PREDICTING RISK OF INJURY PER SHIP

Ship Happens, Cohort-7
Data Science Certificate Program
Georgetown University

Aaron Wright
Brian Fan
Kayla Hinrichs
Sudha Goel
Sushanta Paul

ABSTRACT

Using ship characteristics and historical data from various data stores, we have created a model that can determine the (binary) risk of a ship with certain select attributes such as size, age, and ship type. The project goal for Ship Happens was to determine whether a ship, with specified attributes, was at risk of a negative event occurring (collision, pollution event, injury). Publicly available data from the Marine Information for Safety and Law Enforcement (MISLE) and the Automated Identification System (AIS) provided thorough positional information about specific ships traveling in United States waters as well as a data store of recorded negative ship-based events. Our team utilized these resources to ingest and wrangle the data into a tested and trained model that could determine if a ship with various characteristics was at risk for an incident. The results of this model-based prediction could aid with policy development, logistics planning, asset allocation, and maritime law enforcement efforts while increasing efficiency and reducing risk to personnel.

Contents

1 Introduction

The primary goal of Team Ship Happens was to determine whether there was a relationship between a ship's characteristics and rate of injury incidents using historical data. We wanted to create a model that would be able to predict areas of risk as well as assign risk profiles based on the ship type, location of operations, and the time of year. The final product of this model-based prediction could aid with policy development and logistical planning to support future Search & Rescue operations.

Publicly available data, Automated Identification System (AIS) and the Marine Information for Safety and Law Enforcement (MISLE) was used for this project. Other data sources mainly time, location, weather, specific-operator, and environmental data were considered, but because of storage concerns and time constraints, these datasets were not used for the final project. Initially, several months of AIS data in addition to the MISLE data was ingested, but there were issues overlaying the AIS posit data with the MISLE data due to inconsistencies with geolocation coding. Furthermore, the AIS data store was simply too vast to be digestible with our current machines. AIS data separates the globe into zones and each zone has data for each month with each month totaling millions of rows of posit data. Complicating our data further was an inability to gain valuable insights with the AIS data. MISLE data specifically tracks ships that enter the United States waters, but AIS data tracks every single ship that has a transponder. If we were to constrain our AIS data to a specific location, e.g. the waters around Hawaii, we would only have around eight MISLE instances that are unusable for a model. In addition to these shortcomings, without knowing specific ship routes would be unable to assign risk probabilities to high-risk areas correlated to the MISLE data.

After considering the difficulties associated with incorporating AIS data, we decided to focus on the Marine casualty data located within the MISLE database. Using this data, we would attempt to create a model that could accurately classify and predict the risk of an accident using historical data and specific ship characteristics.

