Practicum Project Design & Implementation Plan

Joseph Miano
jmiano@gatech.edu

1 DESIGN

1.1 Project Summary

We developed machine learning models to predict hospital readmissions in diabetes patients and created an interactive dashboard hosted in the cloud. The results of this project include data visualizations from our exploratory data analyses and machine learning models, a trained machine learning model (logistic regression) that can be used to predict hospital readmissions for new patients, and a consumable dashboard deployed on the cloud. The dashboard enables users to visualize the data, model results, and get results for new input patient data via an interactive interface.

1.2 Tools and Technology

We leveraged the following tools and technologies:

- **Python 3 programming language** with the following additional libraries:
  - **Data Processing**: NumPy, Pandas
  - **Machine Learning**: PyTorch, Scikit-Learn
  - **Deployment**: Dash, Flask
  - **Data Visualization**: Matplotlib, Seaborn, Plotly

  The Python programming language enables fast iteration, efficient syntax, and access to robust data science and visualization tools.
- **Amazon Web Services**: we deployed our machine learning models to the cloud by using an AWS EC2 instance for computation and storage. This enables our model and visualizations to be consumed by users.
- **Programming Environment**: we leveraged PyCharm and Jupyter Notebooks to develop our code and train our machine learning models.

1.3 Data Sources

Our main data source for this project is the dataset from Strack et al. (Strack et al., 2014), which contains 55 features and 100,000 instances of patient-level data related to hospital readmissions in diabetes patients. Each record in the dataset represents a patient with diabetes and contains 55 features related to that patient,

including whether the patient was readmitted in less than 30 days, more than 30 days, or not readmitted to the hospital. We obtained this dataset from the UCI Machine Learning Repository (UCI, 2014), which is a repository containing many different datasets amenable to exploratory data analyses and the training of machine learning models. This dataset was used for our exploratory data analysis, machine learning model training, and leveraged in our deployed EC2 instance for data and model visualizations.

1.4 Diagrams

Figure 1 shows the architecture diagram for our deployed dashboard and model. The user connects to the dashboard that is hosted in an AWS EC2 instance. The AWS EC2 instance also hosts the machine learning model, which interacts with the dashboard to display results interactively on new user inputs. Furthermore, the static data for the initial machine learning model outputs and tables for the dashboard visualization are also hosted in the EC2 instance.

Figure 2 shows the process flow diagram for training and deploying our machine learning model. After obtaining the data, we preprocessed it, which involves data cleaning and feature engineering. Then we performed exploratory data analysis and visualization, model training, and evaluation, and finally deployed the model in production (on an AWS EC2 instance in this case, as shown in Figure 1). The arrow from machine learning model evaluation to data preprocessing forms a cycle, since model evaluation does inform how to preprocess the data to improve performance (e.g., via engineering better features).

1.5 Screen Mock-ups

Figure 3 shows a screenshot of our final dashboard. At the top, we have the dashboard title and description. Just below the dashboard description are two buttons: "Use Model" and "Download Outputs". The "Use Model" button prompts the user to upload a CSV file containing records with features that the model (logistic regression) needs to output predictions. Once the user uploads the data they want predictions for, the "Download Outputs" button can be used to download all the model prediction records. Furthermore, the "Model Prediction Distribution" plot populates from the user data. To the left of the user-input-data model predictions is an interactive plot showing the distribution of values for each column in the original overall dataset. At the bottom, we have 3 visualizations based on the original data that the model was trained and evaluated on.

