Table of Contents

A1. Letter of Transmittal

August 1st, 2022

Mariam Clarke
Chief Technology Officer
Autom8 Intelligent Solutions Inc.
404 NW Ave
Newark, NJ 07101

Dear Ms. Lovendusky,

Flight delays have increased the cost of business travel incurred by companies since the inception of commercial air travel. However, the recent pandemic and pilot shortage have exacerbated the prevalence of flight delays and associated costs. Autom8 plans to leverage machine learning to help alleviate this burden for your company, Lovendusky Consulting.

As the airline industry struggles to normalize and deal with increased flight demand alongside reduced staffing, machine learning can mitigate the consequential implications imposed on your company. The web-based solution will allow users to input the details of a flight and assess the likelihood of delay when planning for and scheduling business trips. This will allow your company to proactively mitigate risk of flight delay prior to booking. Graphical visualizations of the underlying data will also allow provide insight into emerging flight travel patterns. Our preliminary calculations show that implementing this product would reduce your annual business travel budget requirements by a minimum of 25%.

The funding required to develop and produce the prediction system will be approximately $50,800. All associated costs are related to staffing and development efforts. The team taking on this initiative consists of two software engineers and a business analyst. The senior engineer leading this team has advanced knowledge in this field and a proven track record of producing quality products for our clients.

On behalf of all of us here at Autom8, we greatly appreciate Lovendusky Consulting's continued partnership. I look forward to discussing this opportunity with you and am hopeful this will be another great venture we can achieve together. Please do not hesitate to reach out to me with any questions.

Sincerely,

Mariam Clarke
CTO - Autom8 Intelligent Solutions Inc.
973-815-2424 | mariam.clarke@autom8.com

A2. Project Proposal

Problem Summary

Serving as one of the largest enterprise consulting companies in the United States, Lovendusky Consulting has experienced a significant increase in annual travel costs since 2020. Autom8's analysis has shown that an exorbitant amount of the company's travel expenditure has been the direct result of costs associated with flight delays. Resulting expenses noted include the following: additional hotel stays, additional meal costs, missing connecting flights/purchasing additional plane tickets, and inability to attend scheduled obligations with clients. The flight prediction system will give Lovendusky Consulting insight into emerging flight delay patterns. It will enable them to assess the risk of booking a particular flight in real-time. This will lead to reduction in travel costs. It will also ensure Lovendusky Consulting is better able to meet scheduled client obligations and ultimately retain their clients.

Data Product Benefit

Lovendusky Consulting does not currently have a technical tool or system capable of predicting the likelihood of flight delay. The proposed product will allow them to input details of a prospective flight before booking. Details related to flights that will be assessed by the system include airline used, airport flying from, airport flying to, day of the week travelling, length, and duration of flight. Note, users will not be required to input details such as length and duration of flight as this will be calculated automatically by the system based on the other input fields submitted. Data visualizations will also be provided with the product. These visualizations will provide insights related to the significance of flight details allowing for better planning and scheduling.

The system will analyze the input through the machine learning model and output a prediction as likely or unlikely to be delayed. This will allow Lovendusky Consulting's employees to feel more certain their flight will both depart and arrive as expected before booking. This will mitigate negative financial impact related to travel and ultimately the ability to meet client expectations. Autom8's preliminary analysis has calculated a minimum annual travel budget reduction of 25% as a result of the system implementation.

Data Product Outline

The programming language that will be used to develop the product is Python. Python is the industry standard for projects involving machine learning. This is primarily due to the wide-range of analytics and statistics open-source libraries available. The application will be web-based and will utilize a Jupyter notebook hosted in Google's Colaboratory environment. Jupyter notebooks allow for incremental coding and testing via web browser. Each section of code is grouped in cells which can be run individually as needed. Google Colaboratory offers an additional benefit to Jupyter, providing cloud compute resources for process execution. Another great feature offered in Colaboratory is the ability to receive user input through Google forms which can be embedded within the notebook. The flight delay prediction product will leverage the form feature to provide a drop-down list for each flight detail field.

Data Description

The data used to train and test the system will be sourced from a CSV file. The file will be stored in a repository directly accessible from the notebook so there will be no need for the user to store or load it locally. Backup copies will also be maintained for use should the original copy become corrupted. The initial dataset utilized to create the flight delay prediction system was obtained from Kaggle.com. The raw dataset contains approximately 540,000 records of recent flight information. Irrelevant data fields such as id and Flight will be removed from the dataset before training and testing the system. Certain fields such as Airline, AirportFrom, and AirportTo will be re-formatted such that they are compatible with the model inputs expected. Additionally, records not related to the primary US airlines or airports will be removed. This will allow the application's focus to remain relevant to the airlines and airports of interest. The fields provided in the raw dataset include the following:

- id (Data Type – Integer)
- Airline (Data Type – String)
- Flight (Data Type - Integer)
- AirportFrom (Data Type – String)
- AirportTo (Data Type – String)
- DayOfWeek (Data Type – Integer)
- Time (Data Type – Integer)
- Length (Data Type – Integer)
- Delay (Data Type – Integer)

Objective and Hypotheses

The primary objective of the flight delay prediction system is to reduce Lovendusky Consulting's annual travel budget costs by at least 25%. Providing an accuracy rate of at least 70% will be key to achieving this objective. Additionally, the product will provide meaningful visual representations of the data that support the model's prediction results. The hypothesis for this product is that the airline used will be one of the strongest determinants of flight delay. More specifically, one of the four major US airlines – American Airlines, United Airlines, Delta Airlines, or Southwest Airlines will have a stronger association with delayed flights.

Project Methodology

The solution development will follow the SEMMA methodology. This is appropriate for this project because it is designed to be highly iterative in nature. The phases of SEMMA include Sample, Explore, Modify, Model, and Assess. The SEMMA phases will be conducted for this project as follows:

1. Sample – A subset of the data will be extracted from the full data set that provides equal representation of the data to be analyzed.
2. Explore – Relationships between the data points will be examined to determine correlation and relevance to flight delay prediction outcome.
3. Modify – Variables found to be irrelevant to the goal of the model will be removed before re-training and re-testing the model.
4. Model– The processed data will then be used to build the model.
5. Assess– The model will undergo testing and performance metrics such as accuracy assessment will be applied to the results. Based on the findings, this process may be repeated as needed until the required level of accuracy is achieved.

Funding Requirements

The funding required to develop and produce the flight delay prediction system will be approximately $58,200. There will be no cost associated with development environment or licensing as all tools used will be open source. All associated costs will cover staffing and development efforts. The team taking on this initiative consists of two software engineers and a business analyst. The business analyst will assist with documentation, communication between the client and developers, and UAT planning. If out-of-scope requests are added to the project, they will require additional funding.

Solution Impact

The flight delay prediction system will offer numerous benefits to clients and stakeholders. Lovendusky Consulting will be able to input the details of a flight prior to booking and receive a response stating whether it is likely to be delayed. This will allow them to pick another flight or move the proposed meeting time if the only available flight is a delay risk. The data visualizations will also provide insight into emerging patterns regarding flight delays. Autom8 will be able to expand upon the initial data set to ensure the latest relevant flight data is accounted for. Overall, the solution will reduce costs and help Lovendusky Consulting meet client expectations when scheduling onsite meetings with clients.

Data Communication Precaution

The flight delay prediction system does not directly deal with or handle sensitive data as defined by HIPPA, FERPA, PCI DSS, or any such regulation. The data set that will be used for development is open source and publicly available. Application access will be granted explicitly by user or group policy to the accounts provided by Lovendusky Consulting. The application link will only be made accessible to authorized users in view mode. However, the future data used to assess the long-term success of Lovendusky Consulting's travel budget cost savings objective will be sensitive. For this reason, discussions related to that objective's outcome will be treated as highly confidential. Autom8 will follow the principle of least privilege when evaluating this data. This data will be transmitted securely and will remain encrypted at rest.

Developer's Expertise

The senior engineer that will be spearheading this initiative has advanced knowledge in the machine learning field and a proven track record of producing quality products for our clients. Both the senior and junior engineer have extensive experience working with Python. The junior engineer has also held roles in data analytics and data visualization. The strengths of both engineers along with guidance from the senior engineer and BA will ensure this product can be delivered in a timely manner while meeting Autom8's quality expectations.

B. Project Proposal

Problem Statement

Autom8's analysis has shown that Lovendusky Consulting has experienced a drastic increase in annual business travel cost since 2020. The increased prevalence of flight delays was determined to be the root cause. The proposed flight delay prediction system will leverage machine learning to minimize this risk and associated costs. The CEO of Lovendusky Consulting has expressed their interest in piloting this product. The system will allow users to input flight detail information prior to booking. This data will be ingested by the machine learning model and provide an output indicating whether the flight is likely to be delayed. Data visualizations provided with the product will support the model's findings and provide insight regarding emerging patterns and underlying factors that influence flight delays. This will allow for better planning when scheduling onsite consultations with clients.

Customer Summary

The application will be web-based and will not require any special skill sets for interaction. The application will reside in a Google Colab notebook. This environment will require very minimal initial setup time for the user and eliminate the need for local system installation. After the initial setup, system interactions will be as simple as clicking buttons and responding to drop-down lists to provide input criteria related to flight details. The input interface will be familiar to most users as it will be implemented via Google form. The use of drop-down lists rather than raw text input fields will protect the system and increase usability. The system will utilize dictionaries to translate the user input fields from user-friendly format into the expected format for the model. Values such as Time and Length of flight will be abstracted from the user and calculated automatically based on the Airport From and Airport To values provided. The simplicity of the system from an end-user perspective will ensure the product is accessible to all Lovendusky Consulting's employees in roles requiring travel. The visualizations provided will be particularly useful for those in roles related to event planning and those responsible for making business decisions regarding approved travel.

Existing System Analysis

There is currently no existing system or tooling used by Lovendusky Consulting for flight delay prediction. While some of the calculations and analysis could be done manually, there would be a significant risk of human-error. The time and labor needed to perform these efforts manually would also be detrimental to efficiency and the company's bottom line. By leveraging machine learning, vital information can be provided in seconds with a much lower error rate.

Data

The data used to train and test the system will be sourced from a CSV file. The file will be read from a GitHub repository URL embedded into the notebook. Backup copies will also be maintained for use should the original copy become corrupted or altered. The risk of file corruption or altering will be minimized as the users will have view only access to the system. The initial dataset utilized was obtained from Kaggle.com. The raw dataset contains approximately 540,000 records of recent flight information. The dataset source can be viewed here: https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay. The expected update frequency is indicated as annually. However, Autom8 will have the ability to expand upon the dataset by other means should the original dataset fail to be updated at an acceptable frequency to retain relevance. Irrelevant data fields such as id and Flight will be removed from the dataset before training and testing the system. Certain fields such as Airline, AirportFrom, and AirportTo will be replaced with dummy columns using the Pandas.get_dummies() method for model compatibility. Additionally, records not related to the primary US airlines or airports will be removed. This will allow the application's focus and precision to remain centered on the airlines and airports of interest. The fields provided in the raw dataset include the following:

- id (Data Type – Integer)
- Airline (Data Type – String)
- Flight (Data Type - Integer)
- AirportFrom (Data Type – String)
- AirportTo (Data Type – String)
- DayOfWeek (Data Type – Integer)
- Time (Data Type – Integer)
- Length (Data Type – Integer)
- Delay (Data Type – Integer)

Project Methodology

The solution development will follow the SEMMA methodology while implementing a Scrum-based approach for tracking milestone completion. This is appropriate for this project because it is designed to be highly iterative in nature as is supervised machine learning. The phases of SEMMA include Sample, Explore, Modify, Model, and Assess.

The SEMMA phases will be conducted as follows:

1. Sample – A subset of the data will be extracted from the full data set that provides equal representation of the data to be analyzed. This will ensure bias is not introduced to the model.

2. Explore – Relationships between the data points will be examined to determine correlation and relevance to flight delay prediction outcome.

3. Modify – Variables found to be irrelevant to the goal of the model will be removed before re-training and re-testing the model.

4. Model – The processed data will then be used to build the model. Data visualization modeling will also occur in this phase.

5. Assess– The model will undergo testing and performance metrics such as accuracy assessment will be applied. Based on the findings, this process may be repeated as needed until the required level of accuracy is achieved and the ideal machine learning algorithm is realized.

Project Outcomes

The project deliverables include the following:

- Visual representations of the data that support the model's prediction results.

- A user-friendly web-based application that can be used to input flight information and output predicted delay results.

- A user guide that will explain how to effectively interact with the application.

Implementation Plan

The requirements for the flight delay prediction product have been predetermined through datamining Lovendusky Consulting's annual travel budget to identify relevant airlines and airports to be included. The project will be implemented over the course of four 2-week long sprints. The first sprint will include data import and scrubbing and relationship analysis as the milestones. The second sprint's milestones will include model training and assessment. The third sprint's milestones will include visualization development, interactive component development, and user guide completion for the team's BA. The final sprint's milestones include conducting UAT and finalizing the production release.

As mentioned in the Project Methodology section, the SEMMA methodology will be utilized for the development implementation. Lovendusky Consulting will supply Autom8 with a list of users to be included for UAT and a final list for production. The BA will communicate with Lovendusky Consulting to gather the user lists. This communication will require encryption and the BA will provide the instructions to adhere to the encryption requirements. Once received, the BA will add the users to the application's allow list. The BA will ensure the users are able to access the application URL and maintain communication during UAT and warranty periods. The BA will pass along feedback to the engineers as necessary throughout the UAT process and warranty period.

Evaluation Plan

Objective	Success Criteria
Reduce Travel Budget By >= 25%	Compare the travel expenditure before using the ML application with the travel expenditure over the next 6-12 months after production deployment.
Provide Accuracy Rate >= 70%	Conduct performance assessments on the model and use logged responses to user input to monitor continual model performance.
System Uptime >= 98%	Review logs to ensure uptime and detect anomalies in response time before they become problematic to the client.

Resource and Costs

Programming Environment

The programming language that will be used to develop the system is Python 3.7. Open-source libraries such as Pandas, NumPy, Scikit-Learn, Matplotlib, and Seaborn will be utilized to streamline development efforts and provide data visualizations. Windows 10 is the OS that will be used for development. The primary development environment used will be Google Colab – cloud hosted Jupyter notebooks through Chrome browser. As mentioned previously, all development tools used are open source. No cost will be incurred for the programming environment.

Environment Costs

There will not be any cost associated with environment as all tools and infrastructure used are open source.

Human Resource Requirements

The development efforts and labor costs will effectively consume the entirety of the project's budgeted cost of $50,800. The cost breakdown analysis for the team resources is outlined in the table below.