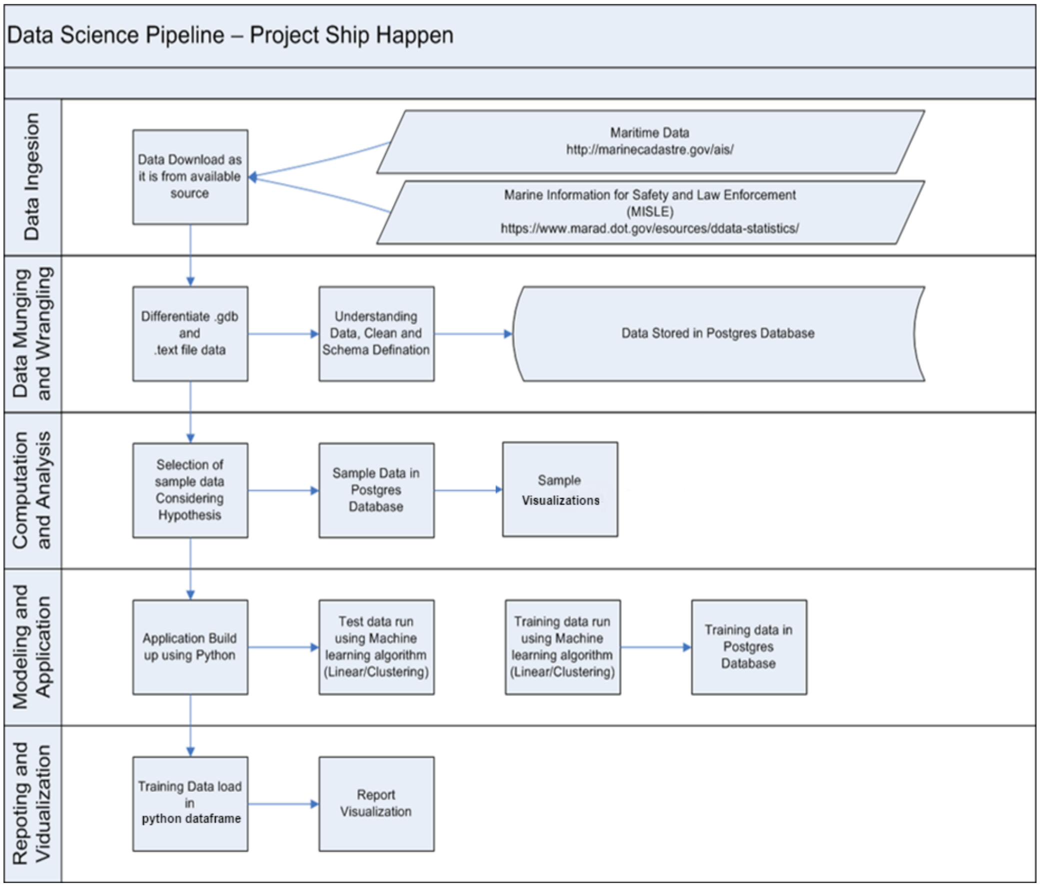
2 Statement of Hypothesis

A model can determine if there is sufficient correlation between ship characteristics and instances of incidents from historical data in order to predict risk of injury per ship for the future. Successfully doing so may inform asset allocation, enforcement efforts, and can aid with policy planning, while reducing risk to personnel and individuals.

3 Methods

3.1 Data Science Pipeline

We utilized a relatively straightforward pipeline process that is depicted in the following image.


Data Science Pipeline – Project Ship Happen
Data Ingesion
Data Download as it is from available source
Maritime Data http://marinecadastre.gov/ais/
Marine Information for Safety and Law Enforcement (MISLE) https://www.marad.dot.gov/esources/ddata-statistics/
Data Munging and Wrangling
Differentiate .gdb and .text file data
Understanding Data, Clean and Schema Defination
Data Stored in Postgres Database
Computation and Analysis
Selection of sample data Considering Hypothesis
Sample Data in Postgres Database
Sample Visualizations
Modeling and Application
Application Build up using Python
Test data run using Machine learning algorithm (Linear/Clustering)
Training data run using Machine learning algorithm (Linear/Clustering)
Training data in Postgres Database
Repoting and Vidualization
Training Data load in python dataframe
Report Visualization

3.2 Data Ingestion and Wrangling

Data was ingested from the Marine Casualty and Pollution Database from the Marine Information for Safety and Law Enforcement (MISLE) site and Automated Identification System) Data. We ingested raw data into ArcGIS to visualize the information provided by AIS. For each month of each year for each time zone (UTM) within the US, there were tens of millions of ship posits (instances) resulting in a huge volume of data which we were unable to use. To test our model with "real world" data, we did use live-streaming AIS, but a very small volume in comparison to the original AIS dataset.

The second dataset used is the MISLE data. The data reflected information collected by U.S. Coast Guard personnel concerning vessel and waterfront facility accidents and marine pollution incidents throughout the United States and its territories. This dataset contains tables related to vessels, injury, pollution and vessel activity. Our team was planning to use longitude and latitude coordinates included in MISLE table to find out if there was any specific location more prone to incidents. However, due to lack of expertise in handling GIS data and time constraints, we had to drop this idea.

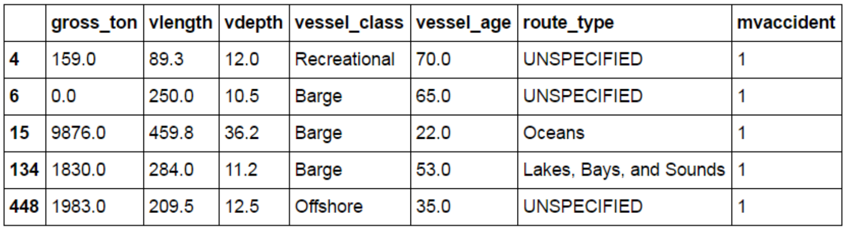
In the end, we used three tables: MisleVessel, MisleActivity, and MisleInjury. The MisleVessel table was joined to MisleActivity and MisleInjury using a left outer join inclusive of the predetermined columns. The result of this join yielded 1,353,830 instances and 16 features after removing the duplicate records. Regression techniques and transformation methods were applied to this dataset and because of this, 260,474 instances and 7 features were selected for the final dataset. Our target variable was ship accident, a binary value, to build our prediction model.

