DataRobot

DataRobot의 위치정보를 활용한 모델링

(DataRobot's Location AI, ease of featurizing)

DataRobot

Agenda

1. Geospatial Data
2. DataRobot Location AI
3. Spatial Feature Engineering
4. Demo
5. Other use cases

GeoSpatial Data

What is geospatial data?

- Geospatial data is data with a geographic component. Think data like latitude/longitude coordinates and satellite imagery.
- We can use this data in its raw format in cases like satellite imagery, or use it to derive new features to include in modeling.
- **Ignoring the spatial component** of our data means we're ignoring potentially valuable information.

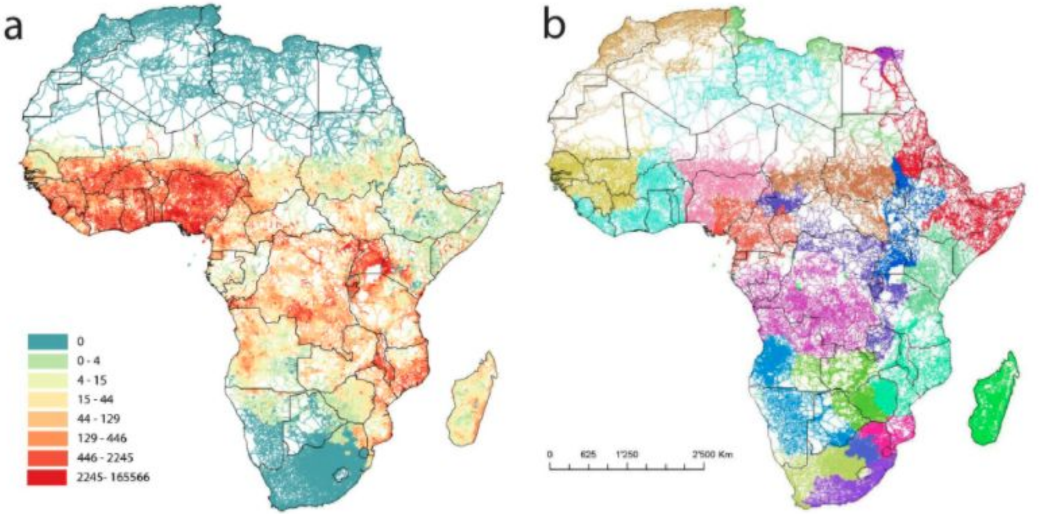
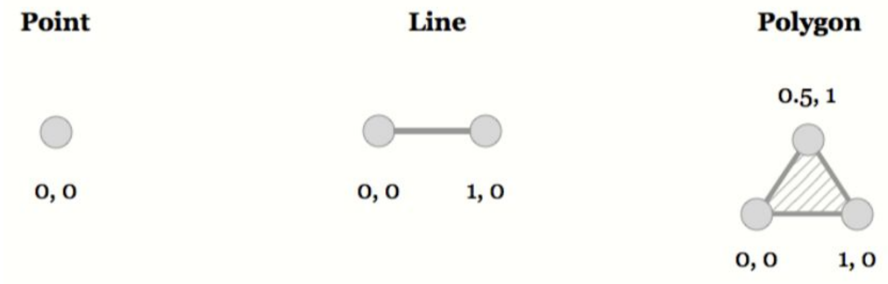


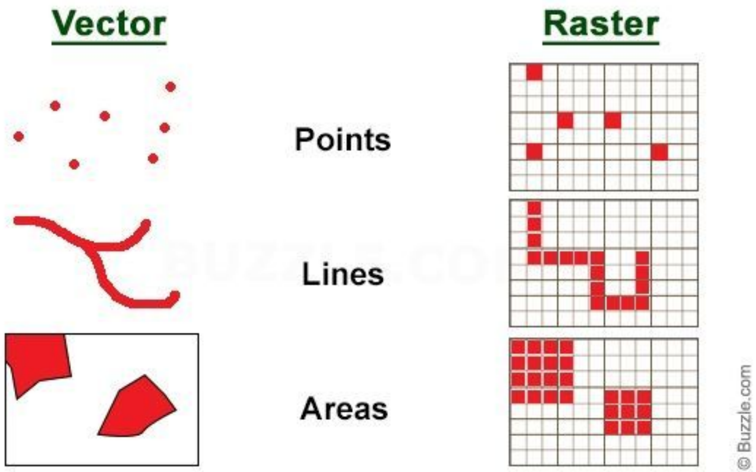
Figure 5. Data and example outputs for *Plasmodium falciparum* malaria weighted road network analyses. (a) Africa road network with each road segment coloured by its maximum value of *P.falciparum* prevalence multiplied by population. (b) Output of community detection on the data in (a), showing the result for 20 communities.

A classic example is analyzing the relationship between disease spread in Africa and road network density.

Vectors

- A vector is a geometry - either a point, line, or polygon
- **Features** with discrete shapes or boundaries like roads or administrative regions
- Easy to tie metadata to or **derive features** from
- This is what our Location AI uses





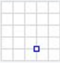
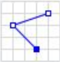
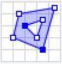
Formats

Only 5 you need to know for DataRobot. All are designed to store vector data

- **ESRI Shapefile**
  - Most common format
  - Multifile
  - DR requires a zipped folder with .shp, .shx, .dbf, and .prj files.
- **GeoJSON (RFC7946)**
  - Single file
  - As name suggests, stores vector information in a json format
- **ESRI File Geodatabase**
  - Approximates a database through a nested folder structure
  - DR takes a zipped .gdb and reads the first layer
- **Well Known Text**
  - Way of representing vector data within a column in a csv or something similar
  - e.g. POINT (-116.0, 32.74)
- **PostGIS**
  - PostgreSQL but for GIS data

GeoJSON

Geometry primitives

Type	Examples	
Point		`{ "type": "Point", "coordinates": [30, 10] }`
LineString		`{ "type": "LineString", "coordinates": [ [30, 10], [10, 30], [40, 40] ] }`
Polygon		`{ "type": "Polygon", "coordinates": [ [[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]] ] }`
		`{ "type": "Polygon", "coordinates": [ [[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]] ] }`

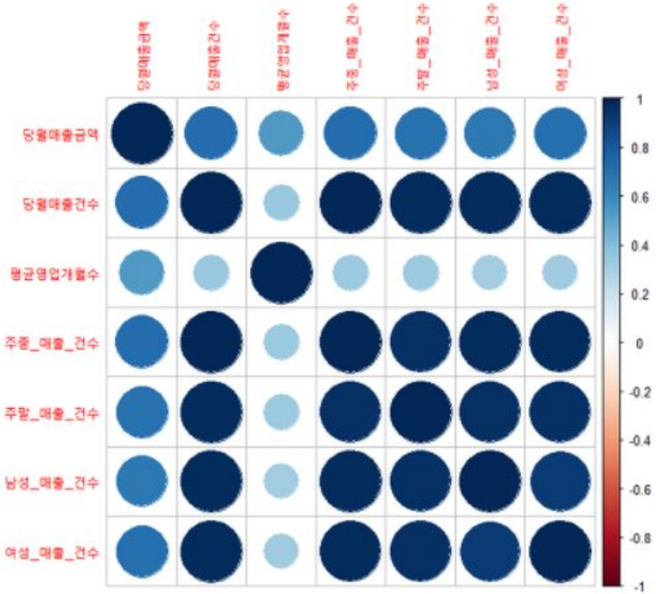
MultiPolygon		`{ "type": "MultiPolygon", "coordinates": [ [ [[30, 20], [45, 40], [10, 40], [30, 20]] ], [ [[15, 5], [40, 10], [10, 20], [5, 10], [15, 5]] ] ] }`
		`{ "type": "MultiPolygon", "coordinates": [ [ [[40, 40], [20, 45], [45, 30], [40, 40]] ], [ [[20, 35], [10, 30], [10, 10], [30, 5], [45, 20], [20, 35]], [[30, 20], [20, 15], [20, 25], [30, 20]] ] ] }`
GeometryCollection		`{ "type": "GeometryCollection", "geometries": [ { "type": "Point", "coordinates": [40, 10] }, { "type": "LineString", "coordinates": [ [10, 10], [20, 20], [10, 40] ] }, { "type": "Polygon", "coordinates": [ [[40, 40], [20, 45], [45, 30], [40, 40]] ] } ] }`