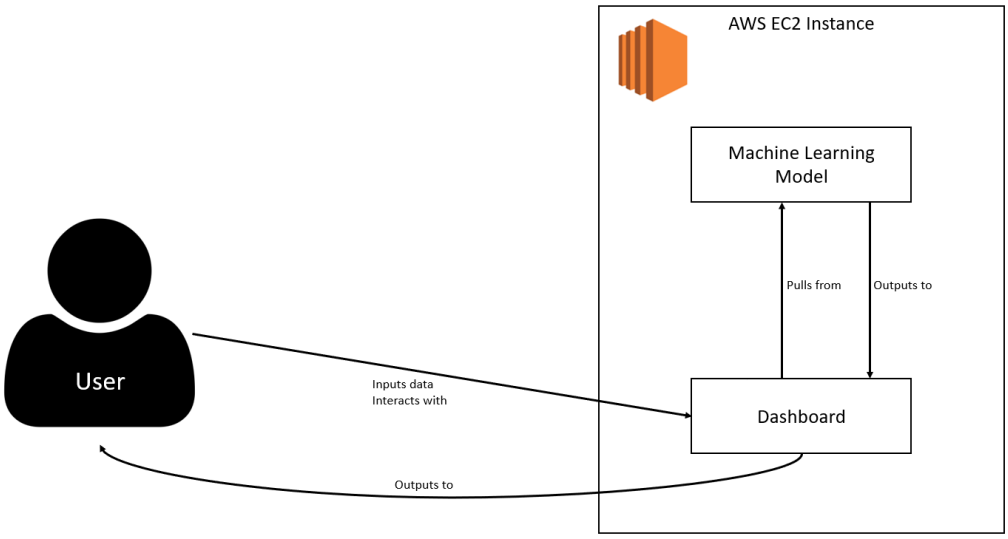

Figure 1—Software Architecture diagram for the deployed machine learning model and dashboard.

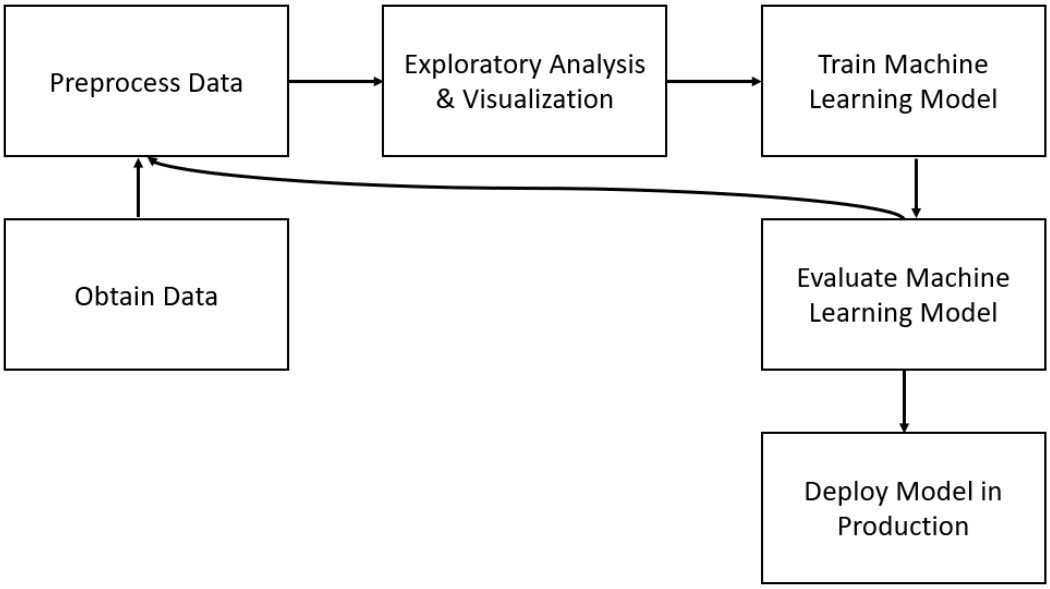

Figure 2—Process flow diagram showing the stages of development for our machine learning model.

The "Plot of data in PCA space" shows a 3-dimensional representation of the input features used to train the model. The "Performance Metrics" plot shows the performance of various models on our validation dataset, including a random forest classifier, adaboost classifier, logistic regression classifier, neural network,

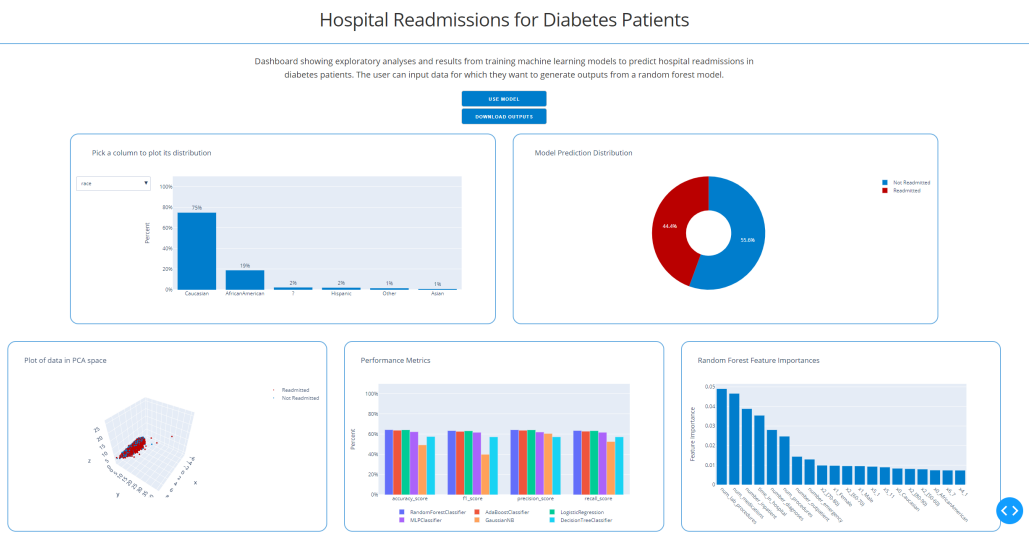

Figure 3—Screenshot of the final dashboard.