Attributes used: gross_ton, vlength, vdepth, vessel_class, vessel_age, route_type.
Target Variable: Marine Accident (Yes/No).

3.3 Computation and Analysis

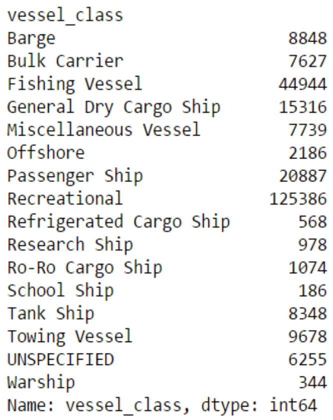
MISLE and MISLE Marine Casualty data were used for the analysis and computation portion of this project. We used a Jupyter Notebook to break down and understand our data in order to find out which features were appropriate for modeling and prediction purposes.

Our Dataset with Selected Features (Sample):



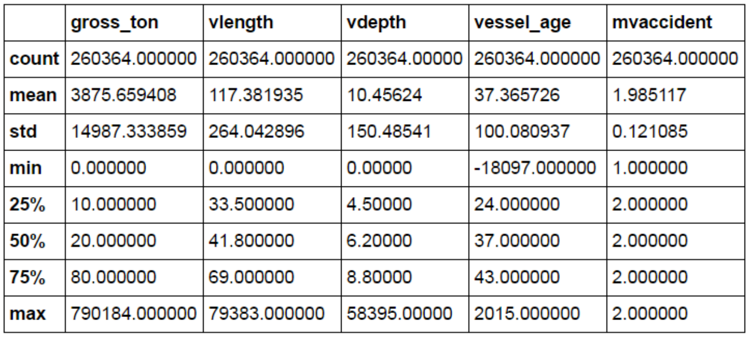
	gross_ton	vlength	vdepth	vessel_class	vessel_age	route_type	mvaccident
4	159.0	89.3	12.0	Recreational	70.0	UNSPECIFIED	1
6	0.0	250.0	10.5	Barge	65.0	UNSPECIFIED	1
15	9876.0	459.8	36.2	Barge	22.0	Oceans	1
134	1830.0	284.0	11.2	Barge	53.0	Lakes, Bays, and Sounds	1
448	1983.0	209.5	12.5	Offshore	35.0	UNSPECIFIED	1

Types of Vessel class with count:



```
vessel_class
Barge                       8848
Bulk Carrier                7627
Fishing Vessel             44944
General Dry Cargo Ship     15316
Miscellaneous Vessel        7739
Offshore                    2186
Passenger Ship             20887
Recreational              125386
Refrigerated Cargo Ship      568
Research Ship                978
Ro-Ro Cargo Ship            1074
School Ship                  186
Tank Ship                   8348
Towing Vessel               9678
UNSPECIFIED                 6255
Warship                      344
Name: vessel_class, dtype: int64
```

Statistical Information about Dataset:



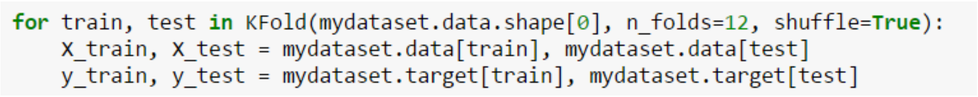
	gross_ton	vlength	vdepth	vessel_age	mvaccident
count	260364.000000	260364.000000	260364.00000	260364.000000	260364.000000
mean	3875.659408	117.381935	10.45624	37.365726	1.985117
std	14987.333859	264.042896	150.48541	100.080937	0.121085
min	0.000000	0.000000	0.00000	-18097.000000	1.000000
25%	10.000000	33.500000	4.50000	24.000000	2.000000
50%	20.000000	41.800000	6.20000	37.000000	2.000000
75%	80.000000	69.000000	8.80000	43.000000	2.000000
max	790184.000000	79383.000000	58395.00000	2015.000000	2.000000

3.4 Model Building

Our target variable was a yes/no category correlated to seven selected attributes. We used multiple classification techniques using the Scikit-Learn Machine Learning Library to build our model.

