IML Hackathon

Preprocessing and EDA (Exploratory Data Analysis)

As we approached the data, we saw that all samples are valid, without noticeable corrupted values or empty rows.

We removed the following features: '*nComments*', '*linqmap_expectedEndDate*', '*linqmap_expectedBeginDate*', '*linqmap_reportMood*', '*Linqmap_nearby*' '*linqmap_reportDescription*', since they were all mostly empty \ zeros. We also removed the 'OBJECTID' feature, since it is not relevant for predicting.
We turned the '*linqmap_reportRating*' feature to an indicator, indicating whether the event is highly ranked (3 or more) or not, since we saw 96% of the events are ranked 0 or 5. Changed the name to '*is_highly_rated*' to be indicative.
We created dummies (one-hot-vector) from the following features: '*Linqmap_type*', '*Linqmap_subtype*', '*linqmap_roadType*'.
New features we added: is_weekend, weekday (day of the week), *relative_time* (minutes since midnight), season, window_1 (indicator 08:00-10:00), window_2 (indicator 12:00-14:00), window_3 (indicator 18:00-20:00).

Features we wrote but didn't use: is_holiday, Dis_from_center_tlv.

New feature driven from x, y coordinates:
For the first task, we decided to focus mainly on the relationship between the four samples in each group of samples. We split the data into disjoint sequential groups(by ubdate_date). Then, we combined the four samples into one sample containing all feature. Then we created the new 5 features *d1, d2, d3 ,x_centroid, y_centroid* as explained below:

1. *d1* - distance between the first and second sample in the group of 4 samples.
2. *d2* - distance between the second and third sample in the group of 4 samples.
3. *d3* - the distance between the third and fourth sample in the group of 4 samples.
4. *x_centroid* := $\sqrt{{x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2}$ - the center of x coordinate.
5. *y_centroid* := $\sqrt{{y_1}^2 + {y_2}^2 + {y_3}^2 + {y_4}^2}$ - the center of y coordinate.

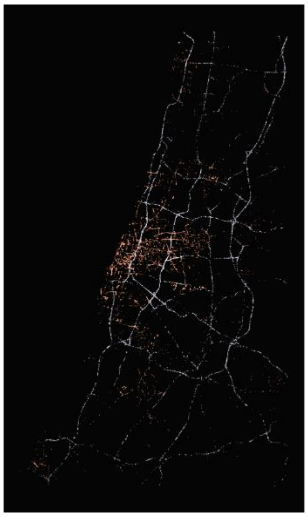


Figure 1: Location of samples

We noticed almost half of the data (7154 samples) has no 'linqmap_city' value, but has 'linqmap_street' value. We suspected that all empty city values indicate a highway, and after plotting the 'x' and 'y' coordinates, we could see it really is the situation:

For us, the most confusing thing about the data was the dates - we couldn’t understand the relation between 'Update_date' and ‘pubDate’, Only later we realized we should only use the 'Update_date' feature.
We found that the feature 'Update_date' contains only 5 distinct days (15-18.5.22, 24.5.22), while 'pubDate' contains 50 distinct dates. We also found that events that their ‘linqmap_type’ value is not ROAD_CLOSED and ‘linqmap_subtype’ value is not HAZARD_ON_ROAD_CONSTRUCTION are also within the 5 days above, so it made sense.

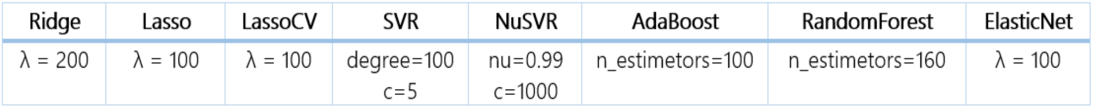
We used the library Pandas Profiling to create a clear and visual overview of the data. It helped us in the preprocessing and the EDA. You are welcome to open "waze_raw_data_profile.html".