https://en.wikipedia.org/wiki/GeoJSON

Use cases in R : 상권 분석, Correlation

서울시 골목상권 Profiling

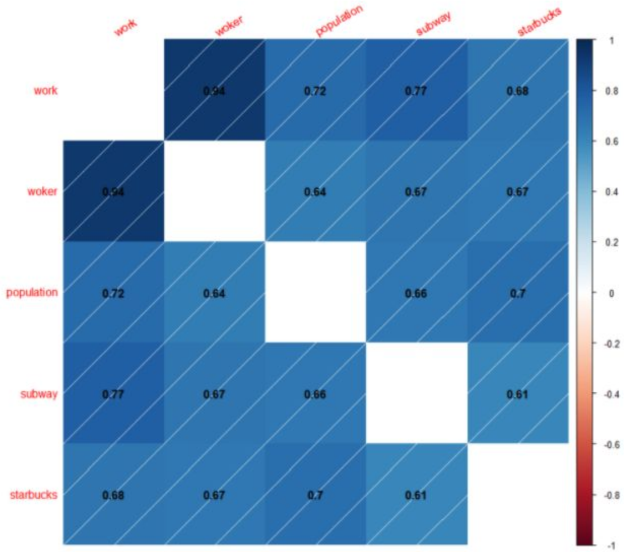
https://datamod.tistory.com/41




```
corrplot(refined_df_co_cor_kor)
```

부산 시 스타벅스 매장 분석

https://uincity.tistory.com/267



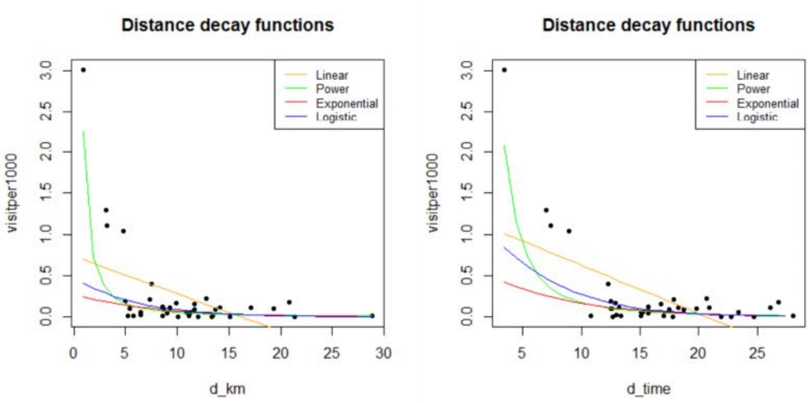
Use cases in R : Intrinsic but not for general ML

MCI package (Multiplicative Competitive Interaction Model)

https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf

```
ijmatrix <- ijmatrix.create(shopping1_KAeast, "resid_code", "gro_purchase_code",
                            "gro_purchase_expen")

ijmatrix
        interaction resid_code  gro_purchase_code freq_ij_abs freq_i_total     p_ij_obs
1      resid1-ALDI1     resid1              ALDI1          10          186 0.053763441
2     resid1-ALDI11     resid1             ALDI11           0          186 0.000000000
3      resid1-ALDI2     resid1              ALDI2           0          186 0.000000000
```

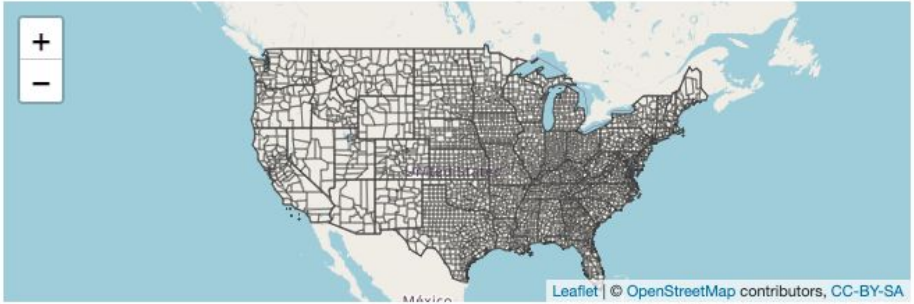


Working with GeoJSON

https://rstudio.github.io/leaflet/json.html

```
topoData <- readLines("json/us-10m.json") %>% paste(collapse = "\n")

leaflet() %>% setView(lng = -98.583, lat = 39.833, zoom = 3) %>%
  addTiles() %>%
  addTopoJSON(topoData, weight = 1, color = "#444444", fill = FALSE)
```