Resource	Description	Cost
BA	$35/hr * 80 hours	$2,800
ML Engineers (2)	$75/hr * 320 hours * 2	$48,000
Software, Infrastructure, Data, Hosting Costs	All Open Source	$0
	Total	$50,800

Timeline and Milestones

The sprint breakdown for each of the four two-week periods is outline below.

Sprint	Start Date	End Date	Task - Resource Assigned	Dependencies
S1: 8/1/2022 – 8/14/2022	8/1/2022 8/8/2022	8/7/2022 8/14/2022	1. Data import and scrubbing – Development Team 2. Relationship analysis – Development Team	Task 1 = None Task 2 = Task 1
S2: 8/15/2022 – 8/28/2022	8/15/2022 8/22/2022	8/21/2022 8/28/2022	3. Model training – Development Team 4. Model assessment – Development Team	Task 3 = Task 1-2 Task 4 = Task 1-3
S3: 8/29/2022 – 9/11/2022	8/29/2022 9/5/2022 8/29/2022	9/4/2022 9/11/2022 9/11/2022	5. Data visualizations – Development Team 6. Interactive components – Development Team 7. Complete user guide – Business Analyst	Task 5 = Task 1-4 Task 6 = Task 1-5 Task 7 = Task 1-6
S4: 9/12/2022 – 9/25/2022	9/12/2022 9/19/2022	9/18/2022 9/25/2022	8. Conduct UAT – Development Team, Business Analyst, Stakeholders 9. Finalize production release – Development Team, Business Analyst, Stakeholders	Task 8 = Task 1-7 Task 9 = Task 1-8

C. Product Attributes

Data Methods & Data Visualization

Random forest was used as the prescriptive method for this project. Initially, logistic regression was considered for the application. However, the highest accuracy score obtained using that approach was just under 70%. XGBoost and naïve Bayes were also compared and achieved a similar accuracy score. The ability to evaluate multiple decision trees and take the average proved to be extremely effective for predicting flight delay. The random forest classifier model was able to achieve an 81% accuracy rate with little need for parameter tuning.

The descriptive methods used involved analysis depicted through data visualizations. A bar chart was used to illustrate the feature importance breakdown of the model. The feature importance breakdown was important to provide as it will be able to adapt once the dataset is expanded. This will allow the users to see emerging trends impacting flight delays and empower their decision-making ability regarding business travel. This method also proved that the day of the week was the most significant attribute in predicting flight delay. A pie chart was used to show the breakdown of the day of week in the dataset used to train the model. This supported the model because it showed the days of the week were evenly distributed and its feature importance was not related to skewed data or bias. Two additional bar charts were used to show the number of flights from each major US airline represented in the dataset and the number of flights from each airline that were delayed. This was an important visualization to provide to the business as three of the four airlines were amongst the top seven in terms of feature importance scoring. It also served to partially prove the hypothesis that one or more airlines would be a significant feature in predicting the likelihood of delay.

Datasets

The dataset used was sourced from a Kaggle site and was maintained for use in CSV format. The original dataset can be found here: https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay.

Data Cleaning

Data preprocessing was a very important aspect of this project. Many steps were taken to adjust the original data for model training. This process will be discussed in depth in prompt D.

Real-Time Queries

The real-time query feature was implemented through a Google form embedded in the notebook. The user can select an airline, airport from, airport to, and day of the week travelling from dropdown lists housed in the form. After selecting, the input is run through the model and an output of likely or unlikely to be delayed is provided to the user.

Adaptive Element

The application was built such that it will easily be able to adapt to an expanded dataset. This will allow the business to see emerging patterns impacting flight delay as the dataset expands. The dropdown lists in the form are also adaptable should additional airlines and airports need to be considered in the future.

Outcome Accuracy

The long-term accuracy outcome will be measured by verifying that Lovendusky Consulting's annual travel budget is reduced by a minimum of 25% after 6-12 months of product implementation. The accuracy outcome of the model, itself, exceeds the 70% minimum objective by 11% at 81%.

Security Measures

Application access will be granted explicitly by user or group policy to the accounts provided by Lovendusky Consulting. The application link will only be made accessible to authorized users in view mode. The data does not contain sensitive information.

Product health Monitoring

Logging the user input fields and delay prediction outcome of the model will allow Autom8 to monitor the health of the model. Logging will also enable detection of anomalies in response time and system errors.

Dashboard

The flight delay prediction system comprises data visualizations and interactive queries via a Jupyter notebook hosted in Google Colaboratory.

D. Post-Implementation Report

Project Purpose

The main objective of this project was to provide a prediction system using machine learning that could be used to determine the likelihood of flight delay. The system's purpose was to solve a problem that businesses such as Lovendusky Consulting are facing due to the increasing costs of business travel as a result of flight delays.

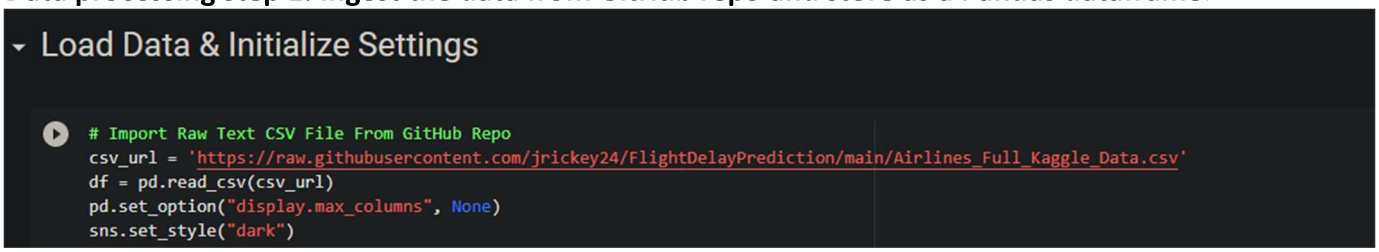
There was no alternative tooling or existing system in place that served this purpose for the client. The project resulted in a user-friendly web-based application. The application provides data visualizations that support the prediction model and provide valuable insight to the business of emerging patterns impacting flight delay. These features enable the users to make informed decisions when planning business travel and scheduling onsite meetings with clients. The application implements interactive queries using embedded Google form dropdowns to collect input. This feature allows the users to provide a prospective flight's details prior to booking and outputs the likelihood of delay. The system will provide cost reduction realization and better client retention for Lovendusky consulting.

Datasets

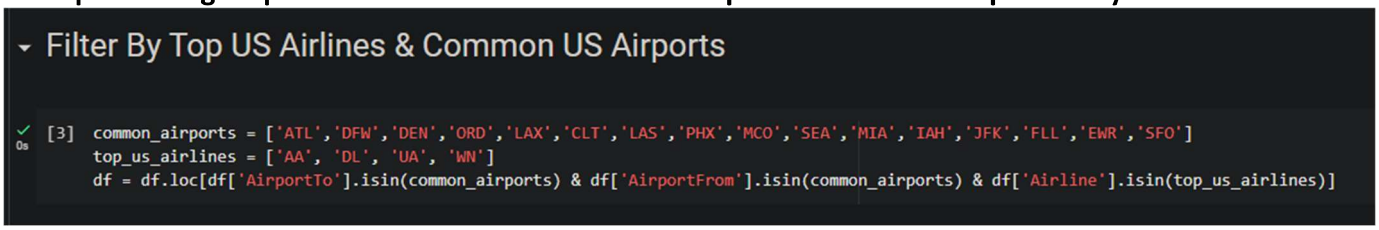
The dataset used to develop the project was acquired from a publicly available dataset hosted on Kaggle. The original dataset can be found here: https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay. The data was stored in CSV format in a GitHub repository. It is read from the URL embedded in the notebook. A visual example of the original dataset is provided below.

	A	B	C	D	E	F	G	H	I
1	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay
2	1	CO	269	SFO	IAH	3	15	205	1
3	2	US	1558	PHX	CLT	3	15	222	1
4	3	AA	2400	LAX	DFW	3	20	165	1
5	4	AA	2466	SFO	DFW	3	20	195	1
6	5	AS	108	ANC	SEA	3	30	202	0
7	6	CO	1094	LAX	IAH	3	30	181	1
8	7	DL	1768	LAX	MSP	3	30	220	0
9	8	DL	2722	PHX	DTW	3	30	228	0
10	9	DL	2606	SFO	MSP	3	35	216	1
11	10	AA	2538	LAS	ORD	3	40	200	1
12	11	CO	223	ANC	SEA	3	49	201	1
13	12	DL	1646	PHX	ATL	3	50	212	1
14	13	DL	2055	SLC	ATL	3	50	210	0
15	14	AA	2408	LAX	DFW	3	55	170	0
16	15	AS	132	ANC	PDX	3	55	215	0
17	16	US	498	DEN	CLT	3	55	179	0
18	17	B6	98	DEN	JFK	3	59	213	0
19	18	CO	1496	LAS	IAH	3	60	162	0
20	19	DL	1450	LAS	MSP	3	60	181	0

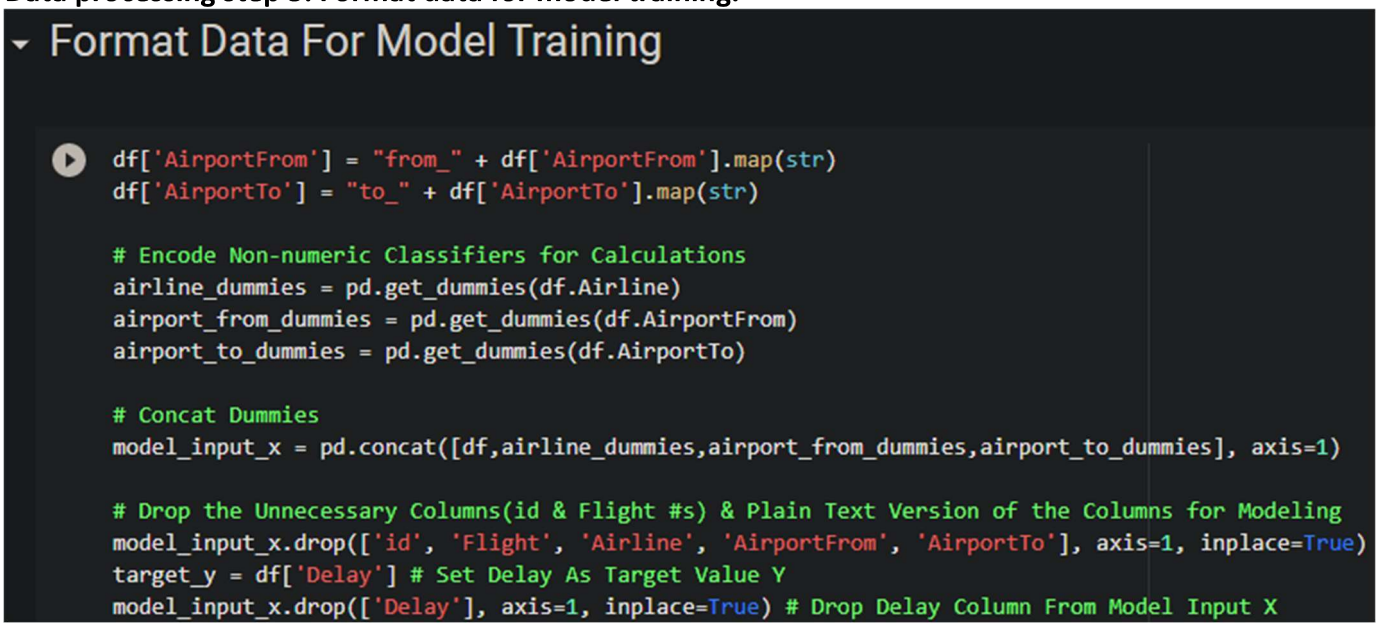
Data processing step 1: Ingest the data from GitHub repo and store as a Pandas dataframe.



Data processing step 2: Filter the dataset to contain top US airlines and airports only.



Data processing step 3: Format data for model training.



The first step in the step 3 process pictured above is to update the values in the AirportFrom column such that they are prepended with “from_” before each airport code. The second step is to do the same for the values in the AirportTo column such that they are prepended with “to_”. Next, the Pandas library pd.get_dummies() function is executed for the Airline, AirportFrom, and AirportTo columns. During this process, a new column is created for each unique Airline, AirportFrom, and AirportTo column value. The values in these new columns are expressed in binary(0 or 1). In this case, 1 indicates the given airline was the airline used and 0 indicates it was not. The next step is to concatenate the new dummie columns with the existing dataframe columns and drop the underlying columns. Other irrelevant columns(id & Flight) are dropped at this time as well. Finally, the Delay column is set as the target value y and the column is dropped from the dataframe.

First 10 rows of the dataframe with dummy columns in place and target value (Delay column) removed.

```
model_input_x.head(10)
```

	DayOfWeek	Time	Length	AA	DL	UA	WN	from_ATL	from_CLT	from_DEN	from_DFW	from_EWR	from_FLL	from_IAH	from_JFK
2	3	20	165	1	0	0	0	0	0	0	0	0	0	0	0
3	3	20	195	1	0	0	0	0	0	0	0	0	0	0	0
9	3	40	200	1	0	0	0	0	0	0	0	0	0	0	0
11	3	50	212	0	1	0	0	0	0	0	0	0	0	0	0
13	3	55	170	1	0	0	0	0	0	0	0	0	0	0	0
60	3	330	84	0	1	0	0	0	0	0	0	0	0	0	0
87	3	331	159	0	0	1	0	0	0	0	0	0	0	0	0
93	3	335	185	1	0	0	0	0	0	0	0	0	0	0	0
105	3	340	118	0	1	0	0	0	0	0	0	0	1	0	0
117	3	345	190	1	0	0	0	0	0	0	0	0	0	0	1

from_LAS	from_LAX	from_MCO	from_MIA	from_ORD	from_PHX	from_SEA	from_SFO	to_ATL	to_CLT	to_DEN	to_DFW
0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

to_EWR	to_FLL	to_IAH	to_JFK	to_LAS	to_LAX	to_MCO	to_MIA	to_ORD	to_PHX	to_SEA	to_SFO
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0

Data Product Code

Producing the predictive model required assessing the accuracy of multiple machine learning algorithms to select the best approach. The first approach tested was logistic regression. Using this method, the accuracy rate was consistently below 70%. Next, XGBoost and naïve Bayes were considered. However, they also produced a similar accuracy rate below the project's objective even after extensively attempting parameter tuning to improve performance. The next method attempted was random forest. Random forest's ability to evaluate multiple decision trees and take the average result proved to be extremely effective for predicting flight delay. The random forest classifier model was able to achieve an 81% accuracy rate with little need for parameter tuning.

