Technical Solution

This section addresses solution for problems discussed in previous section

System overview

Data polish is a web based application primarily for Master’s and PhD student to clean datasets before performing any machine learning or deep learning techniques on their dataset using simple user interface.

Data import

In this page end use will be able to import their dataset for cleaning

It will be saved in cloud and they will get unique ID to track the progress.

A computer screen shot of a cloud

Description automatically generated

Data Profile

In this page end user will the data quality index and full profile of the dataset

A blue background with a square with black text

Description automatically generated with medium confidence

A screen shot of a graph

Description automatically generated

Data cleaning

In this page end user will be able to click the button and data cleaning will be

A blue and white screen

Description automatically generated

Export Data

In this page user will be able to download the cleaned data file

A computer screen shot of a computer screen

Description automatically generated

Basic Architecture

A diagram of a computer program

Description automatically generated

We're developing a modern web application using a stack that includes front-end technologies like Angular, Bootstrap, and Prime NG(framework) for the user interface and on the backend we're utilizing Java, Spring Boot, and Maven to build our API. Our data is stored in a MySQL database, and for file storage, we use Azure Storage Account's blob storage. For data quality and profiling, we've integrated Python into our application.

User interact with application with Angular frontend application. Angular app send HTTP Request to spring boot API which can query database for any required information, and put message into queue stating what it needs and python application takes the message from there and it can also store data files in blob storage.

Cloud Deployment

Once base architecture was fixed the basic application is also developed to be hosted. While hosting we faced several issues and we have to change the architecture multiple times to ensure robust integrations between services.

In order to keep up with learning curve, need for complex deployments, need of multiple environment to test code and limited cloud resources we decided to use AWS EC2 – Ubuntu server running tomcat server for spring boot application and angular applications and other python services are deployed as daemon service in the same server.

For DB we use AWS RDS service with MySQL engine.

For messaging system, we use Microsoft Azure service bus and for storing binary files we are using Microsoft Azure Storage Account Blob Storage

A diagram of a computer

Description automatically generated

Flow of our application

1. Data Upload

A diagram of a software development

Description automatically generated

This is the starting point for the user to our application. This flows starts by uploading dataset by end user in the angular application. From the angular app file is sent to spring boot API where file is saved in blob storage and name of the file will be stored in MySQL DB.

When file name is inserted into DB unique GUID is generated and returned to API. Unique GUID is Job ID and published to azure service bus in the form of message and same ID is returned to user.

Python app running on backend will pick up the message from the queue and start processing it once data profile is done it save the output back into blob storage and puts URI in DB.

Data ProfileA diagram of a software development process

Description automatically generated

All the necessary operations for data profiles will be finished when the user uploaded the data. So when user wants to see the results of data profile they can just give the job id they have into frontend. It will be passed to API from there DB will be queried with the job id and gets back the URI of data profile output

Once the API gets URI it will fetch file from the blob storage and return it front end application where the data will be transformed into visualization.

Data Cleaning

A diagram of a software application

Description automatically generated

When user is satisfied with the data profile they can start cleaning data. They can set the rules from UI for data cleaning row by row, Angular collects all the rules and sends in to spring boot API from there the rules are stored in DB and set of rules and job id is published as message to queue.

The python data cleaning service takes the message and starts working on it get the file from blob and cleans and saves it back finally it save the result data set URI in DB.

Export DataA diagram of a software development process

Description automatically generated

When user request for clean dataset with their JOB ID, angular makes the HTTP request to spring boot application with job id. API will get url from DB and fetches files from blobs and sends to UI where user can download it.