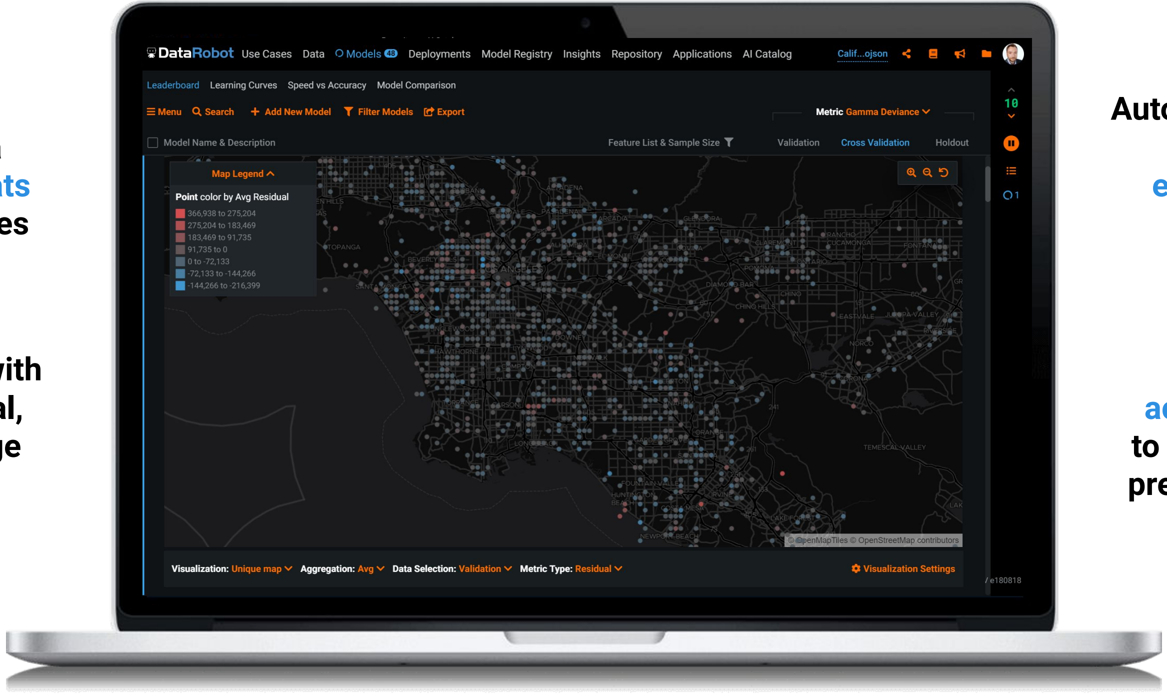


DataRobot Location AI

DataRobot Location AI. Where Predictions Count

Upload your geospatial data in a variety of file formats and well known types

Combine location with numeric, categorical, date, text, and image feature types



Automates specialized spatial feature engineering on your location data

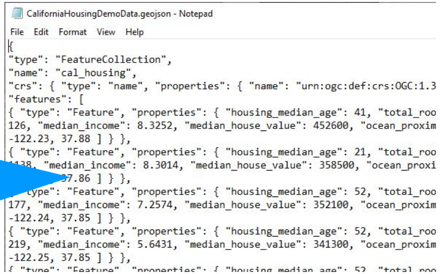
Visualize model accuracy over space to explain where your predictions work best

Add the Power of Spatial Awareness to Your Machine Learning Models

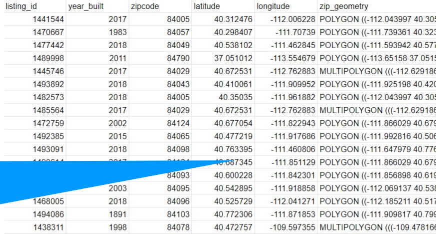
Location AI. Add a Variety Spatial Data Formats

DataRobot accepts a variety of geospatial file formats, including:

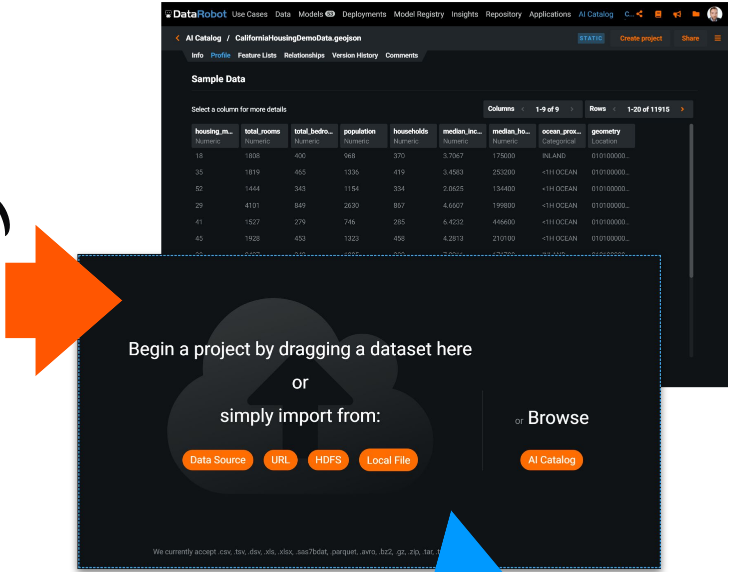
- ESRI Shapefiles
- GeoJSON
- PostGIS tables
- ESRI Geodatabase



Location AI also automatically recognizes well known spatial types like lat/long, lines and polygons in your tabular data