Image of the random forest model and parameters used (800 trees with 15 max features and seed of 42). The model accuracy value is also expressed at 81%.

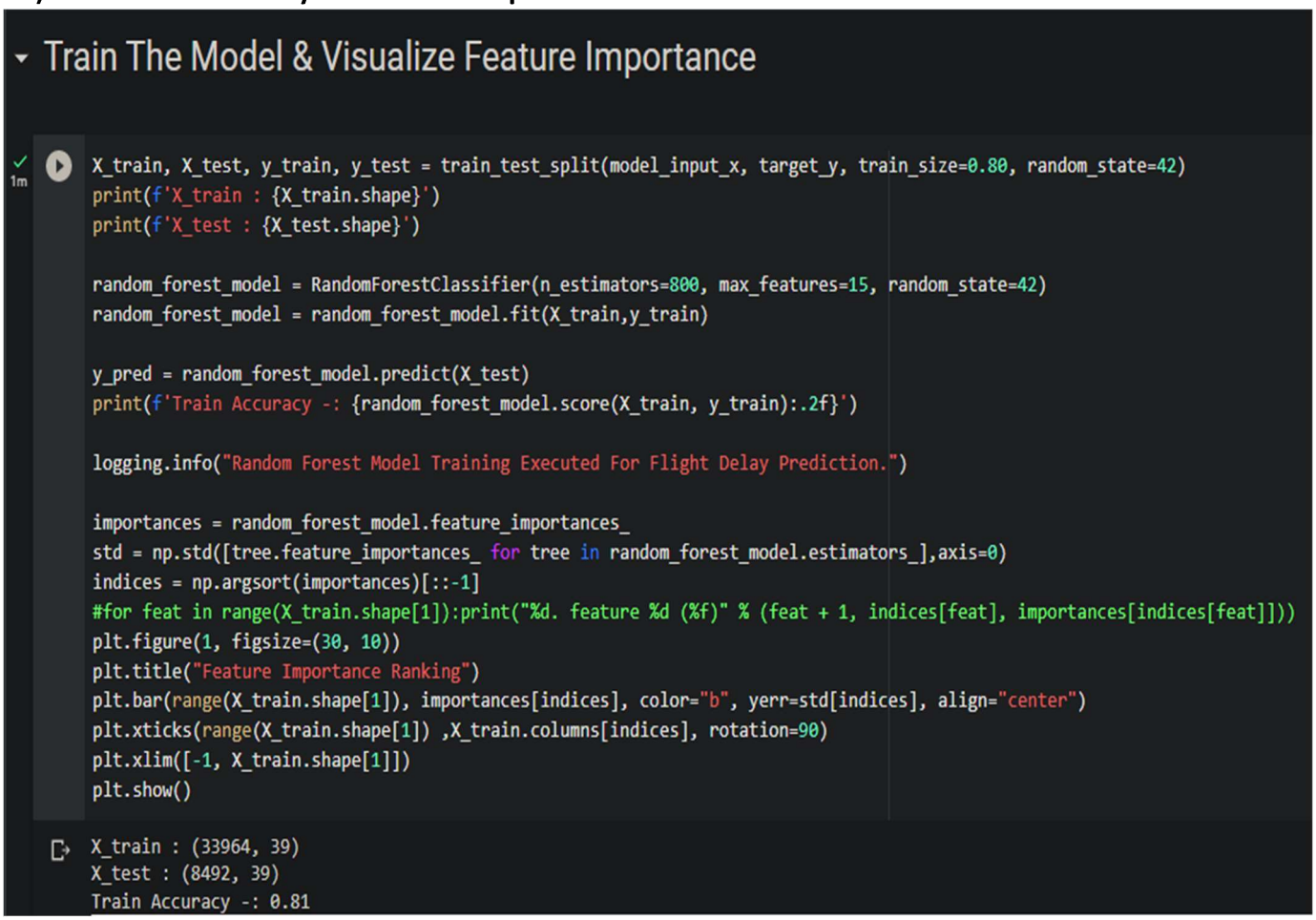


Image 1 of the interactive query method implemented using embedded Google form with dropdown lists.

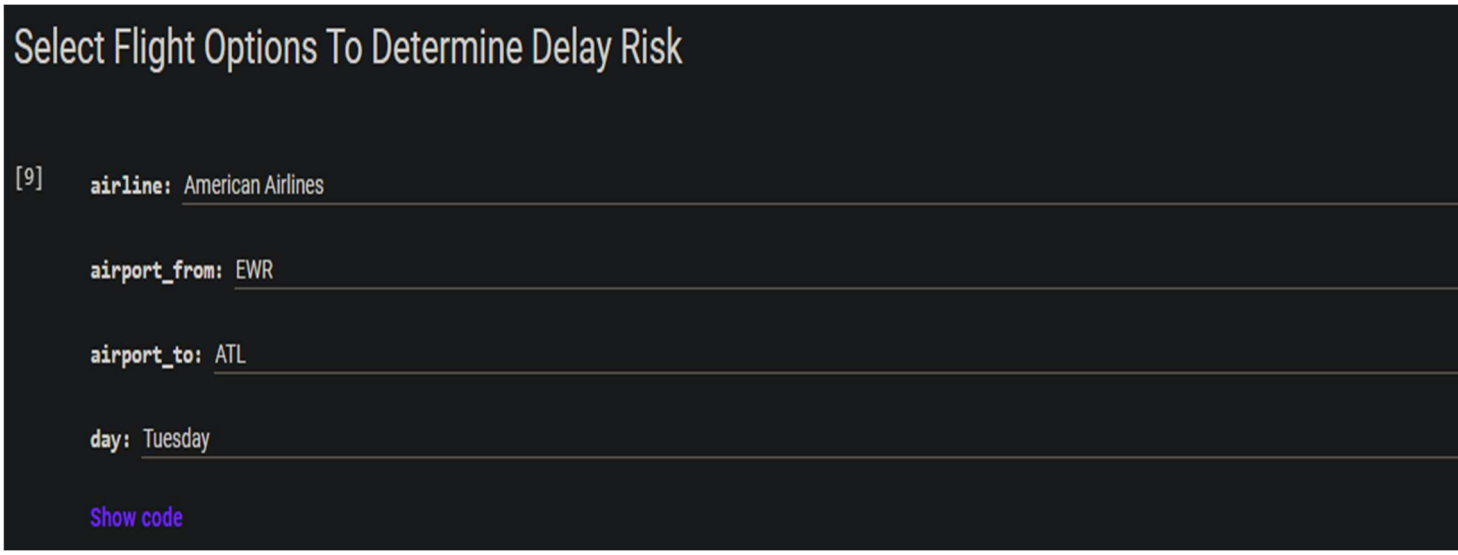


Image 2 of the interactive query method implemented showing airline options dropdown list.

Image 3 of the interactive query method showing airport_from dropdown options.

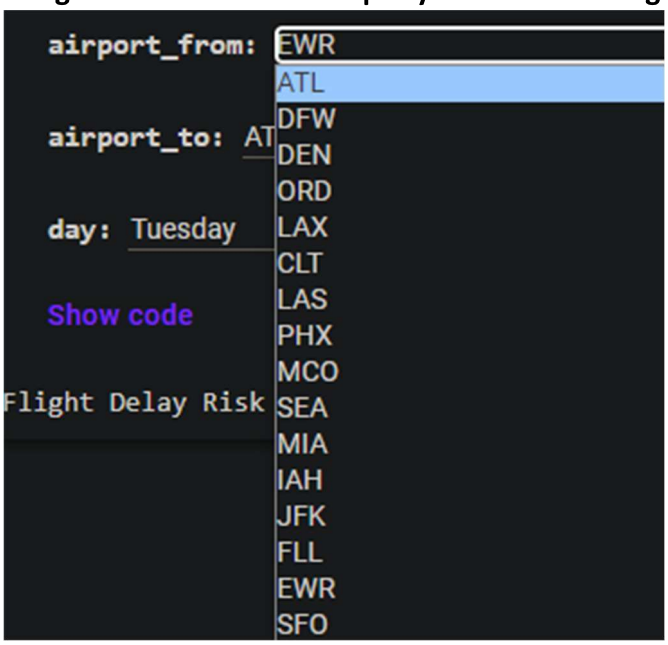


Image 4 of the interactive query method showing airport_to dropdown options.

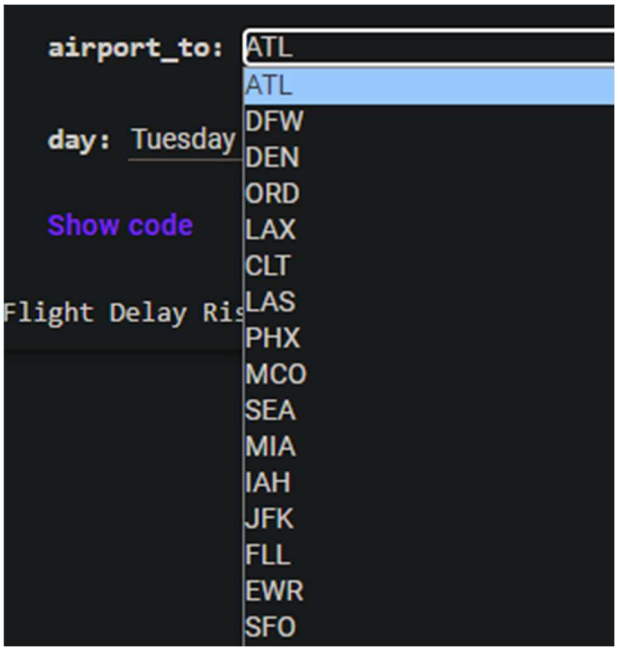
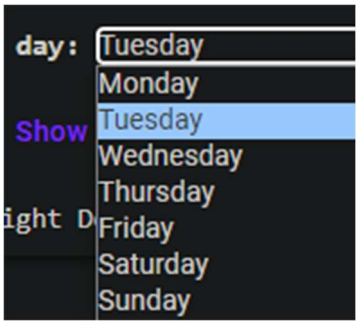


Image 5 of the interactive query method showing day dropdown options.



The interactive component allows the user to provide flight detail information and outputs the prediction response as “Flight Delay Risk Is High!” or “Flight Delay Risk Is Low”. Dropdowns were selected for this instead of free text to minimize the risk of errors from bad user input. This also ensures the input values only represent values for which the model was trained. However, error handling and additional data processing were needed to successfully handle every possible input. For example, if the user inputs the same airport for the from and to fields an error message will be displayed stating, “The Airport To and Airport From Combination Selected is Invalid. Please Select a Valid Option & Try Again.”. As the visuals show, there is no time or length input mechanism directly accessible to the user. This was abstracted from the user as that kind of information would not be readily available to them when booking a flight. Instead, the system generates these values automatically based on the other field values provided by the user.

Unit testing revealed that there were some airport values (to and from) for which the cleaned dataset had no records either within the subset of records, for the given airport, or at all. To handle the case where there is no record in the subset for the airline and airport from and to combination, the model tries to take the average length and time values for matching records with the same from and to airports. If this fails, a series of if/elif statements are executed to determine if the airport to and from was one of the pairs identified as missing from the original dataset entirely. An example case of this scenario is from “MIA” and to “FLL”. In those cases, there is no data available to extract and provide a genuinely accurate response so the user is notified the airport combination is invalid. Alternatively, if the data was found to exist in the original dataset for time and length between the two airports, the dictionary generated for missing values is used to fetch the values. The airport_time_length dictionary contains a list of the average time, length values extracted from the original dataset for which records were found. An additional dictionary, day_of_week_dict was implemented to convert the day of the week value into numerical form for model injection.

Image 1 of the underlying code used to transform the raw user input into model ready state for prediction analysis.

```
airline = "American Airlines" #@param ["Delta Airlines", "American Airlines","Southwest Airlines", "United Airlines"]
airport_from = "EWR" #@param ["ATL","DFW","DEN","ORD","LAX","CLT","LAS","PHX","MCO","SEA","MIA","IAH","JFK","FLL","EWR","SFO"]
airport_to = "ATL" #@param ["ATL","DFW","DEN","ORD","LAX","CLT","LAS","PHX","MCO","SEA","MIA","IAH","JFK","FLL","EWR","SFO"]
day = "Tuesday" #@param ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]

valid_flight = True
is_delayed = False

# Airport To And From Distance For Missing Values
airport_time_length = {"DEN_CLT":[539,185], "SFO_CLT":[733,286], "SFO_FLL":[1375,308], "LAX_IAH":[617,188], "LAX_SEA":[866,164], "LAX_CLT":[825,266],
"LAS_EWR":[862,284], "LAS_IAH":[620,174], "LAS_CLT":[761,246], "PHX_CLT":[568,227], "PHX_IAH":[804,154], "PHX_EWR":[863,275],
"MCO_EWR":[817,155], "MCO_SEA":[798,385], "MCO_IAH":[793,154], "MCO_CLT":[739,97], "MIA_SEA":[495,400], "MIA_CLT":[792,125],
"JFK_IAH":[1230,257], "JFK_CLT":[806,126], "FLL_CLT":[740,118], "FLL_IAH":[715,170], "FLL_EWR":[787,176], "FLL_SFO":[1022,380],
"SEA_LAX":[730,154], "SEA_CLT":[1014,281], "SEA_MCO":[945,320], "SEA_MIA":[1345,334], "SEA_IAH":[693,245], "SEA_EWR":[816,305],
"EWR_FLL":[787,176], "EWR_IAH":[799,241], "EWR_SEA":[829,371], "EWR_MCO":[744,173], "EWR_PHX":[716,335], "EWR_LAS":[778,346],
"EWR_CLT":[733,121], "IAH_LAX":[617,188], "IAH_CLT":[767,141], "IAH_LAS":[620,174], "IAH_PHX":[804,154], "IAH_MCO":[793,154],
"IAH_SEA":[693,245], "IAH_JFK":[1230,257], "IAH_FLL":[715,170], "IAH_EWR":[799,241],"CLT_DEN":[539,185], "CLT_LAX":[825,266],
"CLT_LAS":[761,246], "CLT_PHX":[568,227], "CLT_MCO":[739,97], "CLT_SEA":[1014,281], "CLT_MIA":[792,125],"CLT_IAH":[767,141],
"CLT_JFK":[806,126], "CLT_FLL":[740,118], "CLT_EWR":[733,121], "CLT_SFO":[733,286]}
```