First Task

For the baseline model we used Linear Regression, since it is the most common and simple model we could think of, and only used the 4 predicted features as train

In the Model Selection stage, we have tried the following algorithms:

LinearRegression, Ridge Regression, Lasso Regression, LassoCV, SVR Regression, NuSVR, AdaBoostRegressor, RandomForestRegressor, ElasticNet, SGD Regression.



Ridge	Lasso	LassoCV	SVR	NuSVR	AdaBoost	RandomForest	ElasticNet
λ = 200	λ = 100	λ = 100	degree=100 c=5	nu=0.99 c=1000	n_estimetors=100	n_estimetors=160	λ = 100

Figure 2: Hyperparameters selected

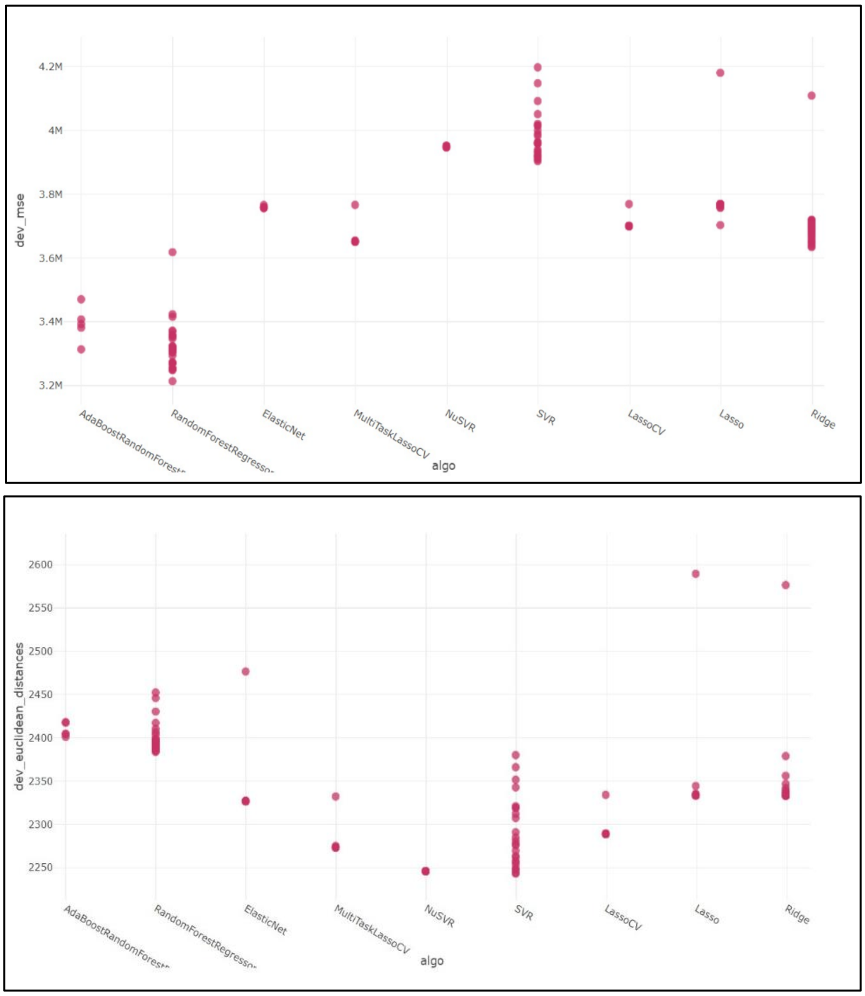


Figure 3: Hyperpramters performance Comparison (in mlFlow)

We have used MLflow library to choose these specific Hyper-parameters. In all the above regression algorithms we used MultiOutputRegressor [1] to make a multi labeled regression on x and y of the fifth example. To evaluate our regression, we used MSE and the mean Euclidean distance between our prediction and the true values of x and y (as instructed).

Eventually, we chose to use the SVR (Support Vector Regression) model because it received the lowest mean Euclidean distance over the validation test.

