BYU MARRIOTT
SCHOOL OF BUSINESS

INTEX Winter 2022

Utah Motor Vehicle Collision Application

Summary

Utah has a unique set of challenges when it comes to motor vehicle accidents. For example, although alcohol is still a problem, it may be less of a problem in Utah than distracted driving. Lawmakers need to understand these unique challenges better in order to regulate effectively. A dataset has been collected on motor vehicle collisions during three years: 2016, 2017, and 2019. This dataset does not include every collision from each year and, clearly, 2018 is missing. However, we will work with what we have.

Objective

Store this data in a searchable database and create a web application that will allow two types of users: database readers and database writers. Readers should be able to view all of the data including summaries and predictions. Writers should have all privileges of the readers including the ability to add/view/update/delete records.

The data should be searchable and include a variety of summaries and predictions that you believe will help regulators make better decisions. At a minimum, you should generate a model to predict crash severity–an ordinal variable scored from 1=no injury, 2=possible injury, 3=suspected minor injury, 4=suspected serious injury, and 5=fatal. This model should be trained and made available for various data features that you will creatively determine and add to your application. However, the model training including all data cleaning and import should appear in a single .ipynb file that facilitates TA grading. You do not need to automate model retraining for the purposes of INTEX. However, the ways in which you use that trained model in your web app is up to you. Be creative. You will be graded on how useful the evaluators believe your creative data features will be. For example, the dataset includes a feature indicating whether distraction was involved in the accident. You could add a feature that allows users to assume they are distracted and see a predicted level of crash severity for themselves based on where they live.

Requirements

The specific requirements are detailed below.

Web Application Requirements (IS 413)

Build a website using ASP.NET to present the findings of the data analyses that you will perform. The specific number of pages on the site and how you present that information will be up to you, but the website should be well-organized, professional, and modern. Remember that the presentation of the information has a large effect on how the information is perceived. This site should be something you will be proud to include in your portfolios.

Build a database from the provided data set. You may use whichever DBMS you prefer, except SQLite. As always, verify that the data set follows the proper rules of normalization.

Requirements

Please be sure to include at a minimum:

- A landing page that explains to the user what the site is and what it does
- Clear navigation
- A page that lists out the summary data per accident for all accidents in the database. (There are a significant number of records, so probably best to break the summary information up by pages.)
    - The ability to filter the data in the summary page to get to specific results
- A page that allows administrators on the site to add/update/delete crash information
    - Proper validation of the information being entered into the database
- At least one in-context usage of the word "Yeet" on the site
- All the finishing touches that make the difference, like changing the default icon for the site, using graphics/pictures/colors, changing the title for the tab on each page to be something relevant, using dropdowns where possible for entry, etc. that enhance the user experience on the site
- Good, clean, readable code (good commenting, indentation, line spacing, etc.)

One option you may consider in your analysis is to use something like this example to build a ML Model that utilizes an .onnx file in ASP.NET:

https://towardsdatascience.com/deploy-sci-kit-learn-models-in-net-core-applications-90e24e572f64

This example uses the ASP.NET API template, but could be easily adapted to instead pull the input information from a form to pass to the HTTP post on the controller. (This is the piece that Prof. Keith and I have tested together, and it worked great for us.)

Machine Learning (IS 455)

Requirements

- **Training a Model**
  - Generate a .ipynb file that retrieves the dataset described here: https://opendata.utah.gov/Transportation/Utah-Crash-Data-2020/herb-zqda. You'll find a link there to an API for the dataset. Although you will store a copy of this data for your web app's database, use the API for model training purposes.
  - See the detailed data dictionary found on that website; it will not be repeated here
  - Train a model to predict crash severity. Recall that crash severity is defined on page 1 of this document. Document your data cleaning, feature selection, and algorithm selection process so that TAs can review your .ipynb file and easily understand your work.
  - You may train other models as well if you have an idea you want to implement. However, this is not a requirement
- **Deploy a Model**
  - Save this trained model as a .onnx file using the demo provided in the web app description above.
  - Creatively determine data features to add to your web app that will use the trained model stored in the .onnx file. Consider–what will help regulators make informed decisions? What would you want to know if you were a regulator? This could be a prediction that is displayed when a user searches for crashes and then clicks on the details of a particular crash. It could be a basic calculator that allows users to enter details of a hypothetical crash and then get a predicted crash severity. Consider that basic calculator as a "last resort" option. If you can't think of any useful data features, at least create the calculator which demonstrates that you know how to integrate a trained model into your web app.
- **Not required**
  - You do not need to set up your environment to automatically retrain your model. I simply need to export a model using the ONNX library, add that model to your web app, and then use it in creative ways to add value to your project.
  - You do not need to allow the ability to retrain the model after new records are added to the database.
  - NOTE: both of these features above are essential parts of a machine learning environment. But they are not required in order to keep this project from getting overly complex

Security, Authentication, and Authorization (IS 414)

The IS414 portion of INTEX includes BOTH applying skills you have already learned AND learning how to implement a few new skills now that you have sufficient technical background to complete them.

