CNN.

Tech stack: Plotly, Matplotlib base map, Scikit learn, pyramid-ARIMA, Tensor-flow

Churn Prediction for an HR company:

Generated data based on the schema provided and built XgBoost and WTTE-RNN model for churn prediction. Wrote connectors for Amazon Redshift to train the model. Hosted the model in AWS.

Tech stack: Scikit learn, Tensorflow, PostgreSQL, AWS, AirTable

Face Recognition software for an EduTech company:

Built the MTCNN model for Face Recognition for a use case. Integrated and deployed the model with their already existing platform for hassle-free functioning.

Tech Stack: Keras, Flask

Peak detection for an IoT company:

The use case was to monitor milk storage units by collecting various data from IoT sensors. During data analysis, we found out that during the peaks in the temperature data, the milk was being filled. Wrote a peak detection script to monitor the storage unit.

Tech Stack: Plotly, Scipy

Address Matching for a Logistics company:

Built a Deep LSTM Siamese network for text similarity. Used pre-trained word embeddings to identify semantic similarities. Used Levenshtein distance to calculate string distance.

Tech Stack: Numpy, Tensorflow, Gensim, NLTK, Scikit Learn

Label Extraction of Invoices:

Used Google Cloud Vision API for extracting the data from the invoice pdf. Retrieved only the required data using regular expressions.

Tech Stack: Google Cloud Vision API, Regex

PRODUCTS

Vishwakarma, a visualization engine for researchers:

Vishwakarma is an open-source pip installable package for visualizing high quality, journal standard images for Probability Density Function, Probability Mass Function, and Probabilistic Graphical Models. Architected and spearheaded the whole server-side development process. The output can be downloaded in LaTeX, PDF, or PNG format. https://pypi.org/project/vishwakarma/

Tech Stack: Python, LaTeX, Pgmpy, Flask

WEBINARS

Machine Learning for Beginners - NYC Taxi Fare Prediction using Linear Regression, Random Forest, LightGBM - Live Coding:

Presented a free webinar in association with Nowalabs (https://nowalabs.com/) on August 15th, 2020. New York City Taxi Fare Prediction is one of the popular beginner-level problems on Kaggle. The goal of this challenge is to predict the fare of a taxi trip given information about the pickup and drop off locations, the pickup date time, and the number of passengers traveling. The webinar included -

- Data Cleaning

  Used Pandas to read and describe the data.
- EDA

  Visualized the target variable, fare amount, using Matplotlib. Plotted pickup and dropoff locations on the NYC map using Folium.
- Feature Engineering

  Implemented haversine distance to calculate the distance between pickup and dropoff points. Visualized the 'distance' using seaborn.
- Model Building

  Built a Linear Regression model, Random Forest Regressor, and LightGBM model. Compared their performances using the RMSE metric. Explained ID3(Iterative Dichotomiser 3) algorithm taking an example and showing the audience how a decision tree is built. Explained the types of ensemble learning.

You can watch the full video on YouTube here - https://www.youtube.com/watch?v=ha5TjeTXwr8.

Rock-Paper-Scissors-Spock-Lizard game using OpenCV and MobileNetV2 with Instructor led live coding:

"*Scissors cuts paper, paper covers rock, rock crushes lizard, lizard poisons Spock, Spock smashes scissors, scissors decapitates lizard, lizard eats paper, paper disproves Spock, Spock vaporizes rock, and as it always has, rock crushes scissors.*"

Presented a free webinar in association with Nowalabs (https://nowalabs.com/) on September 5th, 2020. Rock-Paper-Scissors is a hand game usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. Built RPS game from scratch using OpenCV and MobileNetV2 live under 90 minutes. The webinar included -

- Data Collection

  Used OpenCV to collect the hand gesture images of rock, paper and scissors.
- Model building

  Using MobileNetV2 built a CNN model to train on the collected images to learn the gestures.
- Integration

  Wrote integration functions to build a complete game to play the RPS game with the machine.

You can watch the full video on YouTube here - https://www.youtube.com/watch?v=oPu7RAR9VWw.

Build Neural Networks from scratch using MNIST hand-written dataset:

Presented a free webinar in association with Nowalabs (https://nowalabs.com/) on September 26th, 2020. The webinar included -

- What is a Neural Network? What is feed forward and back propagation?
- What is the cost function and its derivation?
- What are gradient descent and activation functions?
- How to implement a Neural Network from scratch using python and numpy on MNIST hand-written dataset.

You can watch the full video on YouTube here - https://www.youtube.com/watch?v=hOeqD9jHekE.