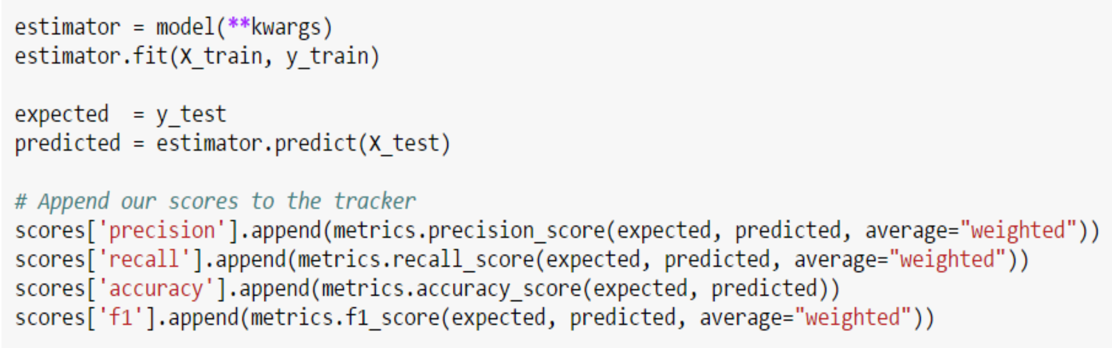
Feature selection: We used the following Regularization techniques and Transformation methods to select the most relevant features: LASSO (L1 Regularization), Ridge Regression (L2 Regularization), and ElasticNet.

Training and Test Dataset: A 12 KFold Cross Validation method was used to get our training and testing dataset. We created a bunch for our metadata and dataset.



```
for train, test in KFold(mydataset.data.shape[0], n_folds=12, shuffle=True):
    X_train, X_test = mydataset.data[train], mydataset.data[test]
    y_train, y_test = mydataset.target[train], mydataset.target[test]
```

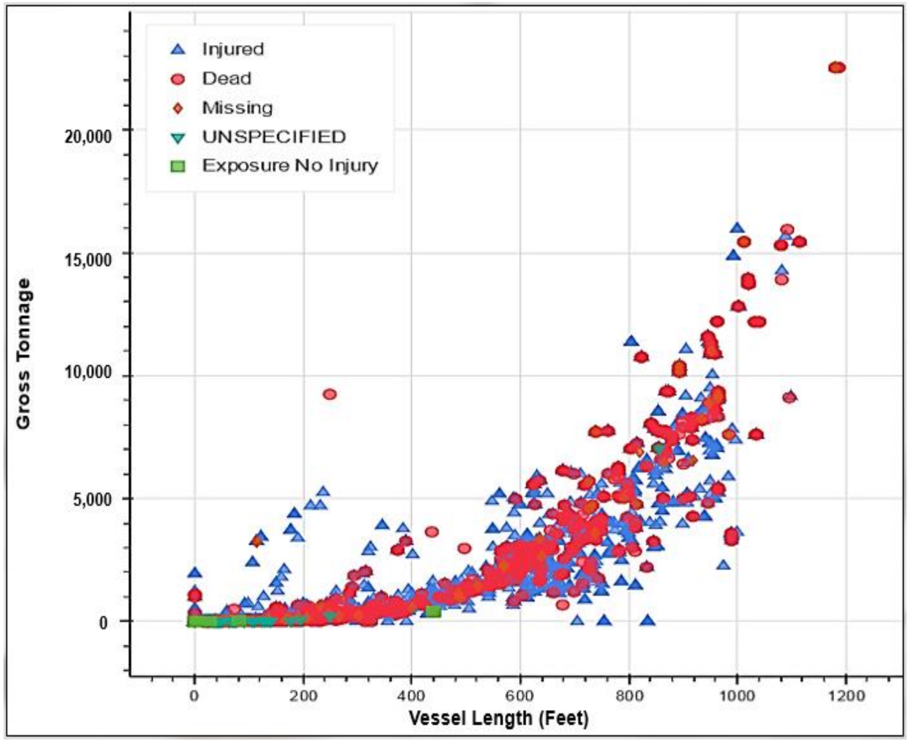
Model Fit and Evaluation: We trained our model with our training dataset to build our predictive model. To select the right model, we tried fitting our model using several classifiers.