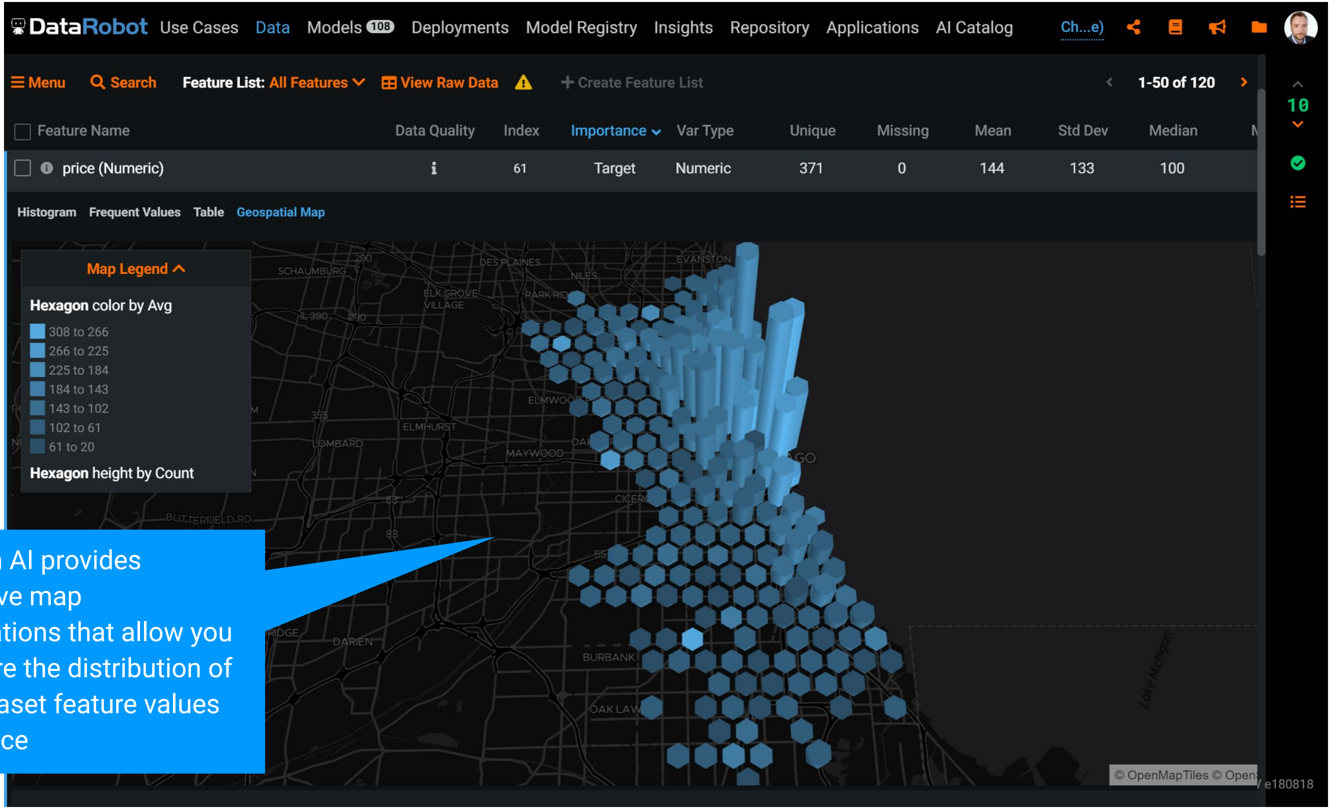


listing_id	year_built	zipcode	latitude	longitude	zip_geometry
1441544	2017	84005	40.312476	-112.006228	POLYGON ((-112.043997 40.30
1470667	1983	84057	40.298407	-111.70739	POLYGON ((-111.739361 40.32
1477442	2018	84049	40.538102	-111.462845	POLYGON ((-111.593942 40.57
1489998	2011	84790	37.051012	-113.554679	POLYGON ((-113.65158 37.05
1445746	2017	84029	40.672531	-112.762883	MULTIPOLYGON (((-112.62918
1493892	2018	84043	40.410061	-111.909952	POLYGON ((-111.925198 40.42
1482573	2018	84005	40.35035	-111.961882	POLYGON ((-112.043997 40.30
1485564	2017	84029	40.672531	-112.762883	MULTIPOLYGON (((-112.62918
1472759	2002	84124	40.677054	-111.822943	POLYGON ((-111.866029 40.67
1492385	2015	84065	40.477219	-111.917686	POLYGON ((-111.992816 40.50
1493091	2018	84098	40.763395	-111.460806	POLYGON ((-111.647979 40.77
[illegible]	[illegible]	[illegible]	40.687345	-111.851129	POLYGON ((-111.866029 40.67
[illegible]	[illegible]	84093	40.600228	-111.842301	POLYGON ((-111.856898 40.61
[illegible]	2003	84095	40.542895	-111.918858	POLYGON ((-112.069137 40.53
1468005	2018	84096	40.525729	-112.041271	POLYGON ((-112.185211 40.51
1494086	1891	84103	40.772306	-111.871853	POLYGON ((-111.909817 40.79
1438311	1998	84078	40.472757	-109.597355	MULTIPOLYGON (((-109.47816



Drag and drop your data into a new project or upload and share via AI Catalog

Location AI. Explore Your Dataset by Location



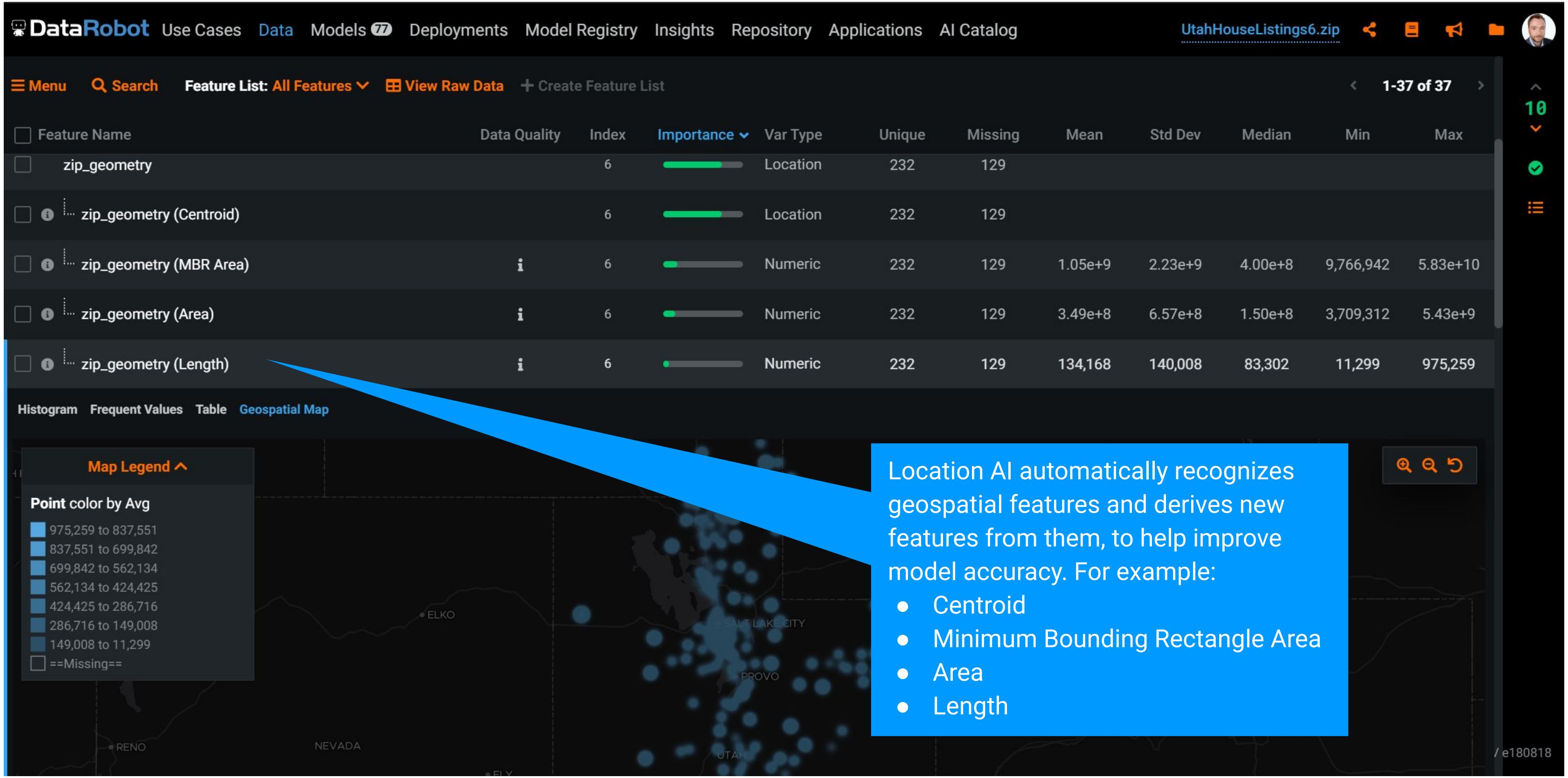
Location AI provides interactive map visualizations that allow you to explore the distribution of your dataset feature values over space

Location AI. Blend Diverse Feature Types Together

Feature Name	Data Quality	Index	Importance	Var Type	Unique	Missing	Mean	Std Dev
bathrooms	i	9		Numeric	17	0	3.01	1.47
sq_ft	i	10		Numeric	2,566	30	3,126	1,949
zip_geometry		6		Location	232	129		
zip_geome...entroid)		6		Location	232	129		
zip_geome...BR Area)	i	6		Numeric	232	129	1.05e+9	2.23e+9
zip_geometry (Area)	i	6		Numeric	232	129	3.49e+8	6.57e+8
zip_geome...(Length)	i	6		Numeric	232	129	134,168	140,008
bedroom_image	⚠	13		Image				
kitchen_image	⚠	12		Image	2,850			
high_school		32		Categorical	115			
amenities		25		Text	811	829		
jr_high		31		Categorical	157	1,043		

DataRobot allows you to mix location features with numerical, categorical, text and image features in the same dataset