MSE over test: 3734589.2415766786

Euclidean distance over test: 2231.790585185445

MSE over train: 3835508.9475474292

Euclidean distance over train: 2253.7083560298165

https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html [1]

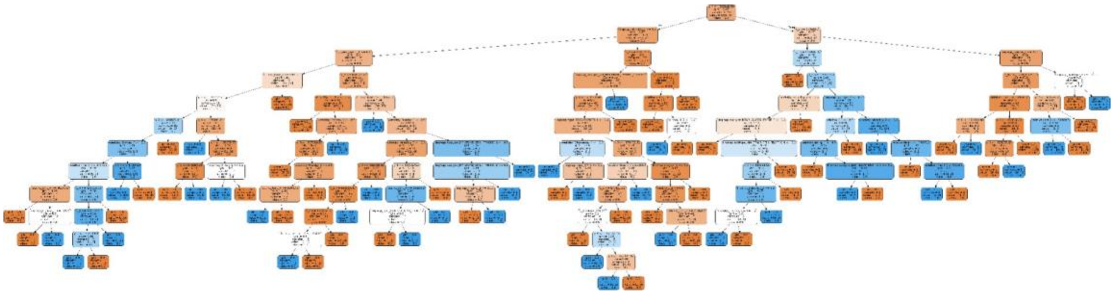


Figure 4: Example of one of the Random Forest Trees visualizations we got

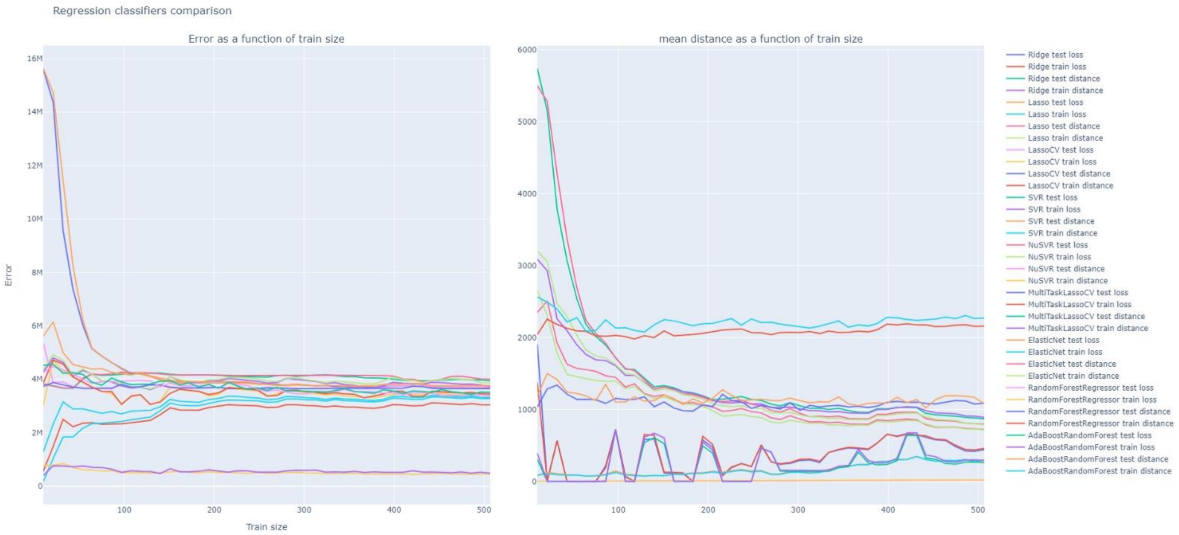
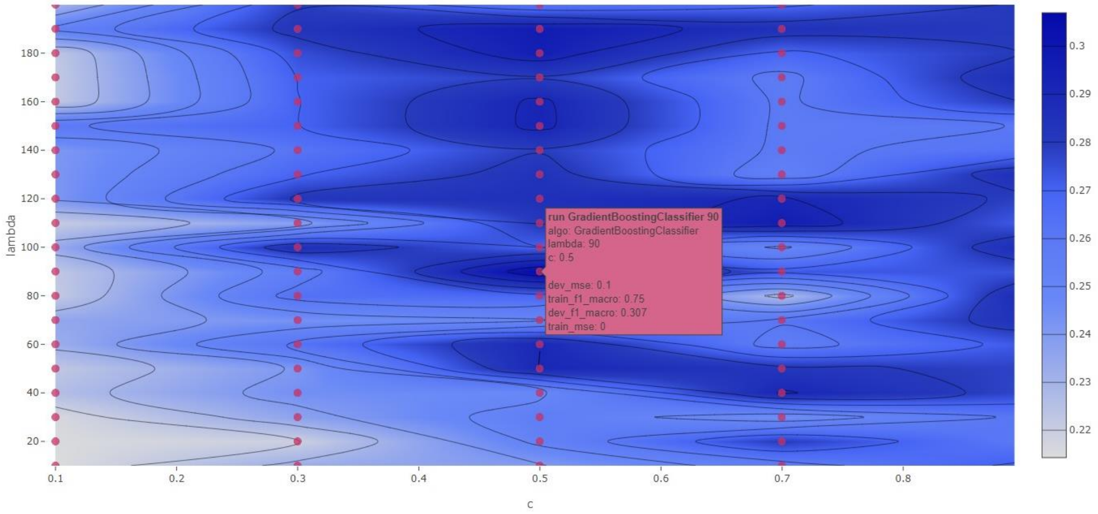


