Machine Learning Project

CLASSIFYING VIOLENT CRIMES AND PREDICTING AREA DANGER

PROJECT MEMBERS:

BORIAN LLUKAÇAJ
ENGJËLL ABAZAJ
ERMIN LILAJ
INDRIT FERATI
JOEL BITRI
KRISTI SAMARA

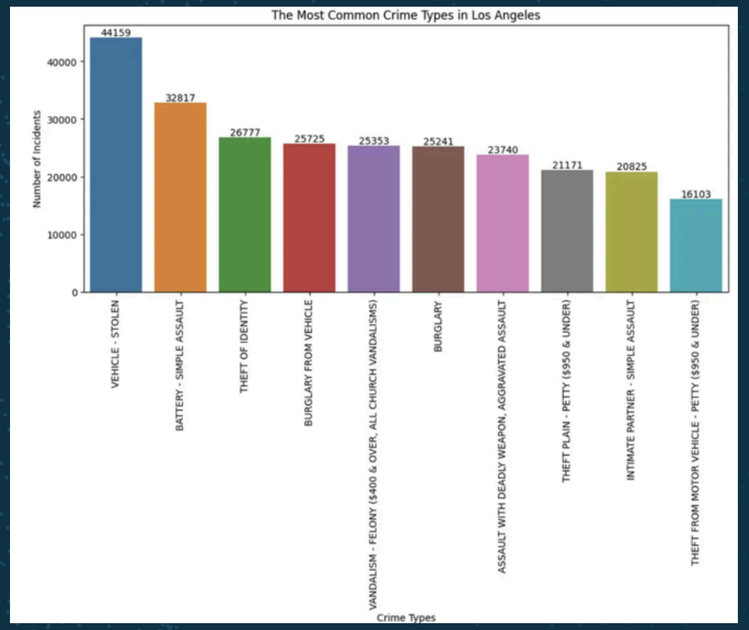
Table of Content

State-Of-Art

Author	Methodology	Findings
Author Methodology Findings Nguyenphantuan. (2023) Retrieved from Medium: https://medium.com/@ptuan5/exploratory-data-analysis-los-angeles-crime-2020-2023-5adab44973c9	Exploratory Data Analysis: Los Angeles Crime (2020–2023)	Crime concentrated in Central and 77th Street divisions; victims aged 20–40
Pangarego, R. (2023) Retrieved from Medium: https://medium.com/@rpangarego/case-study-lapd-crime-data-from-2020-to-oct-2023-analysis-using-python-1d5d6dbf58f8	Case Study: LAPD Crime Data (2020–Oct 2023) Analysis using Python	Higher crime rates on Fridays and Saturdays
Sankul, R., Aruva, T. R., Kankal, S. V., Arrapogula, G., & Khan, S. (2025)	Crime Data Analysis and Prediction using Machine Learning (Random Forest Classifier)	Improved accuracy for proactive crime prevention