**GIT Flow**

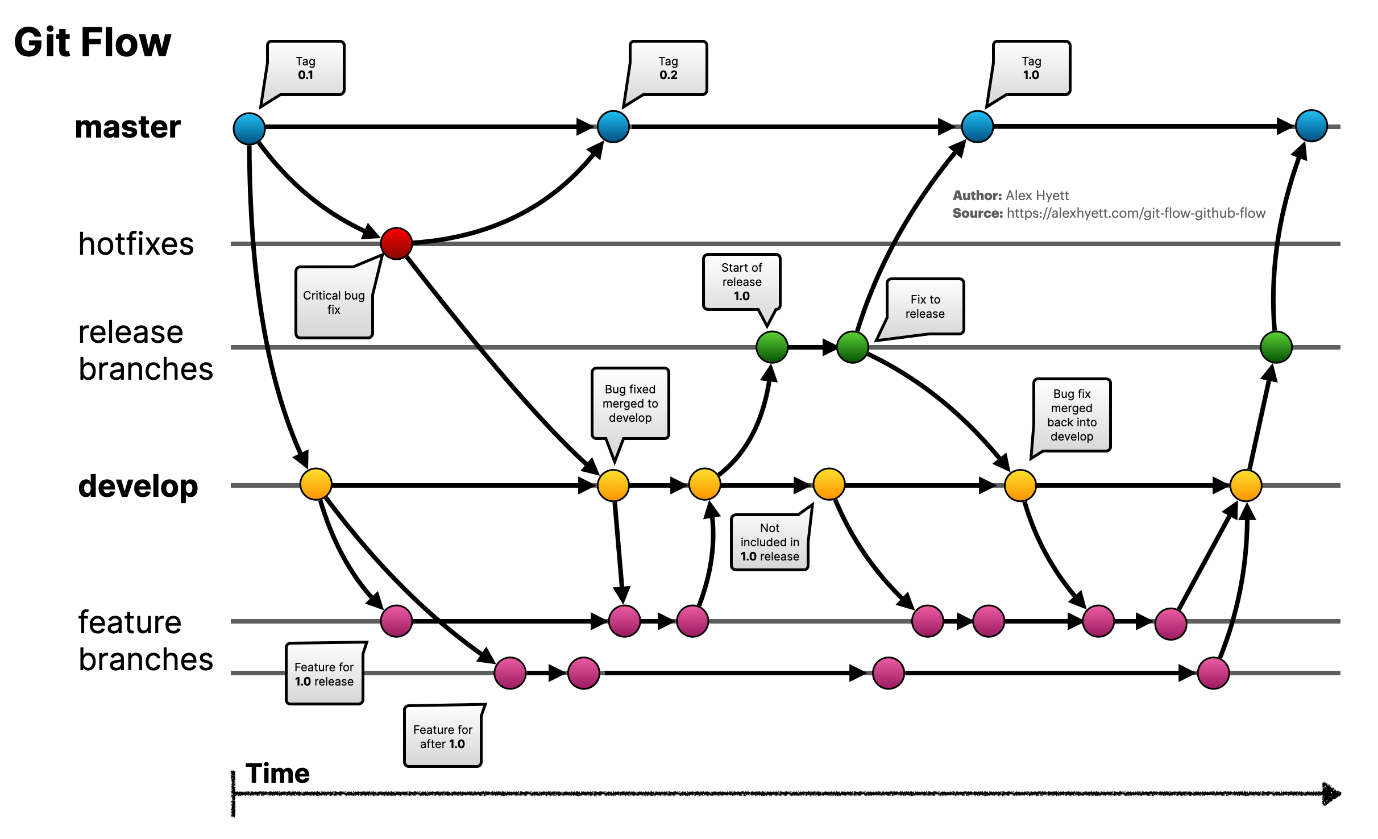
We use version control system to save, keep track of our teams progress and preventing from losing the changes

Version control system we will using for this project is GIThub

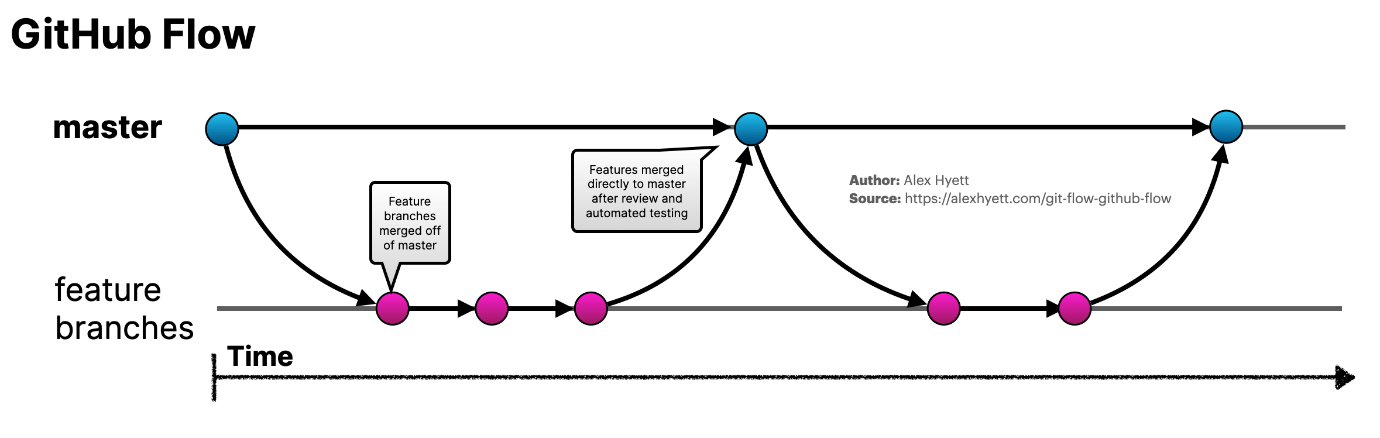
There are multiple flows we can follow to merge our changes into production but the one we choose is GIT flow

We planned to do release after each sprint and we needed multiple branches like development and QA to merger changes and test for each sprint which is not the case with github flow where we use only two branches to trach out changes that’s why we choose gitflow

(Hyett, 2022) &  (Phillips et al., 2011)



(Hyett, 2022)



(Hyett, 2022)

We use gitflow with some tweeks

A diagram of a diagram

Description automatically generated

Feature Branches will be used by developers to push their code every story will have their own feature branch.

Once the developer is happy with feature branch he can raise the pull request to team members in order to merge into develop branch

Our team set the criteria that we need to have minimum of 2 approvals in order to merge

Once all the stories are closed for that sprint all the changes will be moved to QA where it will be deployed and tested

We are using regressing testing so far but planned to do automation.

If branch has been signed off to merge then it will be merged into production branch where I will be deployed and quick testing will be done.

And after the sprint over with production code deployed it will be moved to main branch for backup just in case if anything goes wrong.

This process is and will be followed thoughout the project.

Phillips, S., Sillito, J., & Walker, R. (2011). Branching and merging. *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*. <https://doi.org/10.1145/1984642.1984645>

Hyett, A. (2022, November 10). *Git Flow vs GitHub Flow*. Alex Hyett. https://www.alexhyett.com/git-flow-github-flow/

**CI/CD**