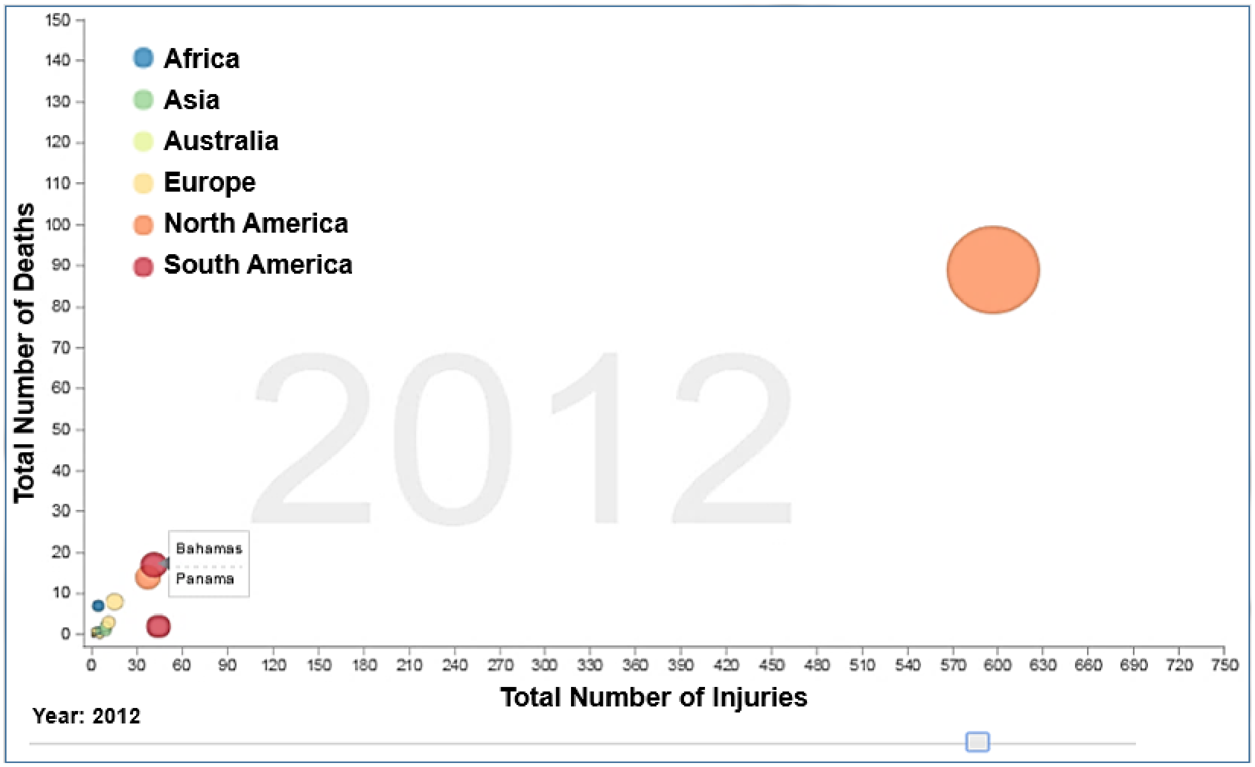
```
estimator = model(**kwargs)
estimator.fit(X_train, y_train)

expected  = y_test
predicted = estimator.predict(X_test)

# Append our scores to the tracker
scores['precision'].append(metrics.precision_score(expected, predicted, average="weighted"))
scores['recall'].append(metrics.recall_score(expected, predicted, average="weighted"))
scores['accuracy'].append(metrics.accuracy_score(expected, predicted))
scores['f1'].append(metrics.f1_score(expected, predicted, average="weighted"))
```

3.5 Visualizations

We utilized the following libraries for visualization purposes: Matplot Library, Seaborn, and Bokeh. The chart below indicates that ships of medium to large size are safer than smaller vessels. However, this could simply be a bias of our data as the dataset is limited to ships in the United States and a large portion of these are small recreational ships.



Using Bokeh, we created a time-series visualization that shows the number of injuries per year, per country.

