

SNEHAL SHIRGURE

240 S TAAFFE ST 617 | Sunnyvale CA 94086 | +1 (213) 432-8962
snehalshirgure@gmail.com | [Snehal Shirgure | LinkedIn](#) | [snehalshirgure \(Snehal Shirgure\) · GitHub](#)

EDUCATION:

MASTER OF SCIENCE(MS) | UNIVERSITY OF SOUTHERN CALIFORNIA, LOS ANGELES, CA **AUG 2017 - MAY 2019**
Computer Science | Coursework: Analysis of Algorithms, Data Mining, Machine Learning, Artificial Intelligence, Web Tech, NLP.

BACHELOR OF TECHNOLOGY | NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL, INDIA **AUG 2012 – MAY 2016**
Computer Science and Engineering | Coursework: Data Structures, Compiler Design, Operating Systems, Networking and Architecture.

SKILLS:

LANGUAGES: PYTHON, C/C++, JAVA, ANGULAR JS, JAVASCRIPT, HTML/CSS, PHP, AJAX, NODEJS , SQL.

TECHNOLOGIES: Metaflow, Kubeflow, MLFlow, Apache Spark, Airflow, Docker, Kubernetes, AWS, Terraform, CI/CD, Android

WORK EXPERIENCE:

MACHINE LEARNING ENGINEER | ZILLOW | SEATTLE **AUG 2019 - JUL 2023**

- Built an internal end-to-end machine learning platform support tool - CloudNotebooks, to empower data scientists with model reproducibility and simplification of their machine learning workspaces for easy and rapid experimentation and deployment (using Python, Flask, JupyterHub, PostgreSQL, Kubernetes, Docker, Terraform, Gitlab CI/CD)
- Load-tested pod utilization in Kubernetes nodes and adjusted requested resources for Jupyter notebooks in Kubeflow and increased efficiency of data scientists for iterative batch training and deploying to production.
- Developed and deployed prototype ML models (random forest, linear regression) with a microservices-based architecture invoking middleware components and database layer, in AWS environment used for training and testing.
- Built front-end user experience for data scientists to help in their post-hoc analysis work for metrics/artifacts using Metaflow UI cards and Kubeflow server APIs.
- Tracked model performance and usage metrics using Metaflow UI over time and debugged issues and areas of improvement by surveying data science teams onboarded onto AI platform.

SOFTWARE ENGINEER INTERN | RUFUS LABS | LOS ANGELES, CA **SEPT 2018-MAR 2019**

- Developed applications on Android OS for wearable mobile devices to track worker productivity and optimize time.
- Followed Agile/Kanban software development framework and used JIRA for issue tracking.

SOFTWARE ENGINEER | SAMSUNG RESEARCH INSTITUTE | NOIDA, INDIA **JUL 2016-JUL 2017**

- Managed development issues as a member of Android for Work team by Google and KNOX - Samsung's Mobile Security solution using Java and Android. Followed Agile/Scrum software development framework.
- Synchronized device data in new models efficiently with Sync Adapters in Java/Android and resolved issues and submitted updates on PLM and Perforce for product lifecycle management as part of Samsung Cloud team.

PROJECTS:

DATA MINING **JUN 2018 – AUG 2018**

- Movie Recommendation system using Collaborative Filtering: Implemented Model Based CF using Spark MLlib with RMSE – 0.94, User Based CF with RMSE - 0.92 and Item Based CF with RMSE - 1.05 to predict movie ratings. [\[link\]](#)
- Used Apache Spark MapReduce to implement Locality Sensitive Hashing (LSH) algorithm with Jaccard Similarity and Pearson correlation to identify similar movies and measure similarity with precision of 1.0 and recall of 0.983. [\[link\]](#)
- Implemented Girvan-Newman algorithm using the Spark Framework to detect communities in the graph. [\[link\]](#)
- Streaming Data Analysis: Used Twitter API tweepy for fixed size sampling (Reservoir Sampling Algorithm). [\[link\]](#)
- Programmed DGIM Algorithm to estimate number of 0/1s in the continuous data stream and checked for error rate (<50%) [\[link\]](#)

NATURAL LANGUAGE PROCESSING **JAN 2018 – FEB 2018**

- Developed a Part of Speech tagger to tag unseen test dataset using HMM and Viterbi algorithm to train model and development dataset to adjust hyper-parameters. Improved accuracy of tagging by 10% through Laplace smoothing. [\[link\]](#)
- Classified a set of hotel reviews into given 4 categories with two Perceptron classifiers. Vanilla perceptron with accuracy 88% and with Averaged perceptron with accuracy 89%. [\[link\]](#)