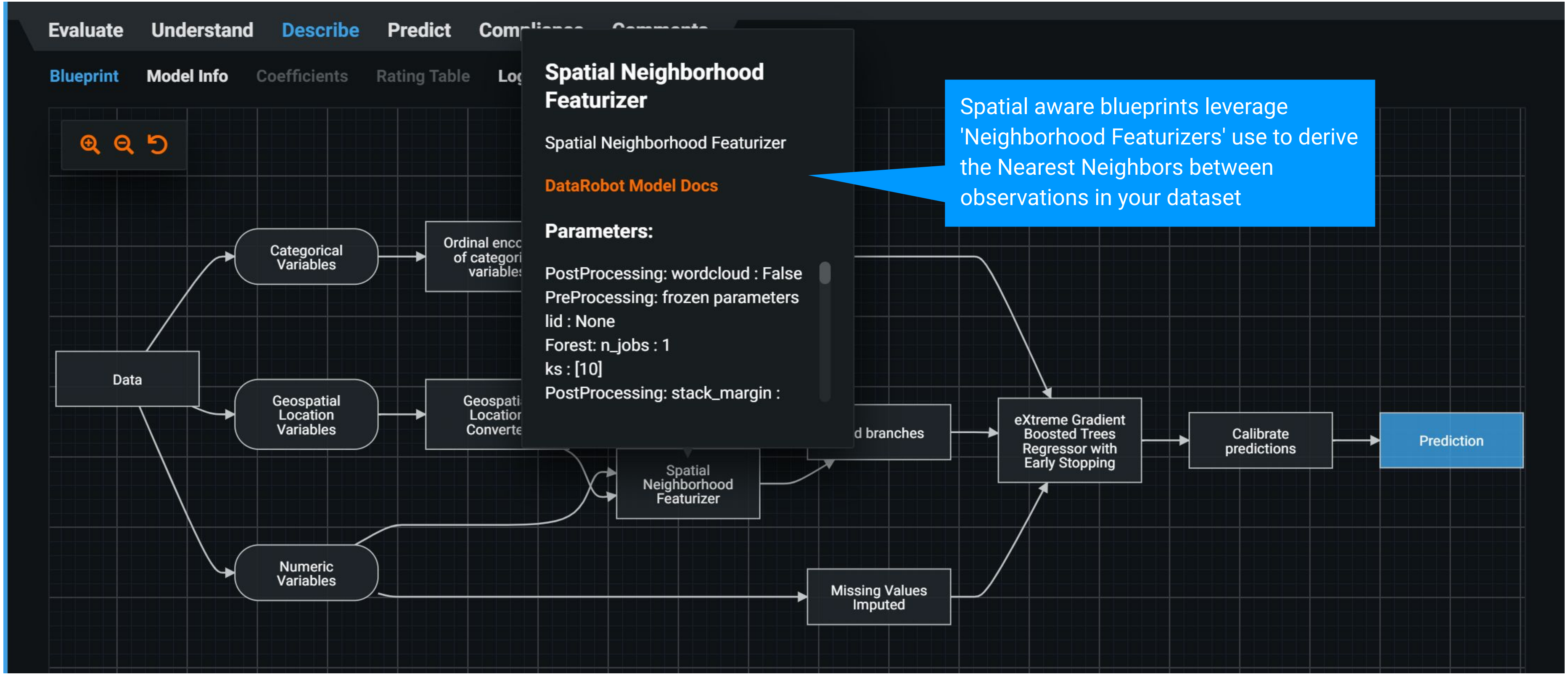
Location AI. Automated Spatial Feature Engineering



Location AI automatically recognizes geospatial features and derives new features from them, to help improve model accuracy. For example:

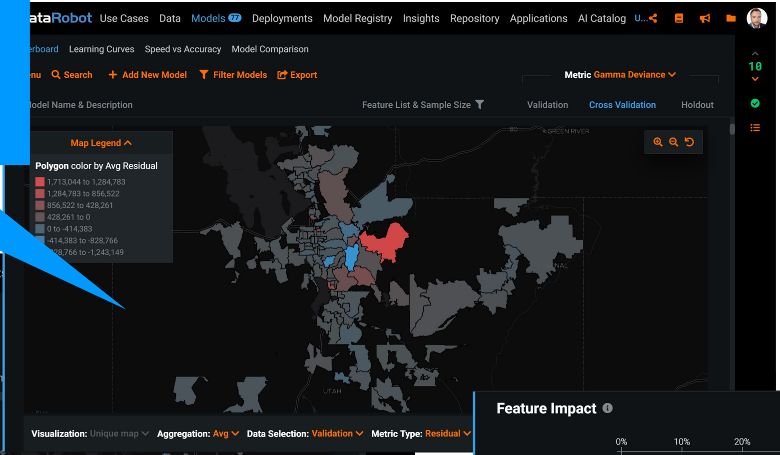
- Centroid
- Minimum Bounding Rectangle Area
- Area
- Length

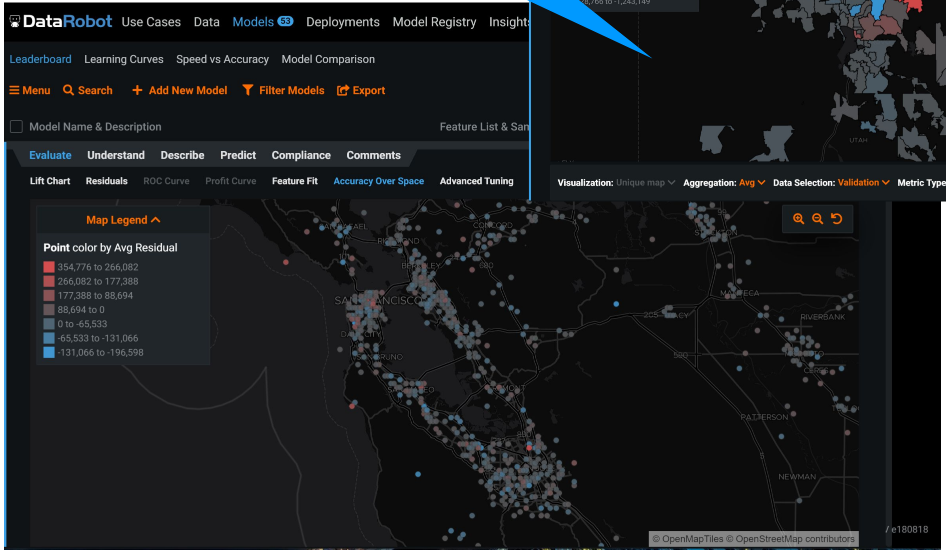
Location AI. Automated Spatial Feature Engineering



Location AI. Spatial Model Explainability

‘Accuracy Over Space’ visualizations help explain model accuracy by location. Information can be displayed as points, heatmaps and shapes





Familiar explainability charts help show the importance, fit and accuracy of your geospatial features in the model

Spatial Feature Engineering

Automated Spatial Feature Engineering

Derived spatially lagged features

Spatially lagged features are derived to gain insight into the spatial structure of the data (i.e., spatial autocorrelation) to help inform DataRobot models of spatial dependence patterns

Location AI implements several techniques for automatically deriving spatially lagged features from the input dataset, including:

- ***Spatial Lag***: A k-nearest neighbor approach to calculate mean neighborhood values of numeric features at varying spatial lags and neighborhood sizes.
- ***Spatial Kernel***: Characterizes spatial dependence structure using a spatial kernel neighborhood technique. This technique characterizes spatial dependence structure for all numeric variables using varying kernel sizes, weighting by distance.