Dataset

DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	Mocodes	Vict Age	Vict Sex	Vict Desce	Premis Cd	Premis Des	Weapon Us	Weapon De	Status	Status Des	Crm Cd 1	Crm Cd 2	Crm Cd 3	Crm Cd 4	LOCATION	Cross Stree	LAT	LON
1.9E+08	3/1/2020 0:00	3/1/2020 0:00	2130	7	Wilshire	784	1	510	VEHICLE - STOLEN		0	M	O	101	STREET			AA	Adult Arres	510	998			1900 S LONGWOOD		34.0375	-118
2E+08	2/9/2020 0:00	2/8/2020 0:00	1800	1	Central	182	1	330	BURGLARY FROM \	1822 1402	47	M	O	128	BUS STOP/LAYOVER (ALSO QUER			IC	Invest Cont	330	998			1000 S FLOWER		34.0444	-118
2E+08	11/11/2020 0:00	11/4/2020 0:00	1700	3	Southwest	356	1	480	BIKE - STOLEN	0344 1251	19	X	X	502	MULTI-UNIT DWELLING (APARTME			IC	Invest Cont	480				1400 W 37TH		34.021	-1
2.01E+08	5/15/2020 0:00	3/10/2020 0:00	2037	9	Van Nuys	964	1	343	SHOPLIFTING-GRA	0325 1501	19	M	O	405	CLOTHING STORE			IC	Invest Cont	343				14000 RIVERSIDE		34.1576	-118
2E+08	9/9/2020 0:00	9/9/2020 0:00	630	4	Hollenbeck	413	1	510	VEHICLE - STOLEN		0			101	STREET			IC	Invest Cont	510				200 E AVENUE 28		34.082	-118
2E+08	5/3/2020 0:00	5/2/2020 0:00	1800	2	Rampart	245	1	510	VEHICLE - STOLEN		0			101	STREET			IC	Invest Cont	510				2500 W 4TH		34.0642	-118
2E+08	7/7/2020 0:00	7/7/2020 0:00	1340	2	Rampart	265	1	648	ARSON	0329 1402	0	X	X	101	STREET			IC	Invest Cont	648	998			JAMES M WALVARADC		34.0536	-118
2.01E+08	3/27/2020 0:00	3/27/2020 0:00	1210	13	Newton	1333	1	510	VEHICLE - STOLEN		0			101	STREET			IC	Invest Cont	510				3200 S SAN PEDRO		34.017	-118
2.01E+08	7/31/2020 0:00	7/30/2020 0:00	2030	11	Northeast	1161	1	510	VEHICLE - STOLEN		0			101	STREET			AA	Adult Arres	510				KENMORE FOUNTAIN		34.0953	-118
2E+08	12/4/2020 0:00	12/3/2020 0:00	2300	1	Central	105	1	510	VEHICLE - STOLEN		0			101	STREET			IC	Invest Cont	510				400 SOLANO		34.071	-11
2.01E+08	6/26/2020 0:00	6/25/2020 0:00	2200	12	77th Street	1259	1	510	VEHICLE - STOLEN		0			101	STREET			IC	Invest Cont	510				FLORENCE WADSWOF		33.9747	-118
2E+08	3/2/2020 0:00	3/1/2020 0:00	1430	4	Hollenbeck	407	1	310	BURGLARY	0344 1607	27	M	W	221	PUBLIC STORAGE			IC	Invest Cont	310				4500 HUNTINGTON E		34.0881	-118



This dataset is created and actively updated
by Los Angeles Police Department (LAPD).
It include every crime recorded from 2020.

Data pre-processing

- Data Preprocessing and Normalization
- Dataset cleaning: Removed irrelevant columns (identifiers, descriptions, redundant codes)
- Spatial data quality: Filtered out invalid coordinates (LAT/LON = 0)
- Filled missing categorical values
- Feature engineering: Created violent crime indicator and district-month aggregation
- Target creation: Binary classification (0: non-violent, 1: violent crime)
- Categorical encoding: Applied label encoding to district and premises codes
- Feature scaling: Standardized numerical features to zero mean and unit variance
- Results: Exported processed dataset (preprocessed_crime_data.csv) for modeling

	A	B	C	D	E	F	G	H	I	J	K
1	Rpt Dist No	LAT	LON	TIME OCC	Day of Week	Is Violent	Vict Age	Premis Cd	Crime Count	DATE OCC	Target
2	0	1.0640688	-1.70974	-0.55105	-0.180741403	-0.58894	-0.71143	21	0.9619996963944251	1/6/2020	1
3	0	1.0701052	-1.71018	-0.37879	0.39344934445341584	0.3796930	-0.4127	66	1.3366051252429192	2/12/2020	1
4	0	1.0574760	-1.7082	0.1633812	0.39344934445341584	-0.26606	-0.13163	119	0.21278883869743673	3/12/2020	0
5	0	1.0577368	-1.72284	0.2876299	-0.37546696	-0.91181	-0.10102	163	0.5873942675459309	4/19/2020	0
6	0	1.0596961	-1.73098	-1.66485	0.565706569	-0.26606	0.3482523	119	0.4375520960065333	5/6/2020	0
7	0	1.0603501	-1.71793	-0.6299	-0.240368903	1.6711990	-0.18577	101	0.8121575248550275	6/1/2020	1
8	0	1.0502489	-1.73032	0.1219650	0.058504742	0.3796930	0.7950359	0	0.21278883869743673	7/16/2020	0
9	0	1.0615029	-1.72271	0.3497542	-0.20945094	1.0254460	-0.10712	0	0.6623153533156297	8/2/2020	1
10	0	1.0669232	-1.72225	0.7908371	-0.008484178	-0.26606	0.8912662	7	0.3626310102368344	9/20/2020	0
11	0	1.0526579	-1.7289	0.9917058	0.5944161058900669	-0.58894	-0.15836	119	0.21278883869743673	10/20/2020	0
12	0	1.0872228	-1.7051	-1.5761	-0.611384463	-0.58894	-1.06943	120	-0.536422019	11/11/2020	0
13	0	1.0545606	-1.69946	0.2578102	-1.021356656	0.3796930	0.1310467	0	0.737236439853285	12/24/2020	1
14	0	1.0551209	-1.70916	-0.88386	0.565706569	0.3796930	-0.40784	119	0.4375520960065333	1/11/2021	0
15	0	1.0505795	-1.71777	0.1840893	-0.27643986	-0.58894	-0.11381	119	0.21278883869743673	2/19/2021	0
16	0	1.0516742	-1.71001	0.2228045	0.043941933	0.056817	-0.49152	119	0.5873942675459309	3/7/2021	0
17	0	1.0512999	-1.73488	0.2643332	0.8205037125062995	-0.58894	-0.42789	119	0.062946667	4/17/2021	0
18	0	1.0549725	-1.7039	0.6810841	1.1303274697211372	-0.26606	-0.38111	102	0.21278883869743673	5/12/2021	0
19	0	1.0433052	-1.70941	1.2816194	0.862371788	0.056817	-1.20976	49	0.21278883869743673	6/13/2021	0
20	0	1.0614911	-1.69917	0.4256840	0.4381086247726713	0.056817	-0.20588	20	0.8870786106247263	7/1/2021	1
21	0	1.0618500	-1.71842	-0.45786	0.7953828673267179	-0.26606	0.3726917	163	1.1118418679338227	8/20/2021	1
22	0	1.0586839	-1.7122	0.074632	-0.640094	-0.26606	0.1955057	0	0.4375520960065333	9/13/2021	0
23	0	1.0570391	-1.73135	-0.66753	0.8958662480450438	-0.58894	0.1935964	139	0.062946667	10/8/2021	0