naive bayes classifier, and a decision tree classifier. The "Feature Importances" displays a sorted bar plot of the features most important in our final random forest model.

2 IMPLEMENTATION

2.1 Project Tasks

- **Week 11**: downloaded dataset and set up programming environment. Ensured all necessary programs and libraries (as outlined in the tools and technology section) were installed and functioning correctly.
- **Week 12**: performed exploratory data analyses and static visualizations.
- **Week 13**: trained machine learning models, determind feature importances, and produced relevant visualizations.
- **Week 14**: finalized machine learning models and set up AWS EC2 environment for cloud hosting. Began development of dashboard for data visualization and user-model interaction.
- **Week 15**: finalized model and data visualization deployment as well as the interactive interface for users to get outputs from the model.
- **Week 16**: wrote-up the project results.

2.2 Project Timeline

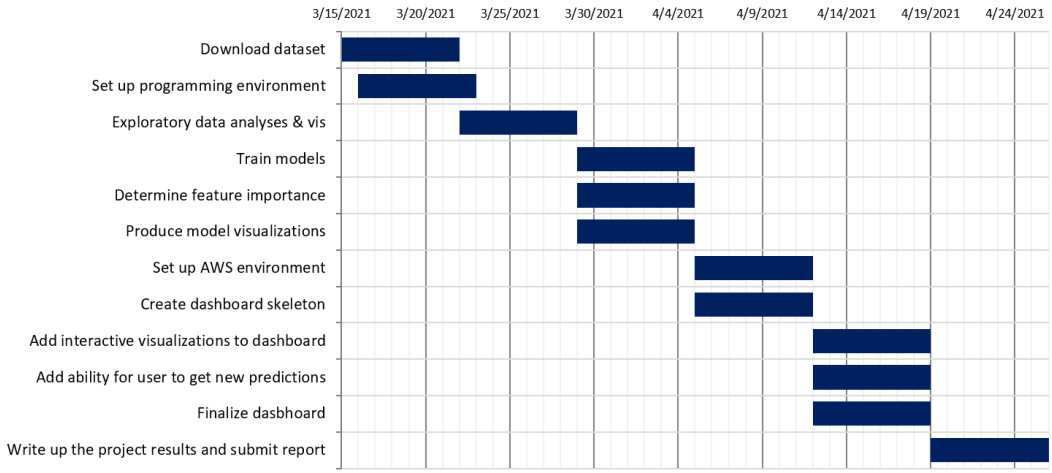

Figure 4—Practicum Project Gantt Chart.

2.3 Needs / Risks

In order to implement this project, we needed to leverage the tools and technologies outlined in the Tools and Technology section. Furthermore, we needed a stable internet connection and a computer with enough processing power to iteratively train and test machine learning models, which we did have access to. The deployment phase of our project involved using an AWS EC2 instance and

setting up an interactive dashboard. Thus, we needed an AWS account for this portion of the project and relied on the infrastructure provided by AWS.

Although we did not anticipate any risks that would prevent us from completing this project, some potential risks included:

- **Technological**: there were technological risks like an AWS server issue or potential damage or malfunction of the computer we were using to perform the project tasks. Although technological risk did not prevent us from completing the project, mitigation strategies involved creating frequent backups of project work and leaving enough time to resolve issues if they did appear.
- **Cost**: it was important to manage and monitor the costs associated with deploying and maintaining our model and dashboard in production. These include the storage costs associated with the AWS S3 bucket and the costs associated with the EC2 instance hosting the model and dashboard.

3 REFERENCES

[1] Strack, Beata, DeShazo, Jonathan P, Gennings, Chris, Olmo, Juan L, Ventura, Sebastian, Cios, Krzysztof J, and Clore, John N (2014). "Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records". In: *BioMed research international* 2014.

[2] UCI (2014). *Diabetes 130-US hospitals for years 1999-2008 Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/diabetes%20130-us%20hospitals%20for%20years%201999-2008.