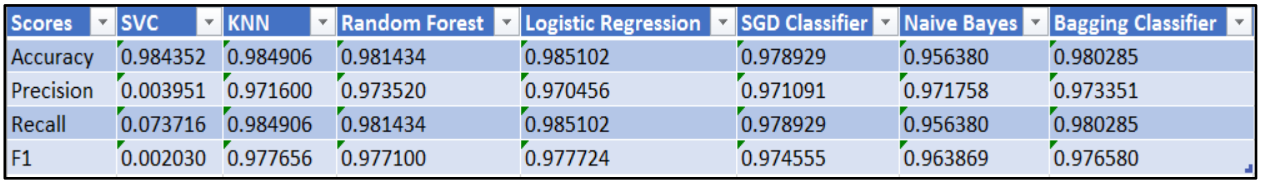


4 Outcomes

After building our model, we did performance testing using several classifiers. Comparing scores for precision, accuracy, f1, etc., we selected KNN and Random Forest to run our test data and actual data from the Port of Baltimore.

4.1 Model Scoring

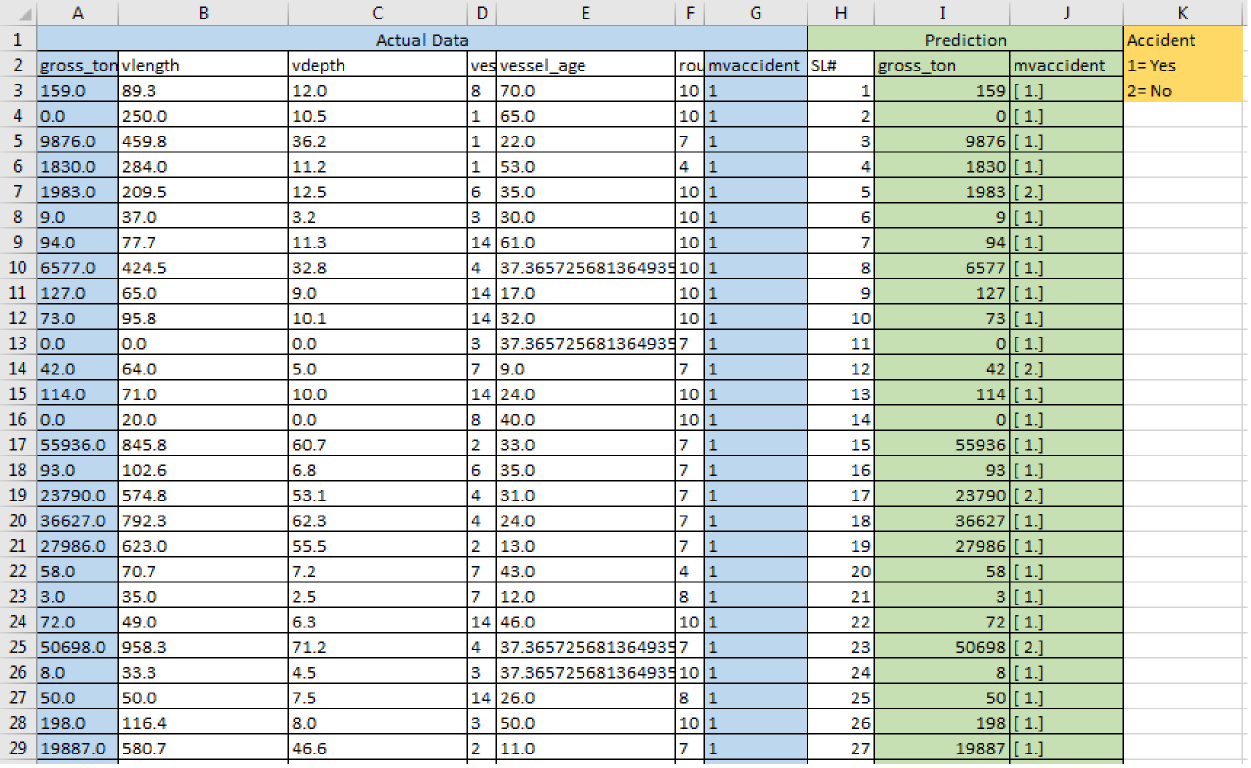
We used the following classifiers for Model Performance Testing: SVC, KNN, RandomForest, Logistic Regression, SGD Classifier, Naïve Bayes, and Bagging Classifier



Scores	SVC	KNN	Random Forest	Logistic Regression	SGD Classifier	Naive Bayes	Bagging Classifier
Accuracy	0.984352	0.984906	0.981434	0.985102	0.978929	0.956380	0.980285
Precision	0.003951	0.971600	0.973520	0.970456	0.971091	0.971758	0.973351
Recall	0.073716	0.984906	0.981434	0.985102	0.978929	0.956380	0.980285
F1	0.002030	0.977656	0.977100	0.977724	0.974555	0.963869	0.976580