Image 2 of the underlying code used to transform the raw user input into model ready state for prediction analysis.

```
# Get The Flight Time and Length Values For Model Computation
time_df= model_input_x.loc[(model_input_x["from_" + airport_from] == 1) & (model_input_x["to_" + airport_to] == 1)]
try:
  time_value = math.ceil(time_df["Time"].mean())
  length_value = math.ceil(time_df["Length"].mean())
except:
    if(airport_from == "MIA" and airport_to == "FLL") or (airport_from == "JFK" and airport_to == "EWR"):
      valid_flight = False
    elif(airport_from == "FLL" and airport_to == "MIA") or (airport_from == "FLL" and airport_to == "SEA"):
      valid_flight = False
    elif(airport_from == "SEA" and airport_to == "FLL") or (airport_from == "EWR" and airport_to == "JFK"):
       valid_flight = False
    elif(airport_from == airport_to):
      valid_flight = False
    else:
      time_value = airport_time_length[airport_from + "_" + airport_to][0]
      length_value = airport_time_length[airport_from + "_" + airport_to][1]
if valid_flight == False:
  print("The Airport To and Airport From Combination Selected is Invalid. Please Select a Valid Option & Try Again.")
```

Image 3 of the underlying code used to transform the raw user input into model ready state for prediction analysis.

```
# Transform DayOfWeek Value To Integer Representation For Model Computation
day_of_week_dict = {"Monday":1,"Tuesday":2,"Wednesday":3,"Thursday":4,"Friday":5,"Saturday":6,"Sunday":7}
day_num = day_of_week_dict[day]

# Transform Airline Value to Expected Input For Model Computation
airline_dict = {"American Airlines": "AA", "Delta Airlines": "DL", "Southwest Airlines": "WN", "United Airlines": "UA"}
airline_code = airline_dict[airline]

# Default Value of All Columns Except DayOfWeek, Time, and Length Are Initialized to Zero
test_df = pd.DataFrame({'DayOfWeek':[day_num], 'Time':[time_value], 'Length':[length_value], 'AA':[0], 'DL':[0], 'UA':[0], 'WN':[0],
       'from_ATL':[0], 'from_CLT':[0], 'from_DEN':[0], 'from_DFW':[0], 'from_EWR':[0], 'from_FLL':[0],
       'from_IAH':[0], 'from_JFK':[0], 'from_LAS':[0], 'from_LAX':[0], 'from_MCO':[0], 'from_MIA':[0],
       'from_ORD':[0], 'from_PHX':[0], 'from_SEA':[0], 'from_SFO':[0], 'to_ATL':[0], 'to_CLT':[0],
       'to_DEN':[0], 'to_DFW':[0], 'to_EWR':[0], 'to_FLL':[0], 'to_IAH':[0], 'to_JFK':[0], 'to_LAS':[0],
       'to_LAX':[0], 'to_MCO':[0], 'to_MIA':[0], 'to_ORD':[0], 'to_PHX':[0], 'to_SEA':[0], 'to_SFO':[0]})

# Set Values of Airline, Airport From & Airport To
test_df[airline_code] = 1
test_df["from_" + airport_from] = 1
test_df["to_" + airport_to] = 1
```

Image 4 of the underlying code used to transform the raw user input into model ready state for prediction analysis.

```
delay_message = "Flight Delay Risk Is High!"
no_delay_message = "Flight Delay Risk Is Low."

delay_prediction = random_forest_model.predict(test_df)
result_value = delay_prediction[0]

if valid_flight == True:
  if delay_prediction > 0:
    is_delayed = True
    print(delay_message)
  else: print(no_delay_message)
```

Hypothesis Verification

The initial hypothesis for this product stated that the airline used would be one of the strongest determinants of flight delay. This hypothesis was partially proven by the feature importance matrix that was run on the model paired with the descriptive visualization of delays by airline. Both visualizations show that flying with Southwest airlines more than doubled the likelihood of flight delay when compared with United Airlines, American Airlines, or Delta. However, the highest correlation to flight delay was found to be the day of the week flying. The top ten features identified are outlined below:

1. DayOfWeek
2. Time
3. Length
4. WN (Southwest Airlines)
5. UA (United Airlines)
6. from_ORD (from Chicago O'Hare International Airport)
7. AA (American Airlines)
8. to_SFO (to San Francisco International Airport)
9. from_LAX (from Los Angeles International Airport)
10. to_LAX (to Los Angeles International Airport)

The hypothesis regarding the use of the product's ability to reduce Lovendusky Consulting's annual travel budget costs by at least 25% will be evaluated once the system has been in place for six months and again after twelve months. Their budget comparisons from the prior three years will be analyzed to make this determination. If the six-month mark shows the saving are substantially behind the predicted savings value at that time, Autom8 will take measures to ensure the model and data are enhanced to reach the twelve-month target.

Effective Visualizations and Reporting

The descriptive methods used involved analysis depicted through data visualizations. These visualizations served to provide data to the user in an easily interpretable format. The visualizations also proved to significant in support the model's predictions.

A bar chart was used to illustrate the feature importance breakdown of the model. The feature importance breakdown is a relevant artifact because it will be able to adapt once the dataset is expanded. This will allow the users to see emerging trends impacting flight delays and empower their decision-making ability regarding business travel. This method also proved that the day of the week was the most significant attribute in predicting flight delay.

Image 1 of the bar chart outlining feature importance ranking.

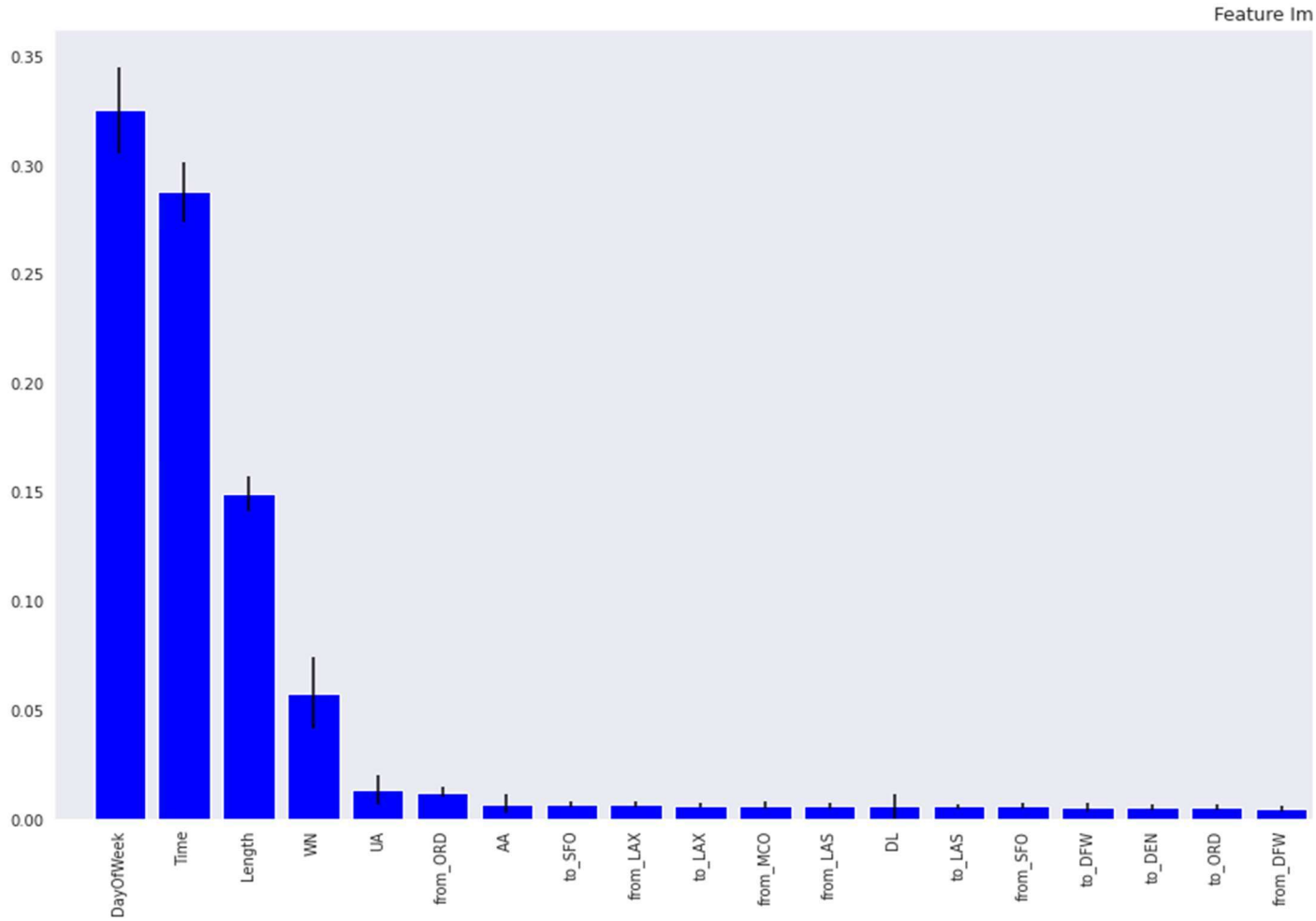
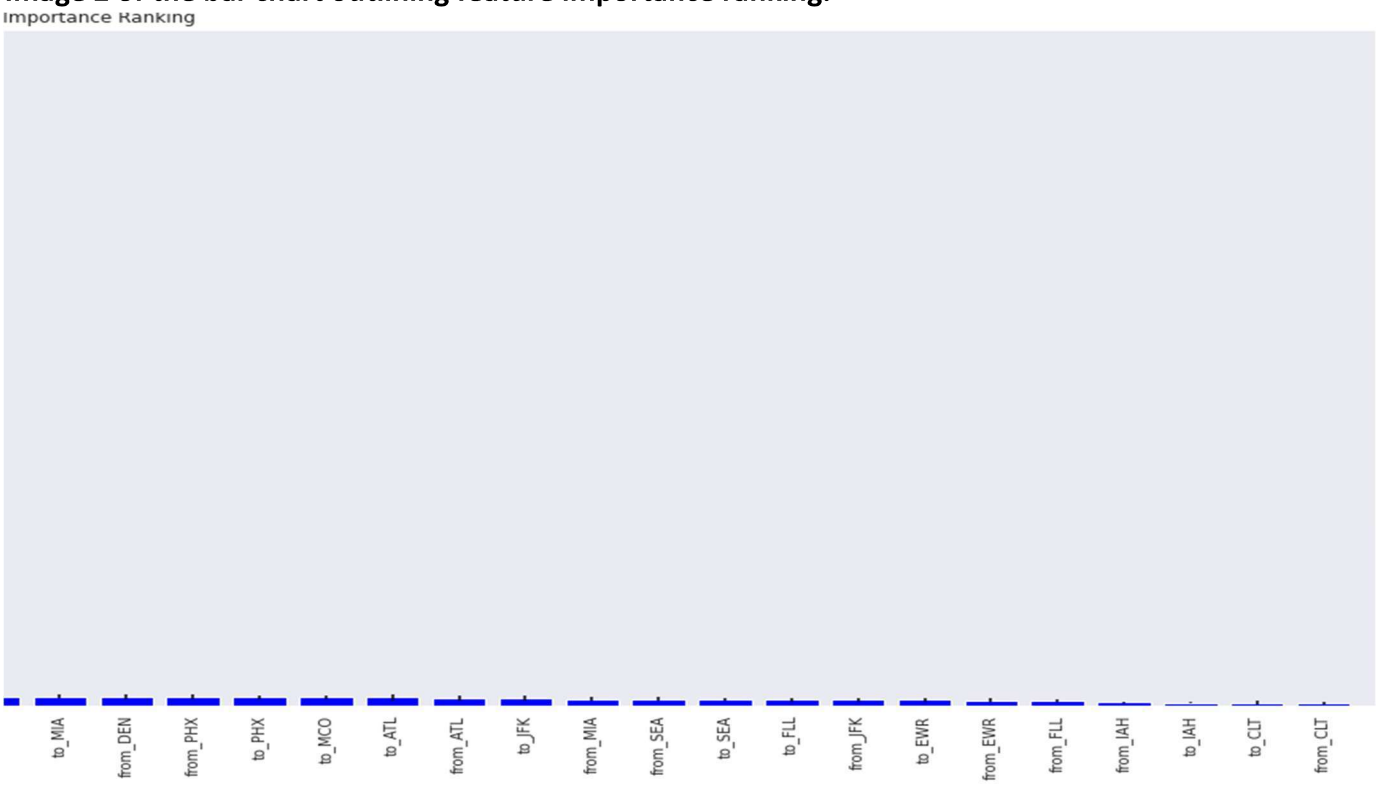
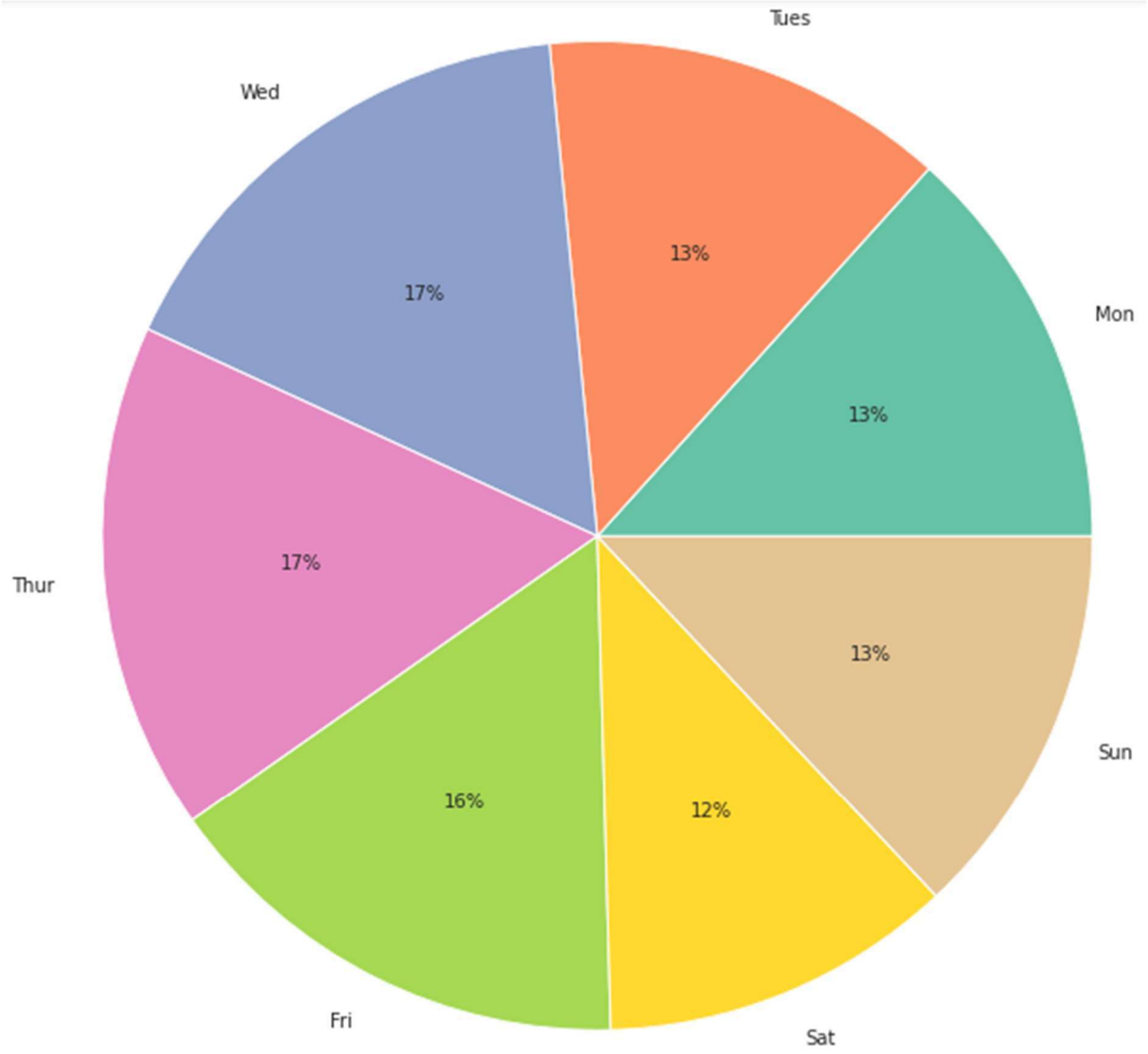


Image 2 of the bar chart outlining feature importance ranking.