Figure 5: Comparison between all different optimized regression models



run GradientBoostingClassifier 90
algo: GradientBoostingClassifier
lambda: 90
c: 0.5

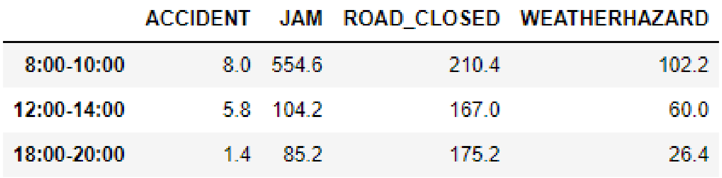
dev_mse: 0.1
train_f1_macro: 0.75
dev_f1_macro: 0.307
train_mse: 0

Figure 6: Hyperparameter choosing for the classification model

* These are some of the graphs we got from the mlFlow interactive comparison window, we didn't upload it due to lack of space.

Second task

For the Baseline estimator, we naively calculated the average for every time slot and event, for each of the days, using all the data we were given (without learning algorithm). Since it is general, these are the results for all days:



	ACCIDENT	JAM	ROAD_CLOSED	WEATHERHAZARD
8:00-10:00	8.0	554.6	210.4	102.2
12:00-14:00	5.8	104.2	167.0	60.0
18:00-20:00	1.4	85.2	175.2	26.4

Figure 7: Baseline prediction

After exploring the data we were given and realizing we only have 5 days to learn from (in update_date), we chose to do statistical learning, and calculate the average with some adjustments. We read online and tried to modify the results according to real-world statistics. We are expected to predict 5.6.22, which is Shavuot holiday, and like Saturday, religiously driving is forbidden. In an article we found from 2021, the traffic volume on Saturdays is 64% more than regular days[2].