4.2 Results Comparison

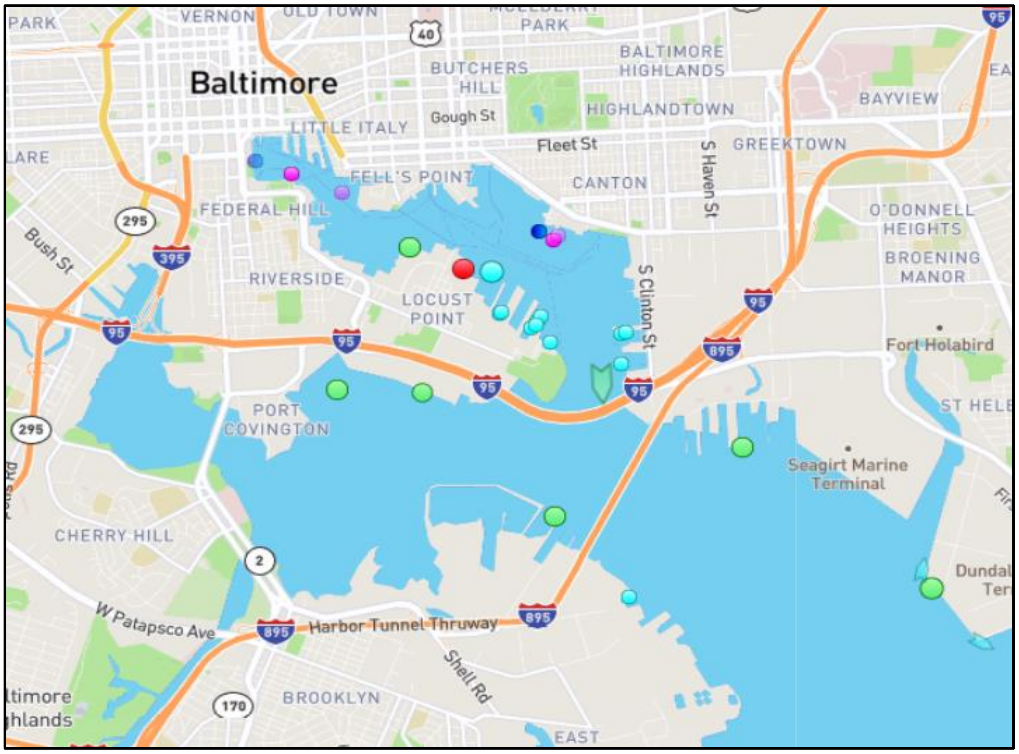
After which, we ran our test data using a random forest and procured the results below comparing the actual data (blue) to the predicted data (green).



	A	B	C	D	E	F	G	H	I	J	K
1	Actual Data							Prediction			Accident
2	gross_ton	vlength	vdepth	ves	vessel_age	rou	mvaccident	SL#	gross_ton	mvaccident	1= Yes
3	159.0	89.3	12.0	8	70.0	10	1	1	159	[1.]	2= No
4	0.0	250.0	10.5	1	65.0	10	1	2	0	[1.]	
5	9876.0	459.8	36.2	1	22.0	7	1	3	9876	[1.]	
6	1830.0	284.0	11.2	1	53.0	4	1	4	1830	[1.]	
7	1983.0	209.5	12.5	6	35.0	10	1	5	1983	[2.]	
8	9.0	37.0	3.2	3	30.0	10	1	6	9	[1.]	
9	94.0	77.7	11.3	14	61.0	10	1	7	94	[1.]	
10	6577.0	424.5	32.8	4	37.365725681364935	10	1	8	6577	[1.]	
11	127.0	65.0	9.0	14	17.0	10	1	9	127	[1.]	
12	73.0	95.8	10.1	14	32.0	10	1	10	73	[1.]	
13	0.0	0.0	0.0	3	37.365725681364935	7	1	11	0	[1.]	
14	42.0	64.0	5.0	7	9.0	7	1	12	42	[2.]	
15	114.0	71.0	10.0	14	24.0	10	1	13	114	[1.]	
16	0.0	20.0	0.0	8	40.0	10	1	14	0	[1.]	
17	55936.0	845.8	60.7	2	33.0	7	1	15	55936	[1.]	
18	93.0	102.6	6.8	6	35.0	7	1	16	93	[1.]	
19	23790.0	574.8	53.1	4	31.0	7	1	17	23790	[2.]	
20	36627.0	792.3	62.3	4	24.0	7	1	18	36627	[1.]	
21	27986.0	623.0	55.5	2	13.0	7	1	19	27986	[1.]	
22	58.0	70.7	7.2	7	43.0	4	1	20	58	[1.]	
23	3.0	35.0	2.5	7	12.0	8	1	21	3	[1.]	
24	72.0	49.0	6.3	14	46.0	10	1	22	72	[1.]	
25	50698.0	958.3	71.2	4	37.365725681364935	7	1	23	50698	[2.]	
26	8.0	33.3	4.5	3	37.365725681364935	10	1	24	8	[1.]	
27	50.0	50.0	7.5	14	26.0	8	1	25	50	[1.]	
28	198.0	116.4	8.0	3	50.0	10	1	26	198	[1.]	
29	19887.0	580.7	46.6	2	11.0	7	1	27	19887	[1.]	