A pie chart was used to show the breakdown of the day of week in the dataset used to train the model. This supported the model because it showed the days of the week were evenly distributed and its feature importance was not related to skewed data or bias.

Image of the pie chart depicting evenly distributed day of the week values.



Two additional bar charts were used to show the number of flights from each major US airline represented in the dataset and the number of flights from each airline that were delayed. This was an important visualization to provide to the business as three of the four airlines were amongst the top seven in terms of feature importance scoring. It also served to partially prove the hypothesis that one or more airlines would be a significant feature in predicting the likelihood of delay.

Image of the bar chart showing the breakdown of flight records grouped by airline.

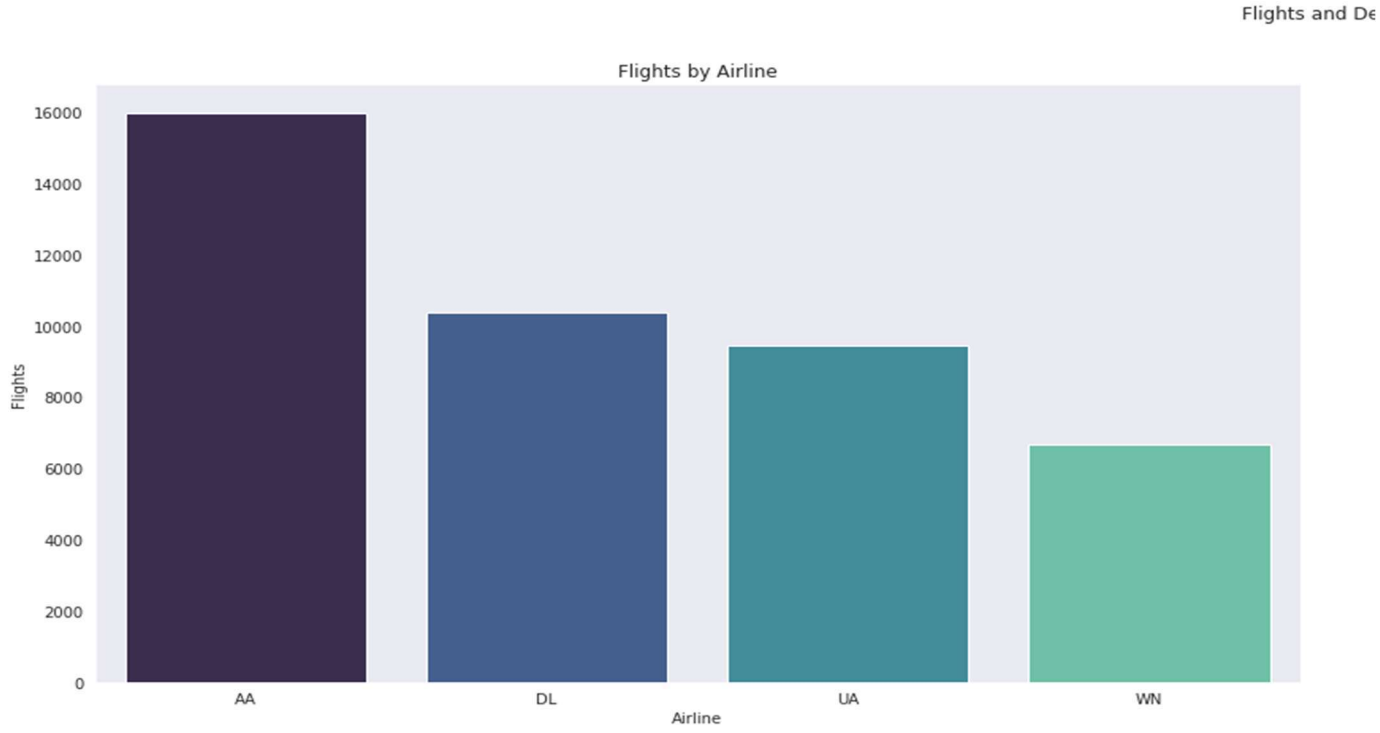
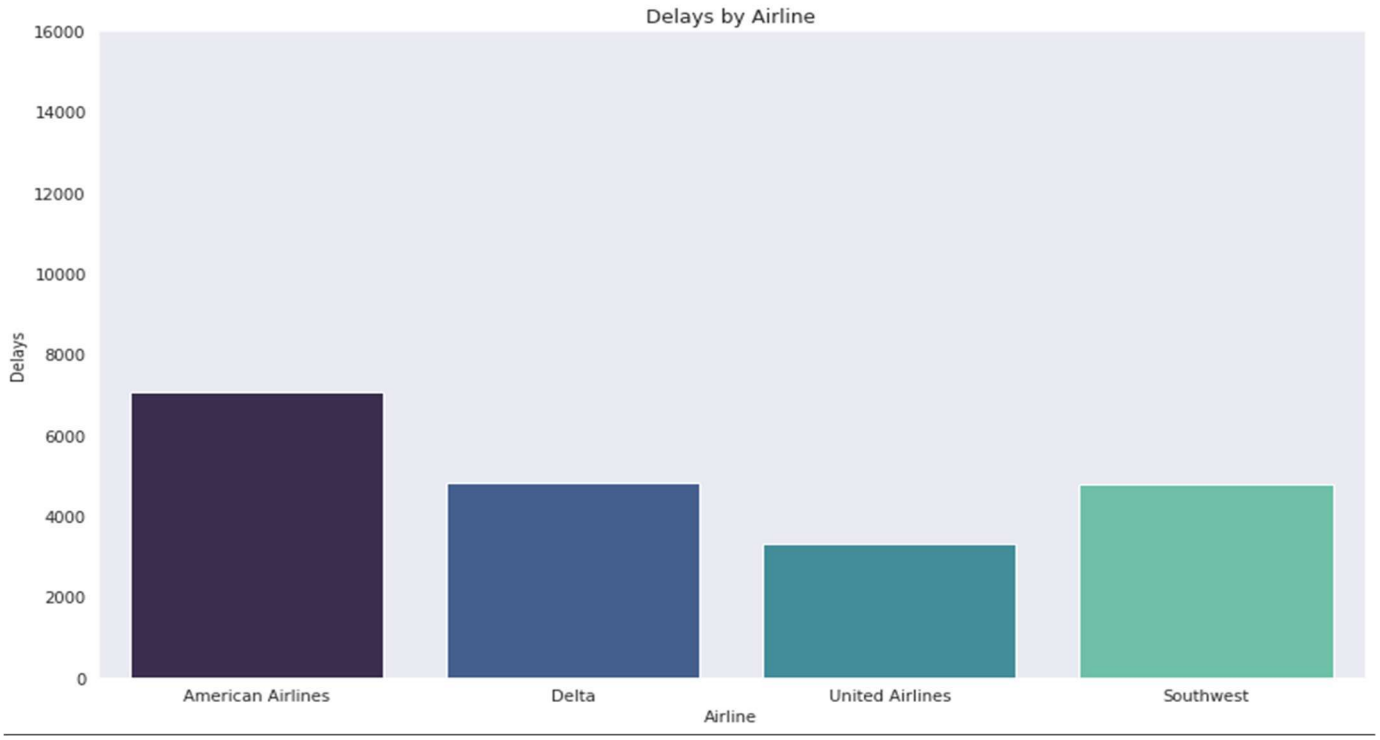


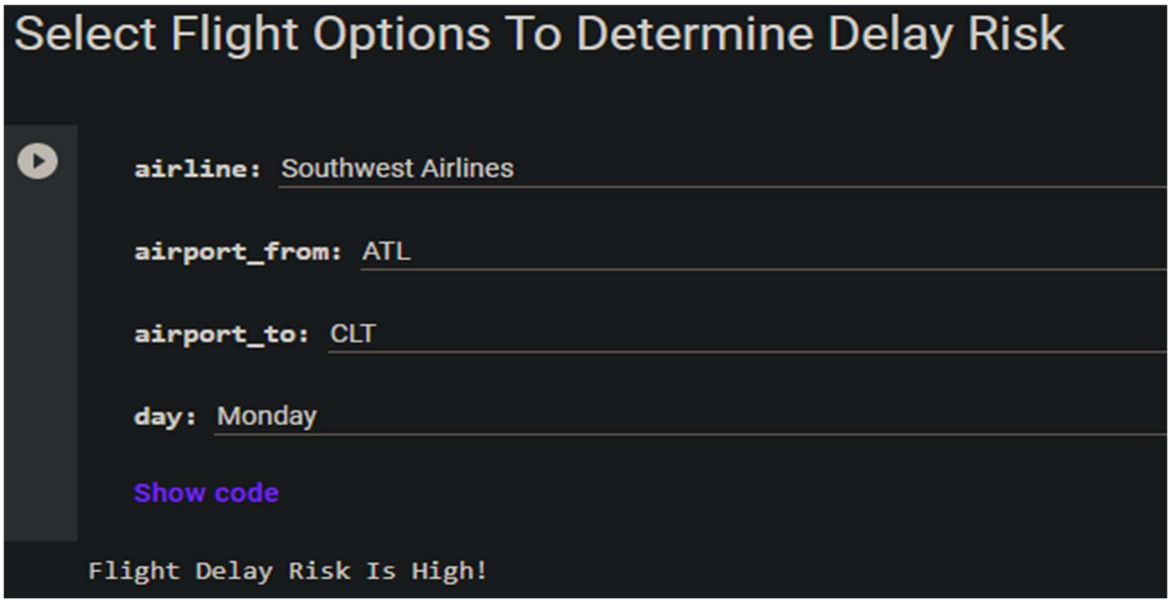
Image of the bar chart showing the breakdown of flight delays by airline.



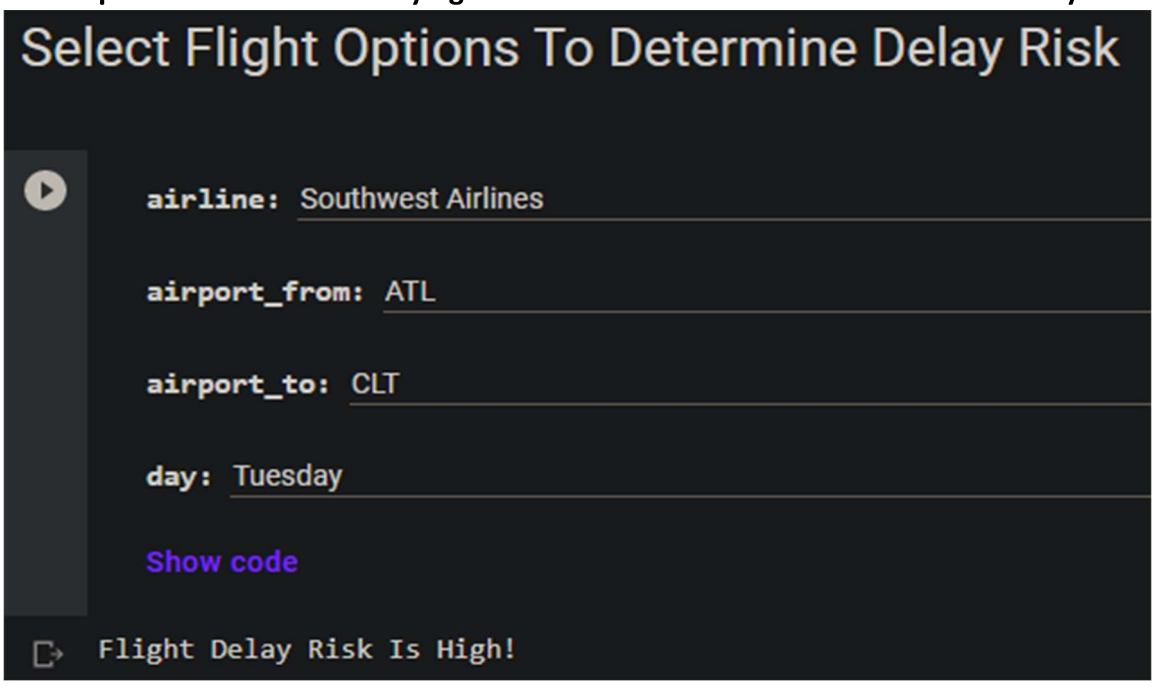
Accuracy Analysis

As previously mentioned, the outcome objective of reducing Lovendusky Consulting's annual travel budget by a minimum of 25% will not be determinable until at least six months post production deployment. The model's accuracy score of 81% as depicted in the Data Product Code section supports the objective to achieve a minimum of 70% accuracy. Log files of the model's response to the user input values will allow Autom8 to continuously evaluate the model's performance. The bar charts providing visualization of the airlines and delays support the model's correlation to flight delay when using Southwest Airlines. Additionally, testing the following examples flying from and to the same location on any given day of the week with Southwest Airlines supports the justification of the system response's prediction accuracy.