Requirements

- □ Encryption
    - □ Use HTTPS for all public connections. It is fine if you use a subdomain, but you must have a valid certificate to enable TLS.
    - □ If you are connecting to a separate server/service, do so securely (e.g., connecting EC2 to RDS over TLS).
    - □ Enable HTTP Secure Transport Security (HSTS) on your site.
    - □ Redirect HTTP traffic to HTTPS.
- □ Authentication
    - □ Provide a method to authenticate users using a username/password.
    - □ Configure ASP.NET Identity PasswordOptions to implement requirements that require better passwords than the default settings (see https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity-configuration?view=aspnetcore-3.1). You should know what “better” means.
    - □ Enable at least one form of two-factor or multi-factor authentication. You have flexibility in the choice and implementation. Note: You may need to have some accounts without 2FA or MFA for grading purposes so that the TAs can access your Website for grading without having your phones or faces.
- □ Authorization
    - □ Use Role-Based Access Control (RBAC) in a manner that only an authenticated administrator has the ability to add, modify, or in rare cases delete data. You may choose to have additional roles beyond an authenticated user and non-authenticated user, but this is not required.
- □ Integrity
    - □ Data should only be able to be changed or deleted by an authenticated administrator and there should be confirmation required to delete data.
- □ Credentials
    - □ Handle credentials (usernames and passwords, API keys, etc.) safely. You may choose to use a secrets manager, a separate file that is not uploaded to a code repository like Github, or an environmental variable. But you should not include credentials in your code. If you choose to use a separate file, you may need to encrypt the contents (e.g., see https://docs.microsoft.com/en-us/aspnet/web-forms/overview/data-access/advanced-data-access-scenarios/protecting-connection-strings-and-other-configuration-information-cs for encrypting the web.config file for connection strings). Please

note that having a functional Website is worth more overall points than protecting credentials properly, so prioritize wisely (as much as this pains your instructor).

- □ Privacy
  - □ Create and populate the content of a GDPR-compliant privacy policy that is linked from the footer of your Web App (at a minimum on the home page). You may use existing templates (see https://gdpr.eu/privacy-notice/) or or generators (see https://www.privacypolicies.com/privacy-policy-generator/ or https://termly.io/resources/templates/gdpr-privacy-policy/), but the content should be tailored to your site.
  - □ Enable a GDPR-compliant cookie consent notification
- □ Hardening
  - □ Close/remove all unnecessary ports and services. We're not sure if BYU or AWS will allow TAs to scan INTEX project instances, but your instances shouldn't have services running and ports open that are not required. You certainly want to have SSH, HTTP, and HTTPS ports (22, 80, 443) open. Under some circumstances, you may need to have a database port open.
- □ Other
  - □ Enable the Content-Security-Policy (CSP) HTTP header. Specify the sources you need for your site to function and no more (e.g., you might choose to define default-src, style-src, img-src, script-src, etc., but only choose what you need).

Not required and no points added, but if you have extra time …

- □ At least one in-context usage of "root beer" or "rootkit" on the site
- □ Encrypting the words "root beer" or "rootkit" using your algorithm of choice (or hashing with your algorithm of choice) and setting a cookie with that content on every client's device
- □ Storing the data on a private blockchain (please don't actually do this during INTEX week)
- □ Setting up the AWS Network Firewall
- □ Setting up AWS Inspector
- □ Setting up DDoS Mitigation (e.g., AWS Shield)
- □ Setting up a regular database backup

Infrastructure and Deployment (IS 404)

The important IS 404 concepts come into play in the deployment of your application. Remember the things we have discussed related to why people use cloud computing, and why AWS is able to charge money for the services they provide (like EC2, RDS, etc.). Companies use AWS for the value they provide, and you should be using the appropriate AWS services and settings for the product you are creating.

We highly recommend you use a personal AWS account to do your deployment. The limitations of the AWS Academy lab environment will cause you unnecessary strife during your INTEX deployments. The AWS Free Tier (for new accounts) includes some level of free EC2, RDS, S3, data transfer, and many other things. If you are judicious in your usage, you should be able to complete Intex with minimal-to-no costs. Once INTEX is over and grading is completed, you should stop or delete your services to avoid continued charges.

Requirements

- Deploy your application to AWS, including a database.
- Have a custom domain. You can either purchase your own (recommended) or use a subdomain of is404.net. The Beanstalk-provided URL does not count as custom.
- You do not have to use AWS services exclusively, but the primary computing and database need to be there.
- Your website should be accessible through HTTPS.
- Select the appropriate AWS resources for each service needed to complete the requirements of the project. Document any choices and services used in the deliverable.

Submission

Deliverable Submission

Deliverable Due Date: **April 8, 2022 8:00 am** (the morning of the presentations).
Deliverable Submission Location: The IS 414 (Security) Learning Suite course under the INTEX 2 assignment. Only one submission per group is needed.