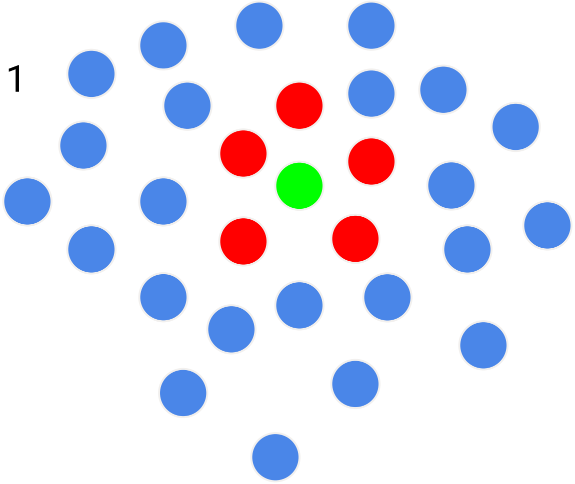
Automated Spatial Feature Engineering

Derived local autocorrelation features

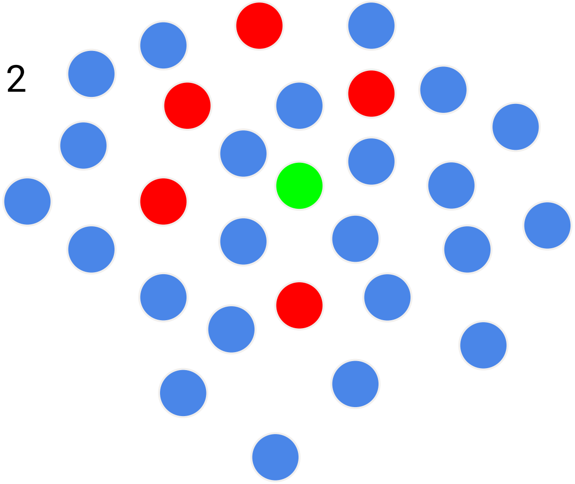
In addition to capturing spatial dependence structure in neighborhood features, Location AI uses ***local indicators of spatial association*** to capture hot and cold spots of spatial similarity within the context of the entire input dataset. The Spatial Neighborhood Featurizer calculates ***neighborhood indicators of association*** for all non-target numeric variables. The derived features characterize the relative magnitude of local spatial dependence in the input dataset. Features derived in this manner can help present particularly impactful ***local spatial dependence structures*** to DataRobot models, improving model accuracy where hot spots and cold spots or abrupt transitions in feature values are present.

About “ks” and “lags”

K= 5
Lag = 1



K= 5
Lag = 2



Searching neighbors:

- Sort all the neighboring locations by its distance to the center location in the ascending order.
- If ks = 5, lags = 1 → The neighbor feature of each center location is calculated based on the feature values of the first 5 nearest neighboring locations.
- If ks = 5, lags = 2 → The neighbor feature of each center location is calculated based on the feature values of the 6th to 10th nearest neighboring locations.

Center location

Neighbor locations picked to calculate neighbor stats

Demo

산림청 산불 이력 데이터

2011~2020 년 산불 이력 (4770건)

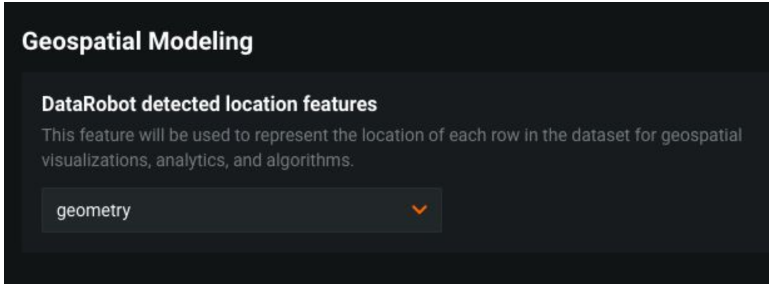
epoch_date	ext_date	epoch_arr	epoch_addr	발생원인_세부원인	피해면적_합계
2020-12-31 13:08	2020-12-31 16:00	경기	경기 양평 옥천 옥천	기타(직접입력)	0.01
2020-12-30 22:29	2020-12-31 05:00	경기	경기 남양주 수동 내방	담뱃불실화	0.06
2020-12-28 15:37	2020-12-28 17:15	충남	충남 금산 남이 대양	주택화재비화	0.03
2020-12-27 04:50	2020-12-27 06:30	경기	경기 평택 포승 도곡	담뱃불실화	0.03
2020-12-26 15:17	2020-12-26 18:10	전남	전남 화순 춘양 대신	성묘객실화	0.05
2020-12-25 16:22	2020-12-25 18:20	강원	강원 영월 영월 영흥	기타(직접입력)	0.3
2020-12-25 14:00	2020-12-25 18:00	충남	충남 천안 동남 관성	기타(직접입력)	0.05
2020-12-24 22:07	2020-12-24 22:				
2020-12-24 21:47	2020-12-24 22:				
2020-12-24 17:29	2020-12-25 08:				
2020-12-24 15:59	2020-12-24 17:				
2020-12-24 15:57	2020-12-24 16:				
2020-12-24 14:50	2020-12-24 15:				
2020-12-23 17:14	2020-12-23 19:				
2020-12-20 11:57	2020-12-20 14:				
2020-12-19 18:08	2020-12-19 21:				
2020-12-19 13:45	2020-12-19 15:				
2020-12-19 13:15	2020-12-19 16:				
2020-12-18 15:43	2020-12-18 17:				
2020-12-18 14:20	2020-12-18 18:				
2020-12-17 20:30	2020-12-17 22:				
2020-12-17 19:41	2020-12-18 02:				