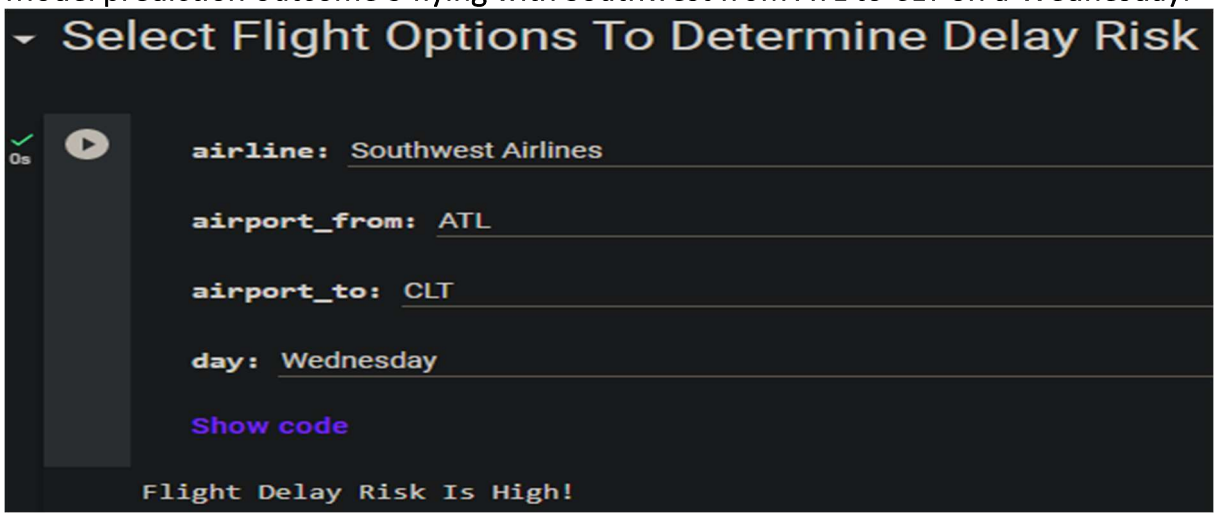
Model prediction outcome 1 flying with Southwest from ATL to CLT on a Monday.



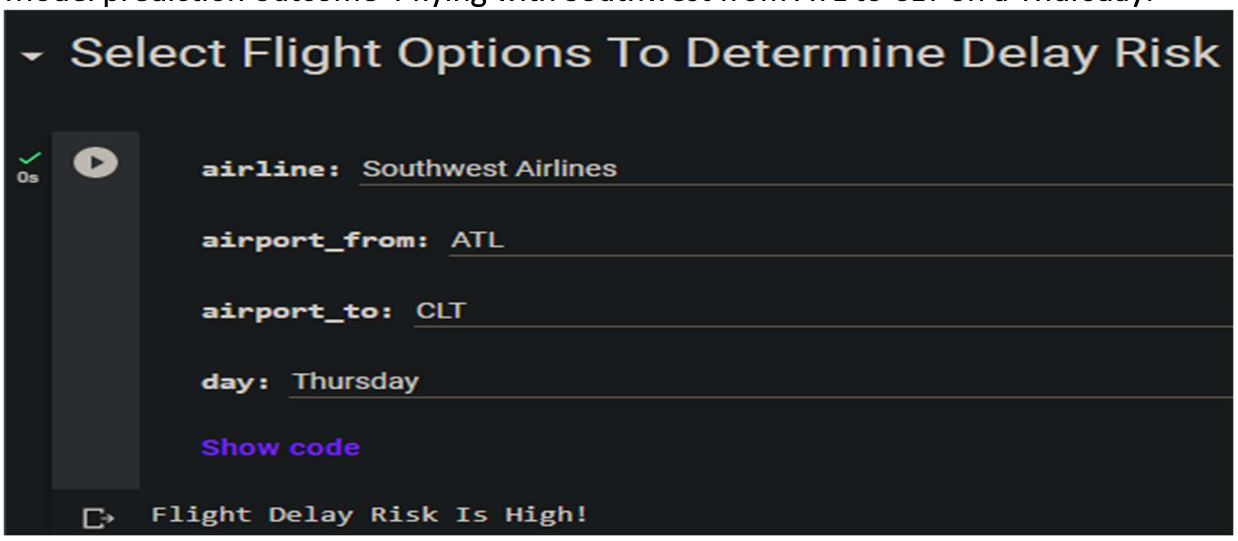
Model prediction outcome 2 flying with Southwest from ATL to CLT on a Tuesday.



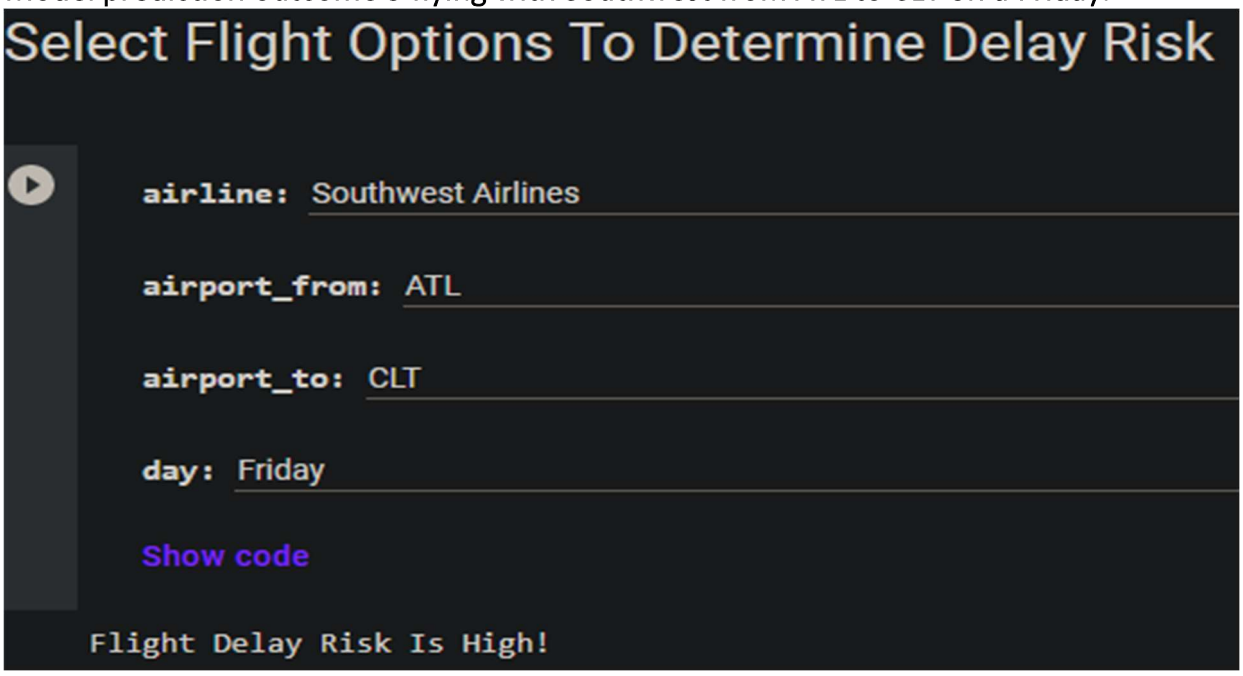
Model prediction outcome 3 flying with Southwest from ATL to CLT on a Wednesday.



Model prediction outcome 4 flying with Southwest from ATL to CLT on a Thursday.



Model prediction outcome 5 flying with Southwest from ATL to CLT on a Friday.



Application Testing

The flight delay prediction system was developed incrementally following the SEMMA methodology. As such, continuous integration (white box testing) was implemented throughout the SDLC. Additionally, the two developers conducted grey box testing by conducting peer reviews throughout the development. These tests were required to pass before the code updates could be pushed to the main branch of the project's repository. UAT was conducted during the project's final two-week sprint. The selected pool of end-users Lovendusky Consulting provided to the team's BA were given access to the system at this time. These end-users performed black box testing to ensure usability and functionality met quality standards. No significant defects were detected during the UAT process.

Application Files

The project files are hosted in the following GitHub repository: https://github.com/jrickey24/FlightDelayPrediction.

The FlightDelayPrediction.ipynb file is included in the project submission. However, the Airlines_Full_Kaggle_Data.csv file was not directly included in the submission due to size, direct accessibility from the repository, or through the Kaggle link previously provided. Note that the additional CSV file in the repository titled, Airlines_AA_DL_UA.csv, was used for testing purposes only and is not necessary for the application. The FlightDelayPrediction capstone repository includes the following files:

- FlightDelayPrediction.ipynb – Copy of the Jupyter Notebook file.
- Airlines_Full_Kaggle_Data.csv – The full dataset used for the project.

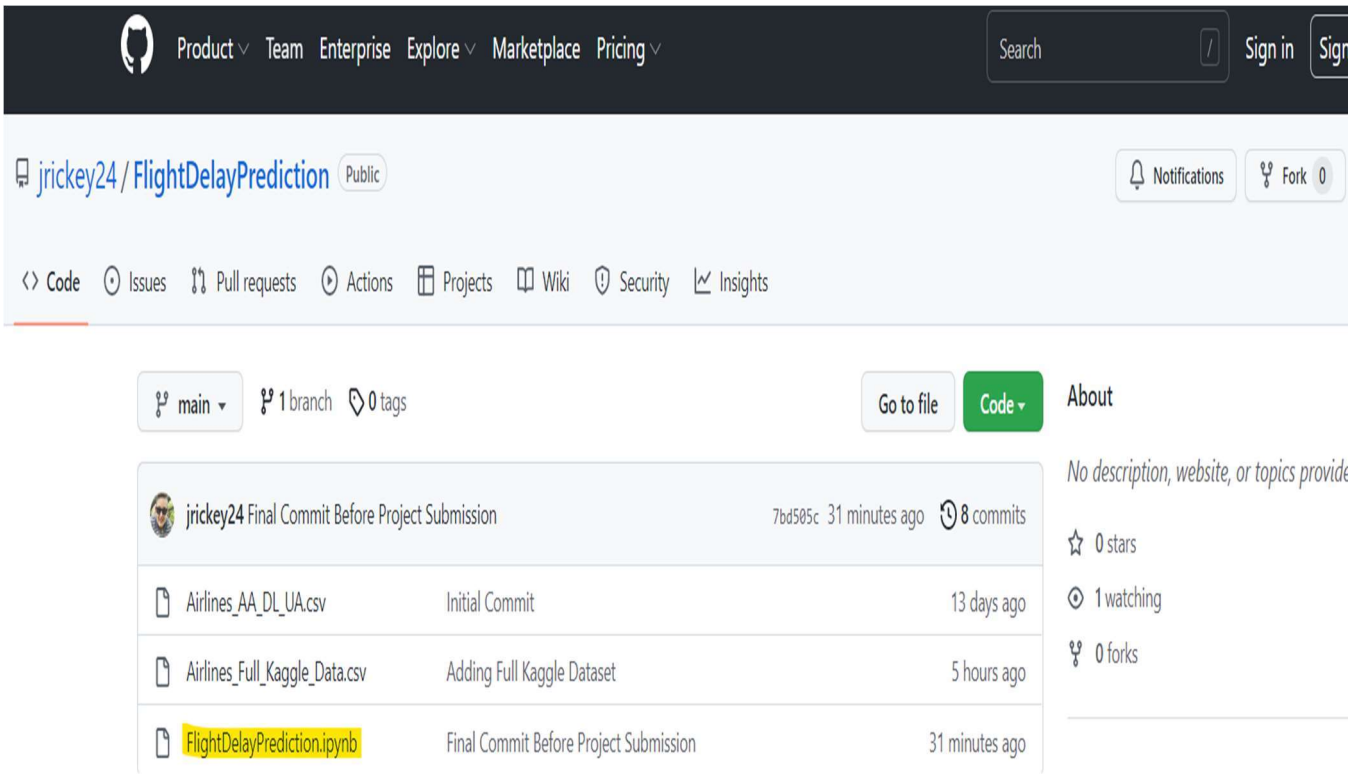
User's guide

The flight delay prediction system is hosted in Google Colaboratory. Please follow the steps below to perform system installation.

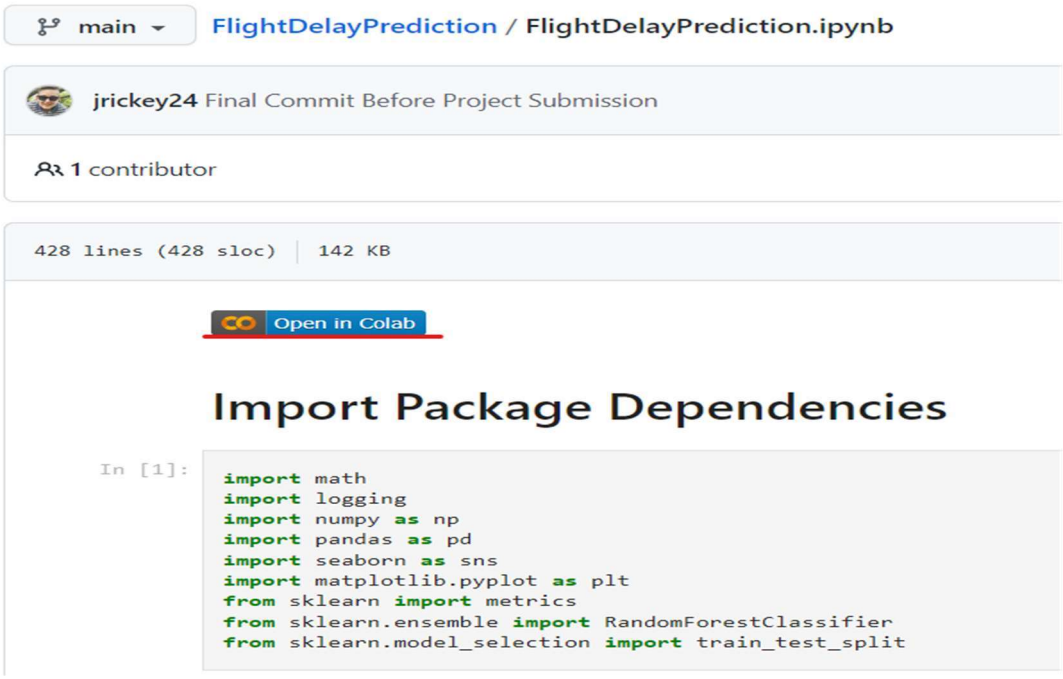
1. Verify that your browser and OS are updated to the latest version or minimally, to a currently supported version. The Colab environment is supported and tested in most major browsers. However, the recommendation is to use a Chrome browser to ensure best compatibility support.