Written Project Description Design

You are to submit a document (e.g., PDF) containing the following:

1. **Group Information**

	Group Information
Group Number	
Group Members	

2. **URLs and User Credentials -** Include completed versions of the following tables. Grading will go VERY badly if you make typos here.

	URL	**GitHub Branch**
Website URL		
GitHub URL		
.ipynb URL		
.ipynb URL (if more than one)		

Note: Please ensure when submitting that your GitHub repository is set to Public for grading purposes. You may submit the URL to the ipynb in GitHub or Google Colab.

While you need to be very careful with credentials, please provide faculty and TAs the login credentials to your ASP.NET Core application (including users with different roles if implemented) so that they can properly grade your solution.

Username	**Password**	**Role**	**2FA/MFA Used (if any)**

Note: The TAs should be able to access the Website without needing a “something the team has” or “something the students on the team are”. You may need to create an account that doesn’t require 2FA/MFA for this purpose.

3. **Details of your solution -** While all (or almost all) of the grading will be completed by evaluating your Web application, code repository, and formal presentations, it can be useful to specify some of the details of your solution. This section of the document has no length requirements and may even be omitted if you are certain that your Web application visibly meets all project requirements (e.g., deployment to AWS). The point of this section is to, without rambling, include information to convince those grading your projects that you successfully applied your skills to meet the requirements. You might also call out any ways that you went beyond the requirements. Again, there is no specific requirement for length or formatting of this section of the document.

Peer Eval

You will also need to complete the INTEX2 Peer Eval by **Friday 04/08 at 11:59 PM**. The survey can be found here: https://byu.az1.qualtrics.com/jfe/form/SV_diq7Vw5W0dKr62G. Please complete this immediately after your presentation. The results of your eval will be kept confidential, but it will affect team member grades. We want this to be fair for those that did the work. The solution is to make sure that you pull your weight and more. Help the team and be a great team member helping where needed and doing quality work!

Presentation

You will be randomly assigned a presentation slot. Presentations will be held Friday April 8th beginning at 12:00 pm. A schedule with the assigned times for each group will be sent out shortly. Each presentation period will be structured as follows:

Presentation and Tech Demo	20 minutes (Student group presents)
Questions & Answers	5 minutes (Student group answers questions)
Judge Deliberation	5 minutes (Student group waits in the hall)
Feedback	5-10 minutes (Student group comes back to room)
Break	5 minutes

The judges for your presentation will take the role of Utah Department of Transportation (UDOT) executives and other interested stakeholders (e.g., politicians, advocacy groups, etc.). Your role is to pitch your system/solution as worthy of continued external investment and/or adoption by UDOT or the state. The judging panel will be evaluating the technical solution in addition to how well the solution helps solve the greater societal issue. The panel has some technical background, but is primarily interested in the functionality, features, usability, feasibility/cost, and impact of your solution.

Presentation Schedule

	TNRB 210	TNRB 220	TNRB 230	TNRB 260	TNRB 270	TNRB 280	TNRB 284	TNRB W242	TNRB W308	TNRB W310
12:00 - 12:40	2-1	1-4	3-14	3-15	1-12	4-5	3-13	2-12	4-7	1-13
12:45 - 1:25	4-3	2-7	2-10	4-4	1-6	4-1	3-7	3-9	3-8	3-6
1:30 - 2:10	2-9	1-8	2-6	1-3	3-5	3-1	2-11	3-3	1-15	2-14
2:15 - 2:30	**Break**									
2:35-3:15	3-2	2-2	1-2	3-4	1-14	4-2	1-11	1-10	2-3	2-5
3:20-4:00	2-4	3-11	1-5	1-9	2-15	1-1	2-8	2-13	1-7	4-8
4:05-4:45	3-10	3-12	4-6	4-9						

Slack

A Slack workspace has been set up for you to communicate with instructors, TAs, and other students (see below). Please use this link:

https://join.slack.com/t/intex2w22/shared_invite/zt-16p5ftsyp-2gKhILUjfc43qfc86Bz37g

You will be able to create your own channels for your team within this workspace.

Questions

You are not able to ask Faculty and TAs for help with technical problems, but you are welcome to ask questions in regard to the case itself or the requirements. However, due to the number of groups and students involved, please post any questions on the “general” channel in our INTEX2 Winter 2022 workspace on Slack so that everybody can benefit from the response and that the faculty are not answering the same question over and over again to different groups separately.
You may ask questions to other students working on the project (use the “intex” channel in our INTEX2 Winter 2022 Slack workspace), but you may not ask questions of former students who have completed a similar project in a past INTEX. Seeking help from previous students in any way or using any of their material would be considered cheating. You may ask questions of members of the industry or other outside help.

INTEX Results & Awards

We will make an announcement about the winning team(s) on Wednesday (4/13), 9:00-10:00am in W408/410.