Models Used:

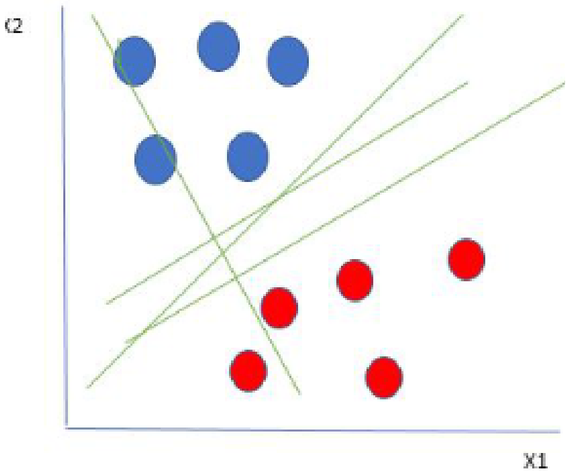
- SVM
- kNN with RF
- XGBoost Classifier
- CatBoost Classifier
- Decision Tree Classifier
- CNN
- MLP
- ExtraTreesClassifier
- AdaBoost Classifier
- LSTM Classifier
- Logistic RegressionClassifier
- kNN Classifier
- RF Regressor
- Hist GradientBoosting Regressor
- XGBoost Regressor
- LightGBM
- LinearRegression with RF

1.SVM

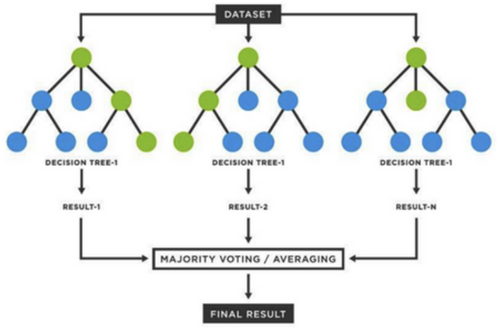
SVM	1	2	3	4
kernel	linear	sigmoid	rbf	poly
gamma	scale	scale	scale	scale
C	1	1	1	1
Class Weight	balanced	balanced	balanced	balanced
precision	0.84, 0	0.82, 0.09	0.86, 0.51	0.84, 0
recall	1.00, 0	0.75, 0.13	0.97, 0.18	1.00, 0
f1-score	0.91, 0	0.78, 0.10	0.91, 0.26	0.91, 0
accuracy	0.84 (decision boundary out of bounds)	0.65	0.84	0.84 (overfitting)

SVM finds the optimal boundary that separates classes in high-dimensional space.

The SVM model achieved the highest overall accuracy of 84% using linear, polynomial, and RBF kernels, with the RBF kernel standing out for its improved handling of minority class detection.