5 Application

After testing with our training data, we decided to test our model with a real-world application. We ran our model with real data from the Port of Baltimore to see if any ships were predicted to be high risk. The Common Operational Picture (COP) on 14 January 2017 provided live streaming ship positional data for 19 commercial ships that are depicted in the image below as various geometric markers.



Out of the 19 ships in the Port of Baltimore, 0 were determined high risk by our model. From this model prediction, we can posit the following:

- The Port of Baltimore is possibly a low-risk port
- There could be underreporting of incidents in the Port of Baltimore
- Our build model has weaknesses

Ideally, operators at a port could use this information as a type of tipping and queuing for operational day-to-day planning, or year-to-year future planning. Lives could be saved if regulatory changes address factors influencing ships with the greatest level of risk.

6 Limitations

We had many limitations in terms of time and complexity of the AIS data, in addition to storage issues due to volume. After deciding to focus on MISLE marine casualty data, we ended up with missing data in our instances as well as only have a very small percentage of ships having ever had an accident. This ultimately contributed to weaknesses in our model and influenced our confidence in the prediction results. Furthermore, with our limited knowledge we had difficulties implementing ArcGIS, QGIS and geolocation data into our model and product.

7 Future Enhancement

Even though we started with three types of ship incident data - casualties, pollution events, events (other incidents), we ultimately only used the casualty data for our project. Due to complexity and time constraints, we were unable to incorporate all incident data or the AIS data. Having locational data would have been useful as marine authorities would be able to better allocate resources in high-traffic, high-incident locations based on a more complete model.

8 Conclusion

Ship Happens ended up using MISLE casualty data to build a classifying model to predict risk of injury per ship. Despite significant limitations, our model was successful and we were able to test it against real-world data from the Port of Baltimore. We ended up with 19 ships with enough data to input into our model and the model deemed none of the ships to be at risk of an accident. Of course, our model was trained on a dataset that mostly consisted of ships without incidents; less than 1% of ships in our dataset had an accident. We had not compensated for this; however, in the real world the chance of having an accident on a ship is also relatively low. If we were to continue working on this problem issue, we would incorporate the rest of the MISLE data and geolocation data as well as weather, time, and operator data in order to create a better predictive model.

9 References

Kirk, Matthew (2014), Thoughtful Machine Learning. O'Reilly Media. Print.

McKinney, Wes (2012), Python for Data Analysis. O’Reilly Media. Print.

Downey, Allen B (2013), Think Python. O’Reilly Media. Print.

Segaran, Toby (2008), Programming Collective Intelligence. O’Reilly Media. Print

Pedregosa et al. (2011), Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830.

Harris et al. (2013), Analyzing the Analyzers. O’Reilly Media.

Wickham, H. Cook, D. and Hofmann, H. (2015), Visualizing statistical models: Removing the blindfold. Statistical Analysis and Data Mining: The ASA Data Science Journal, 8; 203-225.

Heer, J., Bostock, M., Ogievetsky, V. (2010), A tour through the visualization zoo. ACM Queue, 8:5; 1-22.

Rosling, H. (2006), The best stats you’ve ever seen. TED Talk.

Tarleton Gillespie et al. (2014), Media technologies: Essays on communication, materiality, and society. The MIT Press.

User Guide. Scikit-Learn-learn 0.17.1 Documentation. Web.