

# Ayush Bisht

Phone: 919-645-8016 | Email: [abisht@ncsu.edu](mailto:abisht@ncsu.edu) | LinkedIn: <https://www.linkedin.com/in/ayush-bisht-b41130191/>  
GitHub: <https://github.com/ayush-bisht> | Address: Raleigh, North Carolina, 27606

## Education

<b>North Carolina State University</b>	May 2021
Master of Science in Computer Networking	GPA: 4/4
Coursework: DevOps, Natural Language Processing, Software Engineering, IoT Analytics, Computer & Network Security, Operating Systems Principles, IoT Application and Implementation, Design and Analysis of Algorithms, Internet Protocols, Computer Networks	
<b>Netaji Subhas Institute of Technology, University of Delhi</b>	May 2019
Bachelor of Engineering in Electronics and Communication	
Data Structure and Algorithms, Operating Systems, Software Engineering, Mathematics, Digital Signal Processing	

## Technical Skills

Programming/Scripting Languages: Python, C, C++, JavaScript, MongoDB, SQL, Bash, XML, Assembly  
Systems: XINU, Linux, ARM Cortex M0+, Arduino, Raspberry Pi, MSP430, Intel 8085  
Frameworks/Tools: Git, Ansible, Jenkins, Docker, KVM, IBM Watson, Wireshark, Postman

## Work Experience

<b>Product Engineering Intern, Chalk Digital</b>	Feb 2021 – Current
<ul style="list-style-type: none"><li>Developed image classification models for advertisement banners of different sizes including 300x250, 728x90, and more.</li><li>Utilized feature extraction and fine-tuning on pre-trained MobileNetV2 to obtain 97% test accuracy.</li><li>Generated custom detectors using TensorFlow APIs to identify objects in advertisement banners.</li><li>Build prediction models for improving click-through rates of advertisements.</li></ul>	
<b>Summer Researcher, North Carolina State University ECE Department</b>	Jun 2020 – Sep 2020
<ul style="list-style-type: none"><li>Developed a Wireless Emulator, for simulation of distributed wireless nodes used in an ariel wireless platform.</li><li>Performed network performance testing and analysis on various virtualization environments like Docker and KVM.</li></ul>	

## Projects

<b>Continuous Delivery Pipeline (JavaScript, Ansible, Jenkins)</b>	Feb 2021 – Current
<ul style="list-style-type: none"><li>Configured a build environment and build job for a Java application (iTrust) utilizing ansible playbook and Jenkins.</li><li>Implemented a test suit analysis for detecting useful test, and static analysis for detecting code smells.</li></ul>	
<b>Regression Modelling (Python, Statsmodels, Matplotlib, SciPy)</b>	Oct 2020
<ul style="list-style-type: none"><li>Developed multiple regression models to establish a relation between a dependent variable and 5 independent variables.</li><li>Ascertained goodness of fit of a simple linear regression, multivariable linear regression, and higher-order regression models.</li><li>Determined statistical significance of models by calculating p-values, <math>R^2</math> values and F statistic.</li><li>Conducted residual analysis by generating Q-Q plot, histogram, scatter plot and Chi-Squared test of residuals.</li></ul>	
<b>Sentiment analysis and Response type classification (Python, Genism, NLTK, TensorFlow)</b>	Aug 2020 – Sep 2020
<ul style="list-style-type: none"><li>Implemented two baseline models using TF-IDF and GloVe word embeddings for sentiment analysis of movie reviews.</li><li>Improved F1 score by 5% by utilizing Universal Sentence Encoder to apply sentence embeddings.</li><li>Extracted sentence embeddings &amp; Part-of-speech tags from question-response pairs to create a baseline classification model.</li><li>Combined baseline features with word sense disambiguation and sentiment analysis to achieve an F1 score of 0.77.</li></ul>	
<b>XINU scheduler and demand paging (C, Assembly)</b>	Jan 2020 – Apr 2020
<ul style="list-style-type: none"><li>Integrated Random Scheduler and Linux-like Scheduler for XINU operating system.</li><li>Implemented a reader/writer lock system and a priority inheritance mechanism to prevent priority inversion problem.</li><li>Executed demand paging, its associated system calls and supporting infrastructure.</li></ul>	
<b>Social Analytics for Game Day (Python, Scikit-Learn, NumPy, Pandas)</b>	Jan 2020 – Apr 2020
<ul style="list-style-type: none"><li>Extracted audio features including A-weighted coefficients, Mel-frequency cepstrum from basketball games.</li><li>Performed crowd noise classification to identify instances of crowd cheers and boos for automatic highlight generation.</li><li>Utilizing random grid-search hyperparameter tuning for Random Forest classifier achieved 93% training accuracy.</li><li>Built a Support Vector Machine classifier to classify accelerometer data to gain insights about audience's behavior and emotional response at a sporting event and attained 98% test accuracy.</li></ul>	
<b>Communication Systems (Python)</b>	Oct 2019 – Nov 2019
<ul style="list-style-type: none"><li>Created a multi-threaded peer-to-peer system with distributed index to accomplish asynchronous request-response communication between peers using Transmission Control Protocol.</li><li>Created a point-to-multipoint system to achieve synchronous data transfer between systems using User Data Protocol.</li><li>Implemented stop-and-wait Automatic repeat request protocol to achieve reliable data transmission.</li></ul>	