2. Navigate to the following URL within your browser: https://github.com/jrickey24/FlightDelayPrediction

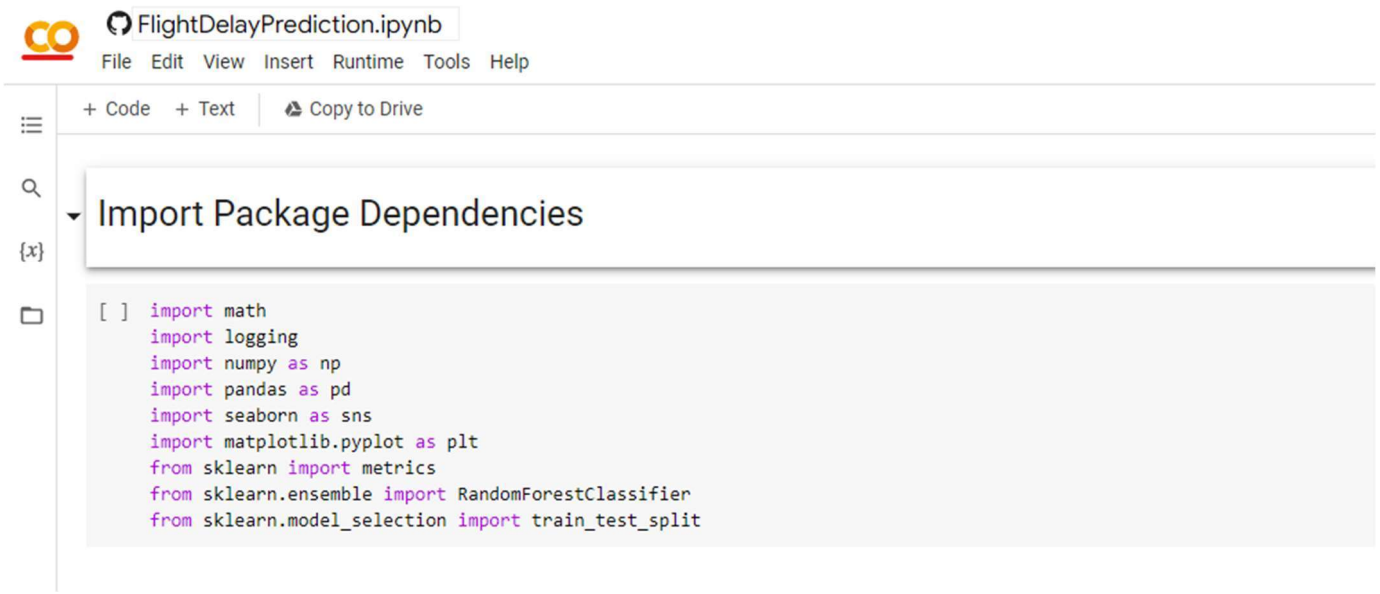
3. Click the "**FlightDelayPrediction.ipynb**" file as shown below.



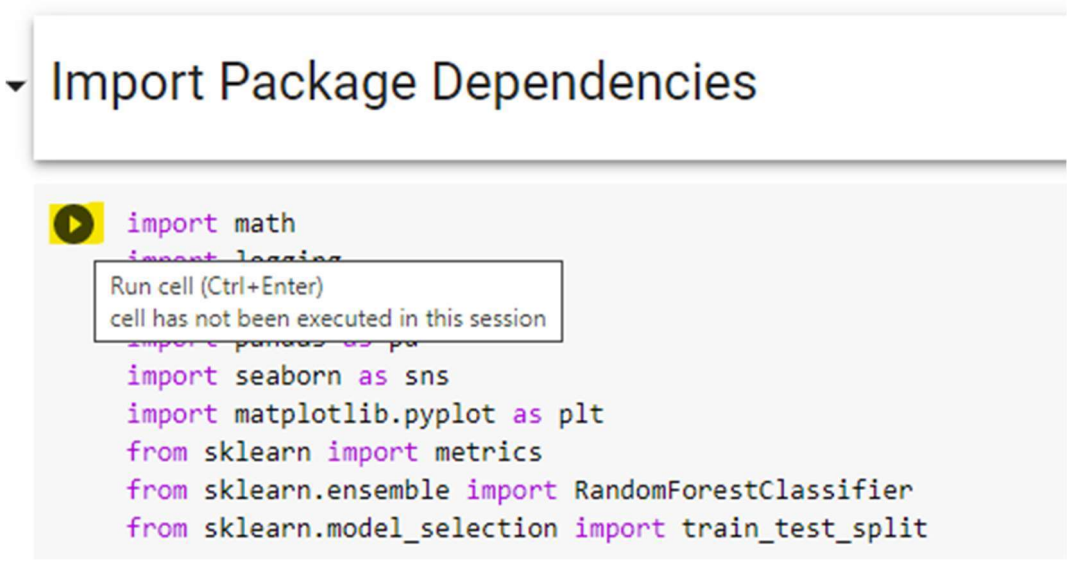
4. In the next screen, you should see the option to select "**Open in Colab**". Please click the button as pictured below to open the application in the Google Colaboratory environment.



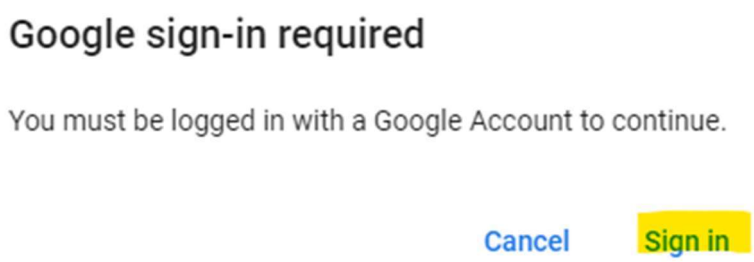
5. To confirm you are now in the **Google Colaboratory** environment, verify you see the following icon underlined in red in the image below in the top left corner of your browser screen.



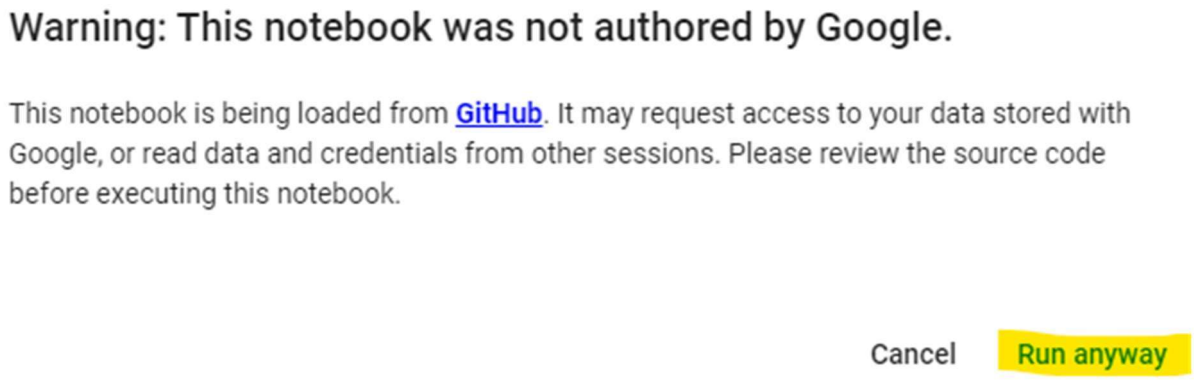
6. Click the play button in the first cell underneath the "**Import Package Dependencies**" heading as indicated below.



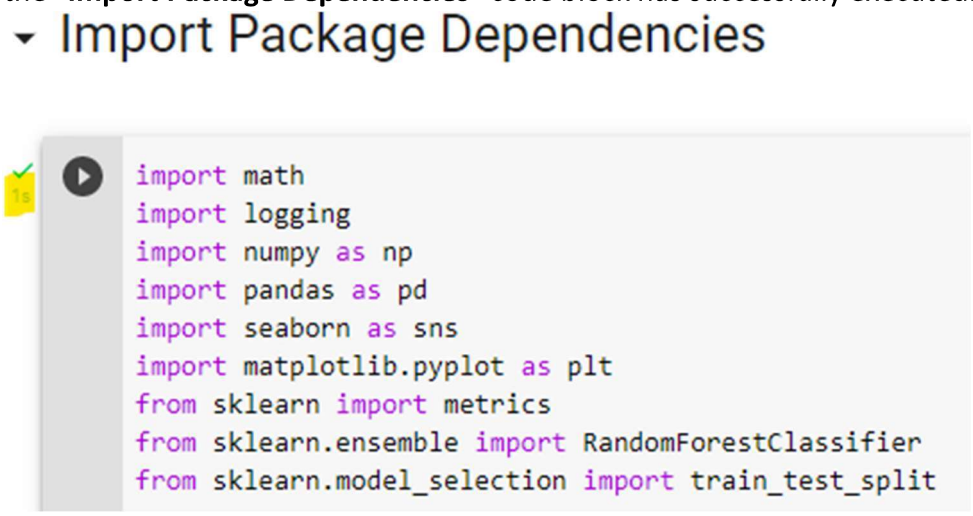
7. If you receive the following message, please follow the link to "**Sign in**" to your Google account in order to authenticate and continue the session.



8. Should you receive the following message, please "**Run anyway**" to continue.



9. You should see a **green checkmark** as indicated in the screenshot below. This confirms the "**Import Package Dependencies**" code block has successfully executed.



10. Once execution has been verified, please proceed to the following code block titled "**Load Data & Initialize Settings"** as indicated below. Repeat the prior step by clicking the play button located below the section header.

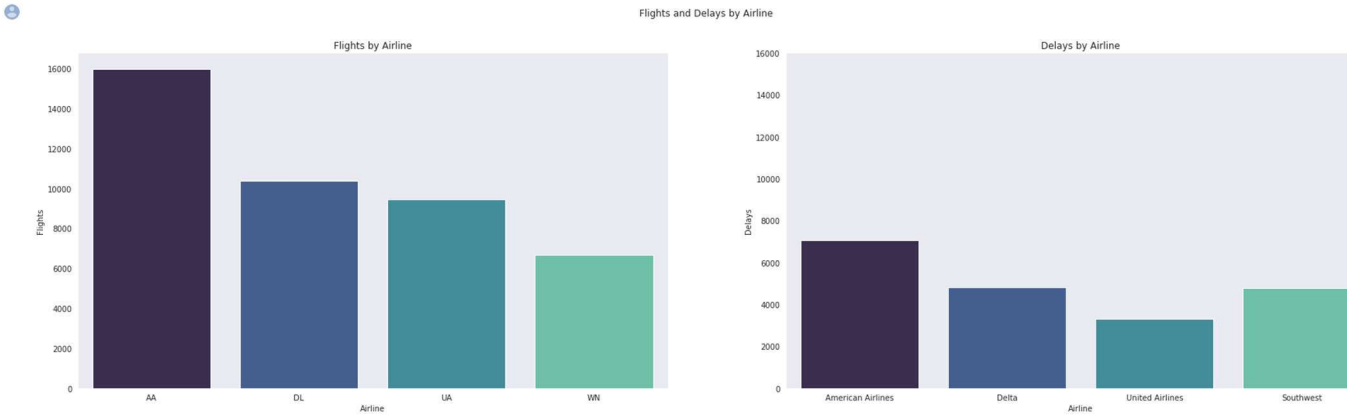
11. Repeat the step to click the play button for the following code block titled "**Filter By Top US Airlines & Common US Airports**".

12. Repeat the step to click the play button for the "**Visualize Flights & Delays By Airline**" section.

Visualize Flights & Delays By Airline

```
count_by_airline = df.groupby('Airline', as_index=False).Delay.count()
aa_details = df.apply(lambda x : True if x['Airline'] == 'AA' and x['Delay'] == 1 else False, axis = 1) # Get Count Of Delayed AA Flights = America
AA = len(aa_details[aa_details == True].index)
dl_details = df.apply(lambda x : True if x['Airline'] == 'DL' and x['Delay'] == 1 else False, axis = 1) # Get Count Of Delayed DL Flights = Delta
DL = len(dl_details[dl_details == True].index)
ua_details = df.apply(lambda x : True if x['Airline'] == 'UA' and x['Delay'] == 1 else False, axis = 1) # Get Count Of Delayed UA Flights = United
UA = len(ua_details[ua_details == True].index)
wn_details = df.apply(lambda x : True if x['Airline'] == 'WN' and x['Delay'] == 1 else False, axis = 1) # Get Count Of Delayed WN Flights = Southwe
WN = len(wn_details[wn_details == True].index)

delays_by_airline = {"American Airlines": AA, "Delta": DL, "United Airlines": UA, "Southwest": WN}
keys = list(delays_by_airline.keys())
values = list(delays_by_airline.values())

fig, axs = plt.subplots(ncols=2, figsize=(30,8))
fig.suptitle('Flights and Delays by Airline')
sns.barplot(x='Airline', y='Delay', data=count_by_airline, palette="mako", ax=axs[0])
sns.barplot(x=keys, y=values, palette="mako", ax=axs[1])
axs[0].set(title='Flights by Airline', xlabel='Airline', ylabel='Flights')
axs[1].set(title='Delays by Airline', xlabel='Airline', ylabel='Delays')
axs[1].set_ylim(0,16000);
```