In another article, we found data about the average traffic volume in different days, as you can see in the graph below[3]:

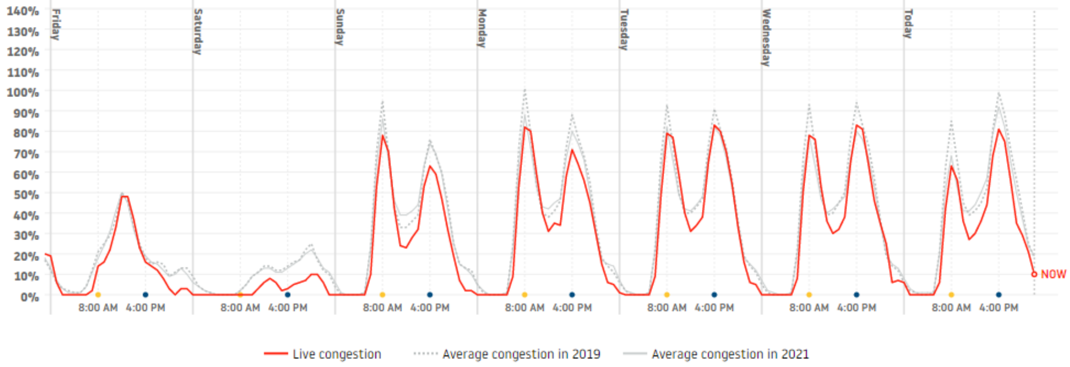
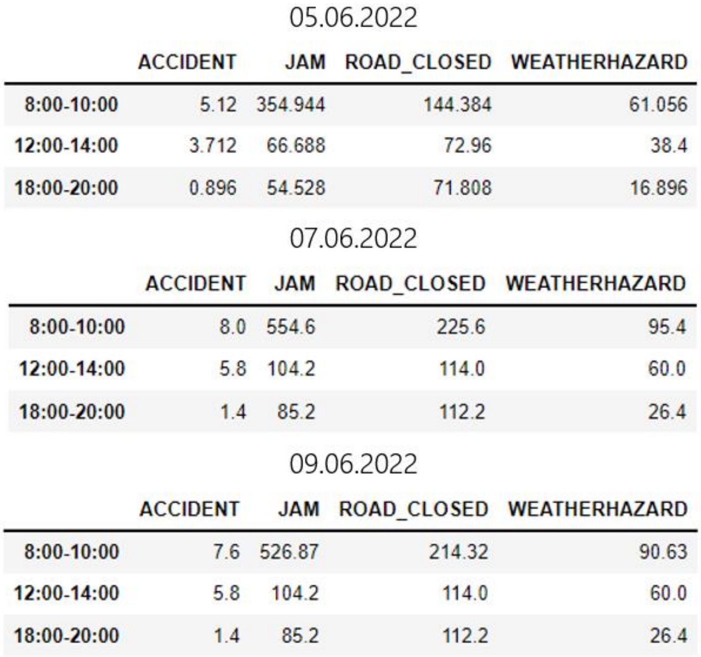


Figure 8: Average traffic volume in Israel, by day

On Thursday (marked a "Today"), we can see that the average traffic in the time slot 8:00-10:00 is about 5% lower than the rest of the work week. Also, even though the traffic in the afternoon is higher, in the time slot 18:00-20:00 the traffic is average. Unsurprisingly, our final predictions were quite similar to the base model.

[2] https://www.ynet.co.il/news/article/Hkf333Ni00

[3] /https://www.tomtom.com/en_gb/traffic-index/tel-aviv-traffic



05.06.2022

	ACCIDENT	JAM	ROAD_CLOSED	WEATHERHAZARD
8:00-10:00	5.12	354.944	144.384	61.056
12:00-14:00	3.712	66.688	72.96	38.4
18:00-20:00	0.896	54.528	71.808	16.896

07.06.2022

	ACCIDENT	JAM	ROAD_CLOSED	WEATHERHAZARD
8:00-10:00	8.0	554.6	225.6	95.4
12:00-14:00	5.8	104.2	114.0	60.0
18:00-20:00	1.4	85.2	112.2	26.4

09.06.2022

	ACCIDENT	JAM	ROAD_CLOSED	WEATHERHAZARD
8:00-10:00	7.6	526.87	214.32	90.63
12:00-14:00	5.8	104.2	114.0	60.0
18:00-20:00	1.4	85.2	112.2	26.4

Figure 9: Second task's predictions