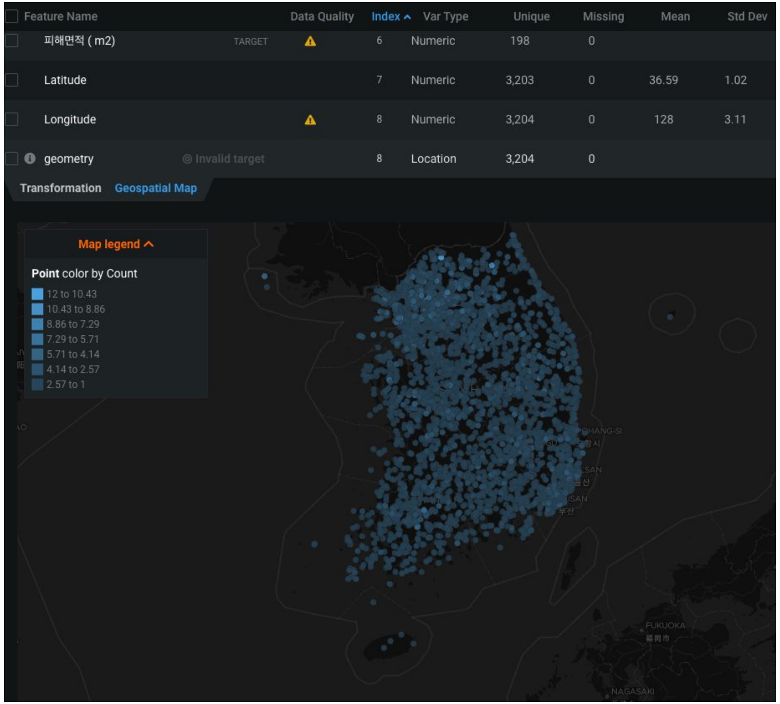
Google Geocode 기반
위/경도 Column 추가

epoch_date	ext_date	epoch_arr	epoch_addr	발생원인	피해면적 (m2)	Latitude	Longitude
2020-12-31 13:0	2020-12-31 16:00	경기	경기 양평 옥천 옥천	야영장모닥불	100	37.5196872	127.4635847
2020-12-30 22:2	2020-12-31 05:00	경기	경기 남양주 수동 내방	입산자 실화	600	37.6881723	127.3215807
2020-12-28 15:3	2020-12-28 17:15	충남	충남 금산 남이 대양	화목보일러 재 비화	300	36.0647238	127.371445
2020-12-27 04:5	2020-12-27 06:30	경기	경기 평택 포승 도곡		300	36.9888552	126.8480663
2020-12-26 15:1	2020-12-26 18:10	전남	전남 화순 춘양 대신	원인미상	500	34.9712299	126.9379291
2020-12-25 16:2	2020-12-25 18:20	강원	강원 영월 영월 영흥	입산자 실화	3000	37.1920944	128.4730785
2020-12-25 14:0	2020-12-25 18:00	충남	충남 천안 동남 관성	입산자 실화	500	36.8033118	127.3303017
2020-12-24 22:0	2020-12-24 22:25	경기	경기 남양주 화도 차산	입산자 실화 추정	100	37.6232149	127.3001357
2020-12-24 21:4	2020-12-24 22:48	경기	경기 광주 곤지암 봉현	목재펠릿보일러 재투기	500	37.342731	127.3983135
2020-12-24 17:2	2020-12-25 08:10	경북	경북 경주 안강 옥산	입산자 실화	30000	36.011763	129.163165
2020-12-24 15:5	2020-12-24 17:20	경북	경북 의성 비안 자락	입산자 실화	500	36.3371869	128.4906321
2020-12-24 15:5	2020-12-24 16:30	부산	부산 사하 장림	원인미상	100	35.0768871	128.970675
2020-12-24 14:5	2020-12-24 15:37	인천	인천 강화 송해 양오	입산자 실화 추정	600	37.8004766	126.4481907
2020-12-23 17:1	2020-12-23 19:20	경기	경기 안양 동안 비산	원인미상	600	37.4069689	126.9426549
2020-12-20 11:5	2020-12-20 14:00	울산	울산 울주 서생 나사	원인미상	800	35.3559552	129.3355846
2020-12-19 18:0	2020-12-19 21:00	전북	전북 순창 동계 어치	건축물 화재 비화	100	35.441324	127.2426959
2020-12-19 13:4	2020-12-19 15:40	경기	경기 양평 강하 성덕	입산자 실화	1500	37.4727209	127.4202625
2020-12-19 13:1	2020-12-19 16:00	경기	경기 구리 인창	입산자 실화	100	37.6146576	127.1325968
2020-12-18 15:4	2020-12-18 17:20	경남	경남 창원 마산회원 회원	건축물 실화	300	35.2207275	128.579688
2020-12-18 14:2	2020-12-18 18:00	부산	부산 사하 감천	입산자 실화	100	35.0945911	129.0092612
2020-12-17 20:3	2020-12-17 22:00	충남	충남 보령 청라 소양	야영객 취사행위	400	36.4206687	126.6678364
2020-12-17 19:4	2020-12-18 02:20	경북	경북 울진 울진 읍남	원인미상	7300	36.9953623	129.4031027

EDA

Primary location feature 지정 및 Geometry 생성

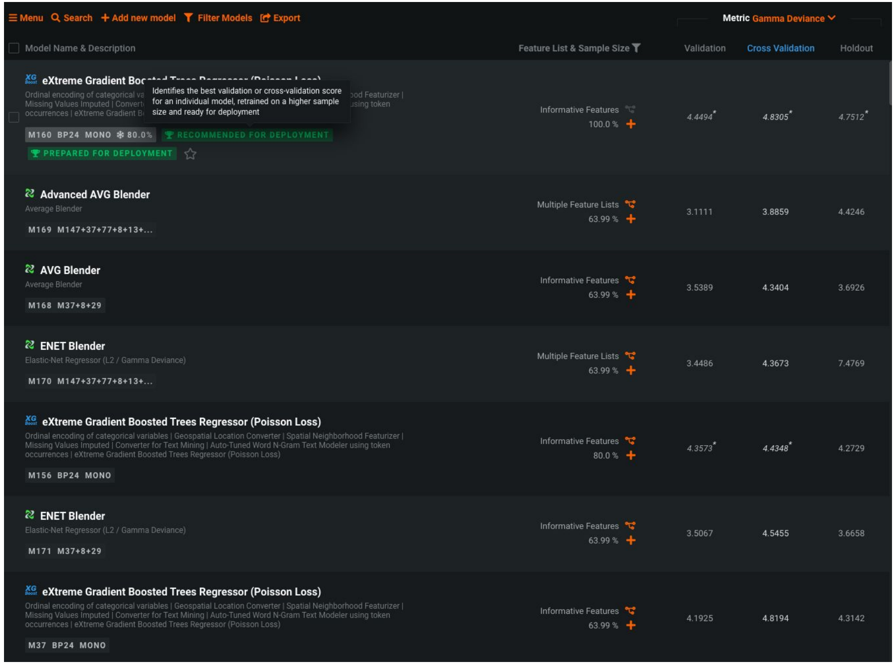




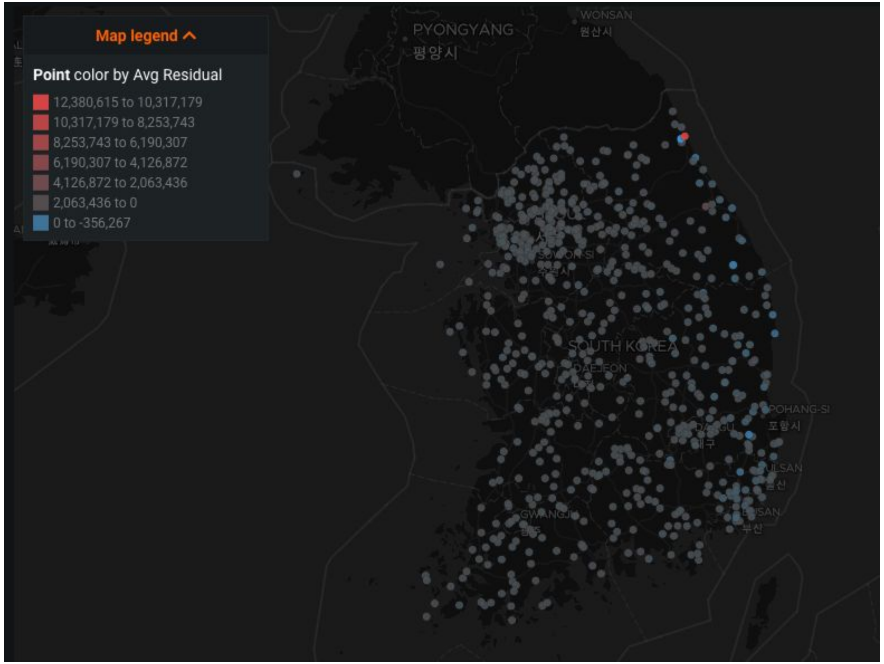
Feature Name	Data Quality	Index	Var Type	Unique	Missing	Mean	Std Dev
피해면적 (m2) TARGET	⚠	6	Numeric	198	0		
Latitude		7	Numeric	3,203	0	36.59	1.02
Longitude	⚠	8	Numeric	3,204	0	128	3.11
geometry Invalid target		8	Location	3,204	0		