13. The following graphs should now be displayed.

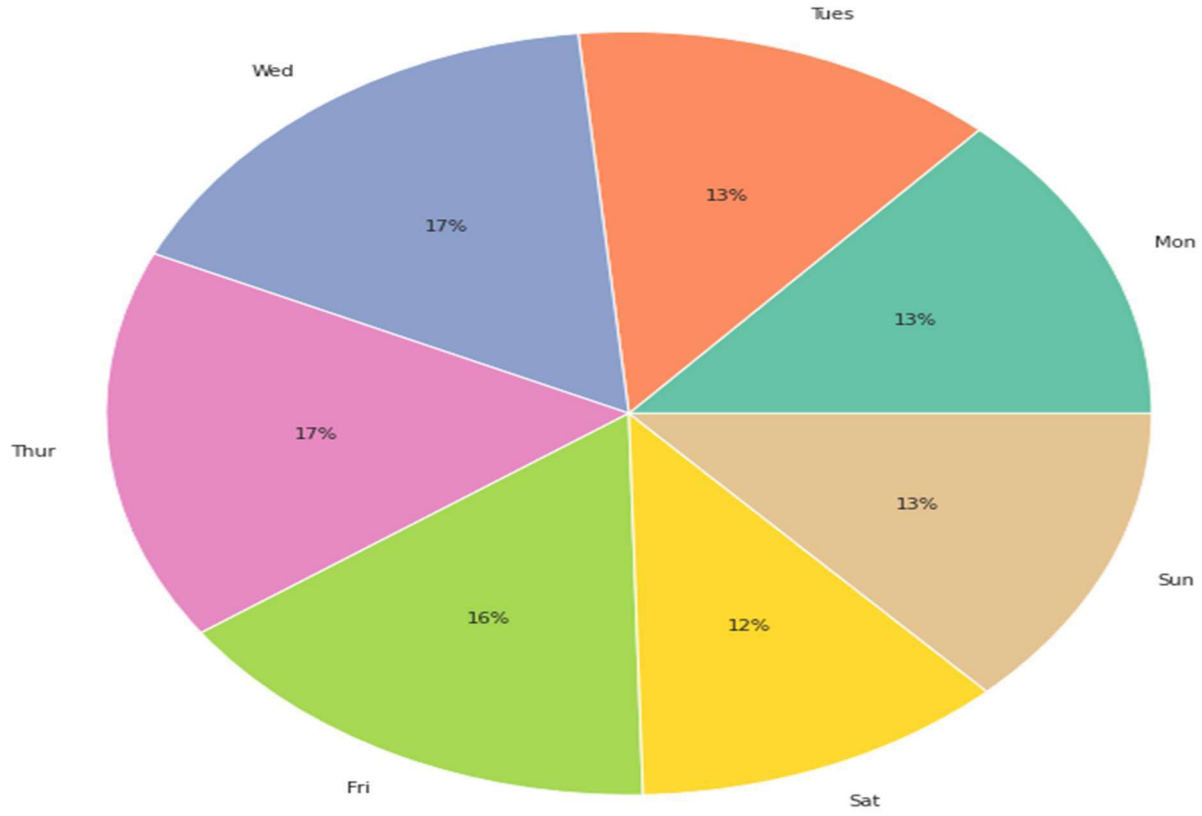


14. Click the play button below the "**Visualize Day Of The Week Distribution**" heading.

Visualize Day Of The Week Distribution

```
day_of_week = df['DayOfWeek'].value_counts(normalize=True)

m = day_of_week[1]
tu = day_of_week[2]
w = day_of_week[3]
th = day_of_week[4]
f = day_of_week[5]
sa = day_of_week[6]
su = day_of_week[7]

plt_values = [m,tu,w,th,f,sa,su]
plt_labels = ['Mon', 'Tues', 'Wed', 'Thur', 'Fri', 'Sat', 'Sun']

colors = sns.color_palette('Set2')[0:len(plt_labels)]
plt.pie(plt_values, labels=plt_labels, colors=colors, autopct='%.0f%%', radius=3
plt.show
```

15. The following pie chart should now be displayed.



16. Next, click the play button below the "**Format Data For Model Training**" heading.

Format Data For Model Training

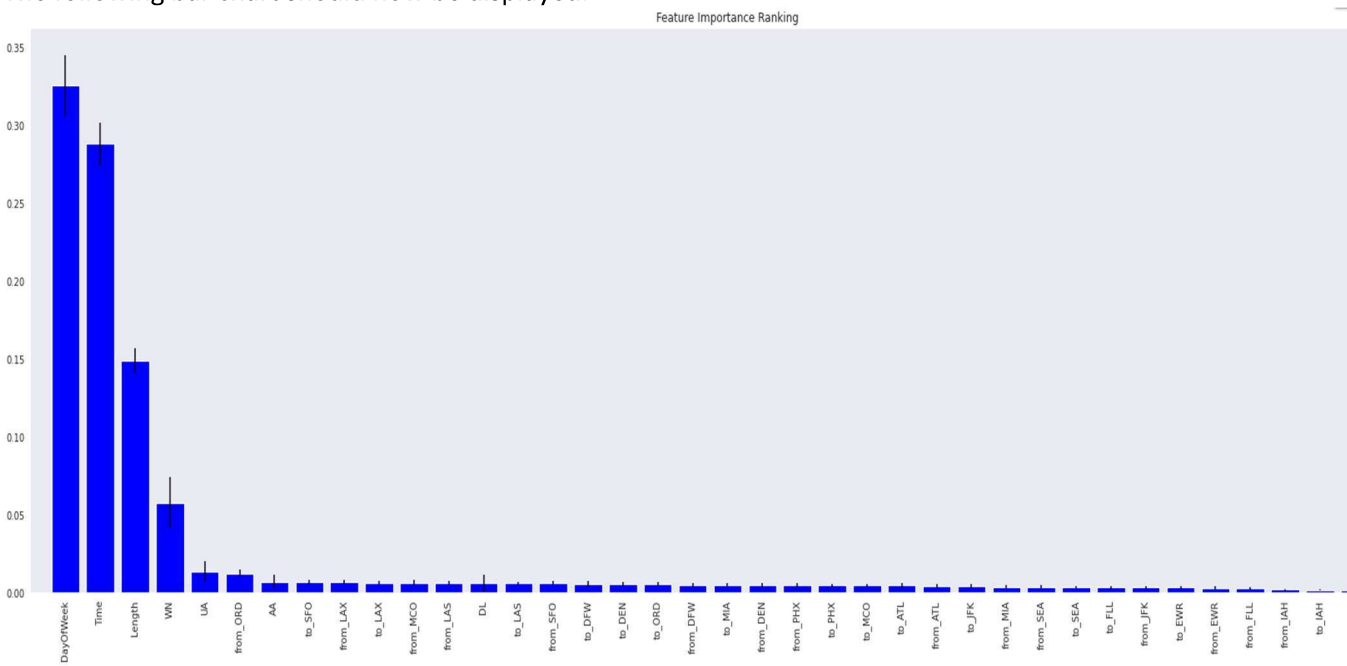
```
df['AirportFrom'] = "from_" + df['AirportFrom'].map(str)
df['AirportTo'] = "to_" + df['AirportTo'].map(str)

# Encode Non-numeric Classifiers for Calculations
airline_dummies = pd.get_dummies(df.Airline)
airport_from_dummies = pd.get_dummies(df.AirportFrom)
airport_to_dummies = pd.get_dummies(df.AirportTo)

# Concat Dummies
model_input_x = pd.concat([df,airline_dummies,airport_from_dummies,airport_to_dummies], axis=1)

# Drop the Unnecessary Columns(id & Flight #s) & Plain Text Version of the Columns for Modeling
model_input_x.drop(['id', 'Flight', 'Airline', 'AirportFrom', 'AirportTo'], axis=1, inplace=True)
target_y = df['Delay'] # Set Delay As Target Value Y
model_input_x.drop(['Delay'], axis=1, inplace=True) # Drop Delay Column From Model Input X
```

17. Proceed by clicking the play button below the "**Train The Model & Visualize Feature Importance**" heading. *Please note: The runtime for this code block will take longer than the previous blocks run. The average time to complete is 60 seconds. Please wait for the execution to complete before moving forward to avoid interruptions or errors. *

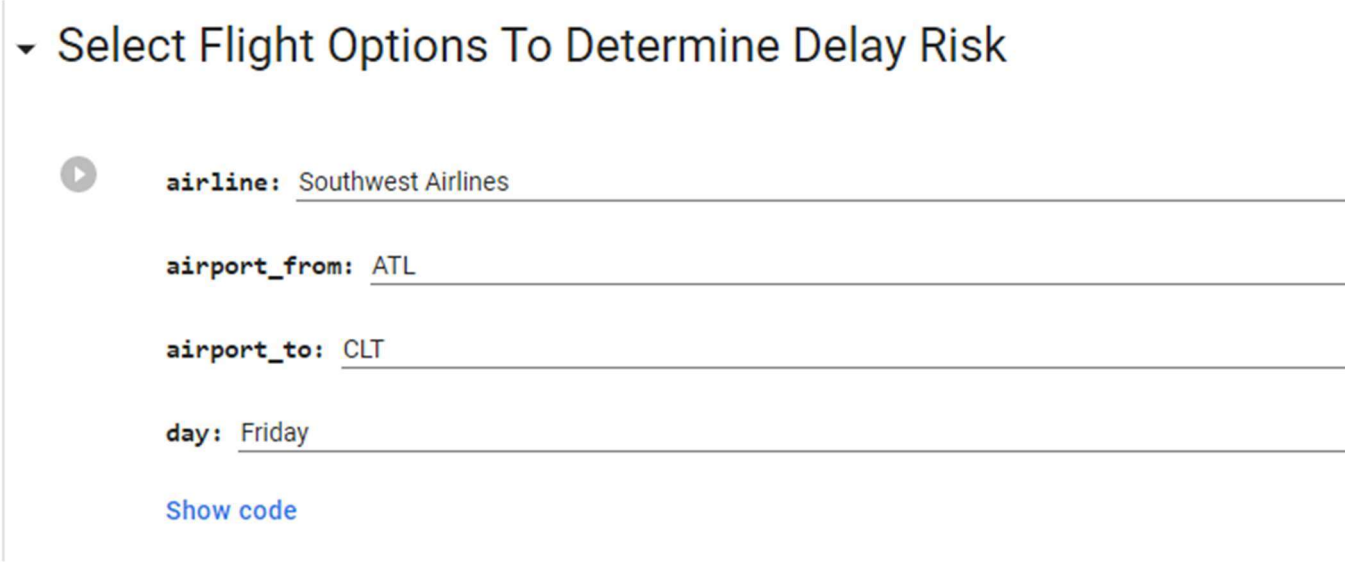
Train The Model & Visualize Feature Importance

```
X_train, X_test, y_train, y_test = train_test_split(model_input_x, target_y, train_size=0.80, random_state=42)
#print(f'X_train : {X_train.shape}')
#print(f'X_test : {X_test.shape}')

random_forest_model = RandomForestClassifier(n_estimators=800, max_features=15, random_state=42)
random_forest_model = random_forest_model.fit(X_train,y_train)

y_pred = random_forest_model.predict(X_test)
print(f'Train Accuracy -: {random_forest_model.score(X_train, y_train):.2f}')

logging.info("Random Forest Model Training Executed For Flight Delay Prediction.")

importances = random_forest_model.feature_importances_
std = np.std([tree.feature_importances_ for tree in random_forest_model.estimators_],axis=0)
indices = np.argsort(importances)[::-1]
#for feat in range(X_train.shape[1]):print("%d. feature %d (%f)" % (feat + 1, indices[feat], importances[indices[feat]]))
plt.figure(1, figsize=(30, 10))
plt.title("Feature Importance Ranking")
plt.bar(range(X_train.shape[1]), importances[indices], color="b", yerr=std[indices], align="center")
plt.xticks(range(X_train.shape[1]) ,X_train.columns[indices], rotation=90)
plt.xlim([-1, X_train.shape[1]])
plt.show()
```

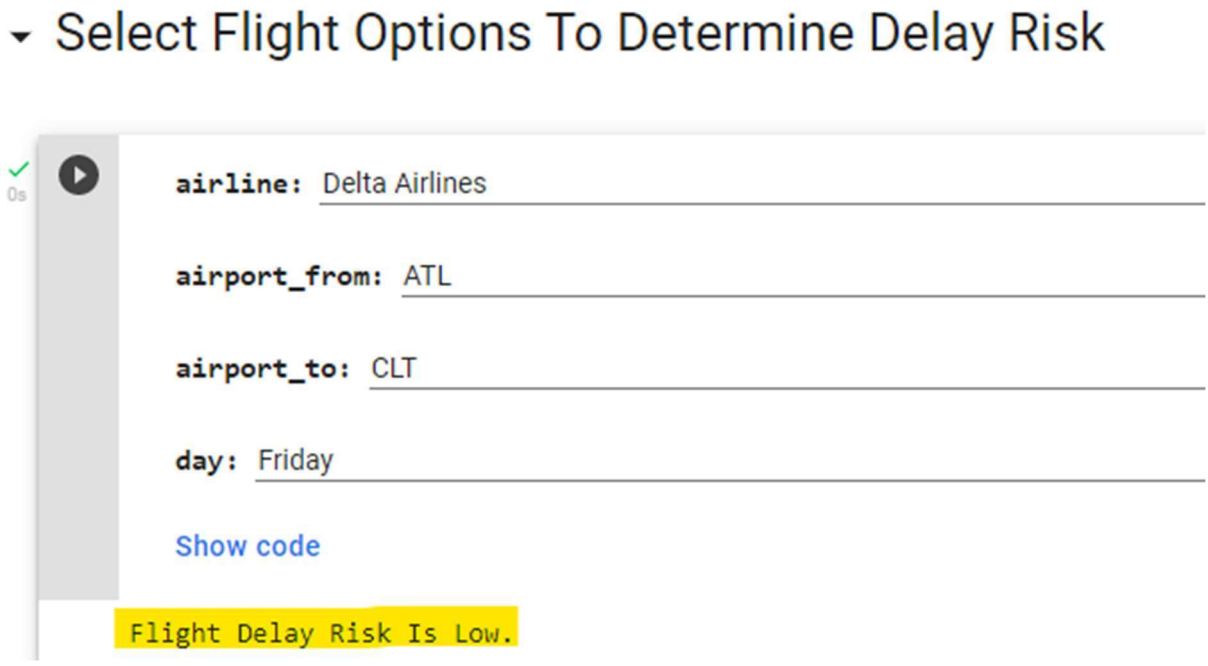
18. The following bar chart should now be displayed.



19. Next, select an **airline, airport_from**, **airport_to**, and **day** option from the dropdown lists of the form below the "**Select Flight Options To Determine Delay Risk**" heading.



20. Finally, after selecting your flight information values, click the play button located next to the form. *Please note: Moving forward within the same session, you may run the prediction on as many flights as you would like without needing to re-run the prior code blocks. * Once completed, you should see an output message below the form indicating whether the flight information provided is likely to be delayed. An example output is provided below.



Summation of Learning Experience

This capstone project has both challenged me and provided an excellent opportunity to gain exposure to machine learning in a hands-on manner. Previous programming courses such as Scripting & Programming – Foundations, Scripting & Programming – Applications, Data Structures & Algorithms I & II, Software I & II, and Introduction to Artificial Intelligence have helped prepare me for the coding aspect of the capstone project. Discrete Mathematics I & II and Introduction to Probability & Statics have also equipped me with the logical and statistical foundations necessary to complete the project. Courses such as Software Engineering and Software Quality Assurance have given me an understanding of quality methods and adequate exposure to the SLDC necessary to produce a viable application. Business of IT – Project Management has solidified my understanding of the project planning tasks necessary for the development of a quality product. Additionally, Introduction to Communication and Technical Communication have prepared me for the level of writing required for the written portion of the capstone project. Thanks to the quality education provided by WGU, I was able to make a career transition within a year of starting the CS program into the software development field. The experience and knowledge I have gained over the past 3+ years professionally were also instrumental to my ability to complete the capstone project. My time in the CS program at WGU has instilled a lifelong passion for continuous learning and growth. I will remain forever grateful to all the excellent course instructors, evaluators, and program mentors I have had the opportunity to encounter along the way. I have no doubt that it is, essentially, the culmination of all these reasons that led me here. Go night owls!

Sources

No outside sources were directly quoted or paraphrased.