Autopilot results

Leaderboard

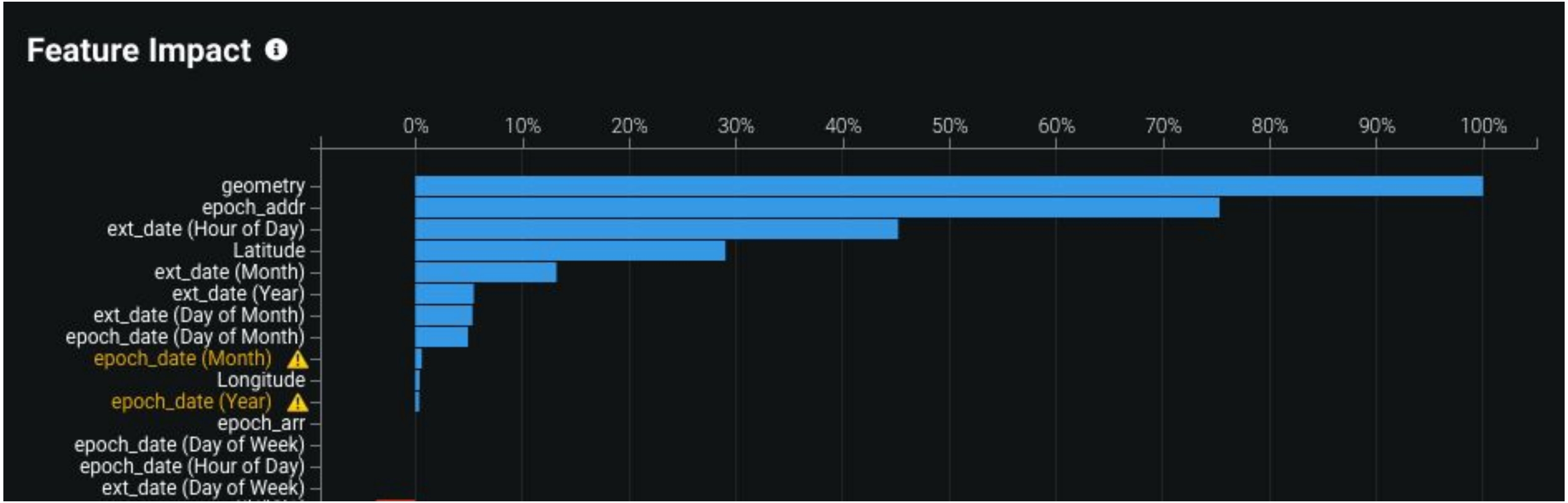


Accuracy over space



Feature Impact

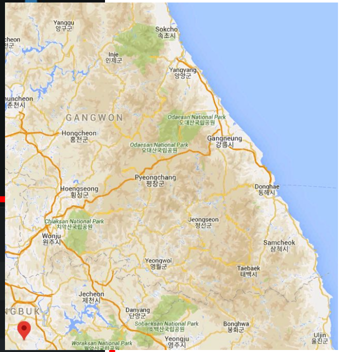
Location relevant features (geometry) dominates



Insights

Any new ideas?

Hotspots ?

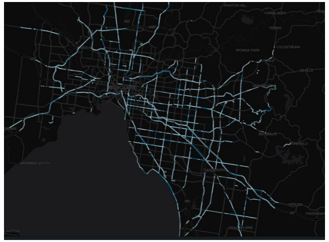


Other use cases

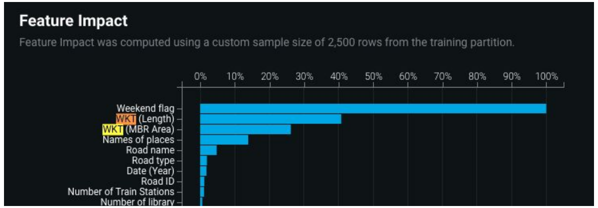
Road Accidents

Road accidents are influenced by multitude of factors, from external factors (eg. Weather) which are outside the control of the driver, and also internal factors (eg. experience) attributed to the driver.

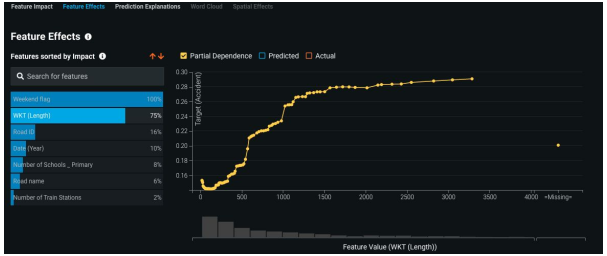
- **Geospatial maps for Polygon line**



- **Every single model has the "Length" feature in the top 5 most predictive**



- **The Length of the road is very predictive, longer the road higher the exposure to risk**



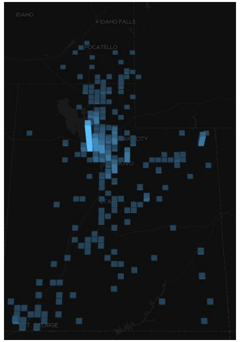
House Price

House price is affected by number of bedrooms, number of bathroom, number of floors, even image of kitchen room and exteriors,

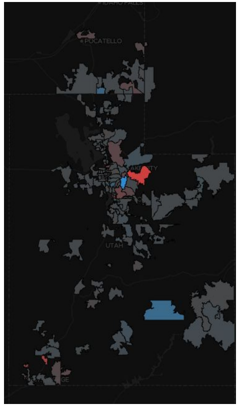
and location and topology of terrace

- **ZIP geometry of polygon**

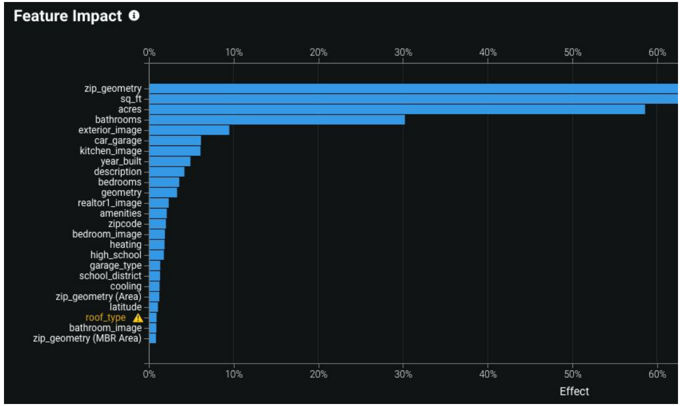
  POLYGON((-112.043997 40.305434,...))



- **Accuracy over space**



- **Zip geometry dominant in FI**

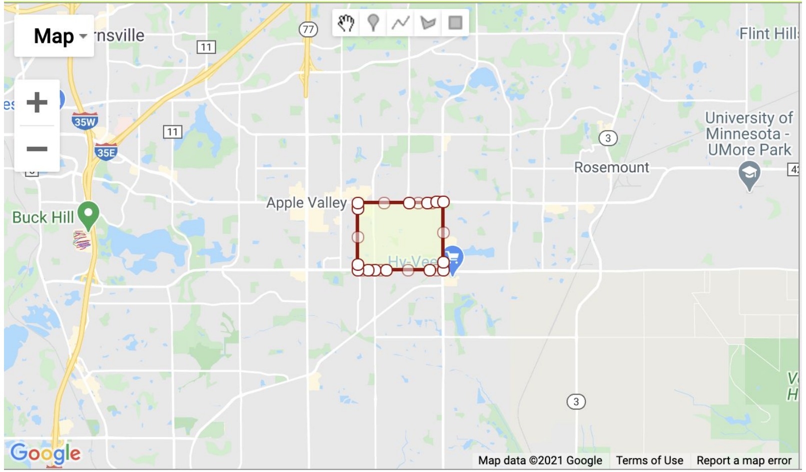


Store Location Optimization

Where should we put our next store in order to maximize sales?

- These are the top 3 reasons why the model says that we will have sales of USD **5428** in this area if we set up a shop here:

Reason	Value
geometry	POLYGON((...
secondary_education	996
population	3216



Appendix.

More details of Spatial Feature Engineering