2.KNN with Random Forest

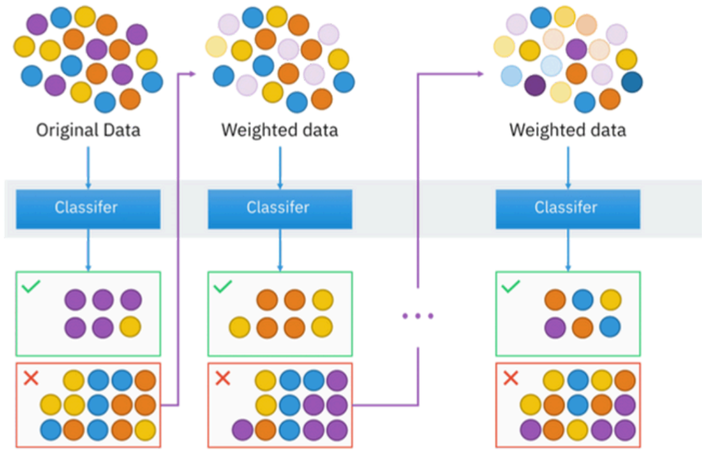


Hybrid Model	1	2	3
KNN (n_neighbors)	5	7	5
RF (n_estimators)	100	200	100
Voting Type	hard	hard	soft
Precision	0.87, 0.74	0.89, 0.72	0.91, 0.65
Recall	0.99, 0.22	0.97, 0.35	0.95, 0.53
F1-Score	0.93, 0.34	0.93, 0.47	0.93, 0.58
Accuracy	0.86574	0.87686	0.88130

This model combines k-Nearest Neighbors with a Random Forest-based feature selection or weighting for improved accuracy.

The hybrid ensemble model combining KNN and Random Forest achieved its highest accuracy of 88.13% using soft voting, demonstrating superior performance in both overall prediction and minority class handling compared to hard voting configurations.

3.XGBoost Classifier



A highly efficient, scalable gradient boosting algorithm optimized for speed and performance in classification tasks.

XGBoost achieved its best performance in Model 8 with 89.76% test accuracy and an F1-score of 0.658 for the minority class, demonstrating strong generalization and effective handling of class imbalance through careful parameter tuning.

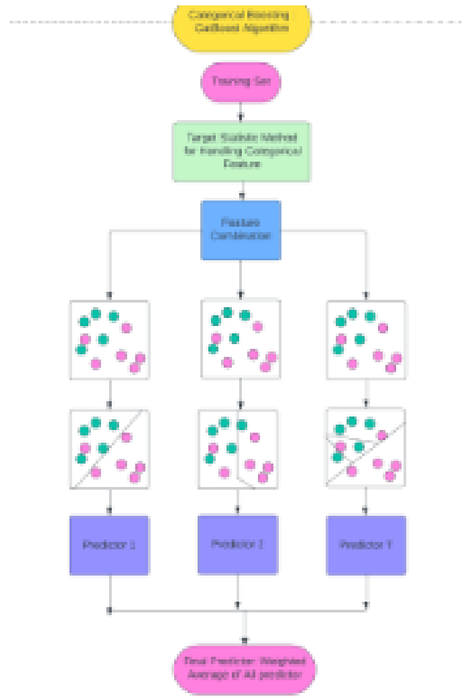
Model	Objective	Eval Metric	Max Depth	Eta	Subsample	Colsample_bytree	Gamma	Lambda	Accuracy	F1 Score (Class 1)	Precision (Class 1)	Recall (Class 1)
Model 1	binary:logistic	logloss	3	0.1	0.8	0.8	0	1	0.892342	0.646978	0.666839	0.628265
Model 2	binary:logistic	logloss	4	0.05	0.9	0.9	0	1	0.895639	0.660685	0.674898	0.647059
Model 3	binary:logistic	logloss	5	0.3	0.7	0.7	0	1	0.889353	0.644852	0.65005	0.639736
Model 4	binary:logistic	logloss	6	0.2	1	1	0	1	0.895524	0.662791	0.671934	0.653893
Model 5	binary:logistic	logloss	3	0.15	0.85	0.85	0	1	0.893071	0.650025	0.668645	0.632414
Model 6	binary:logistic	logloss	4	0.25	0.95	0.75	0	1	0.892534	0.650025	0.665134	0.635587
Model 7	binary:logistic	logloss	7	0.05	0.6	0.6	0	1	0.895792	0.651634	0.685814	0.620698
Model 8	binary:logistic	logloss	8	0.01	0.8	0.8	0	1	0.897593	0.658225	0.691481	0.628021
Model 9	binary:logistic	logloss	6	0.2	0.9	0.8	0.1	1	0.893377	0.653117	0.667601	0.639248
Model 10	binary:logistic	logloss	5	0.3	1	1	0.2	1.5	0.893607	0.657284	0.665001	0.649744

4.CatBoost Classifier

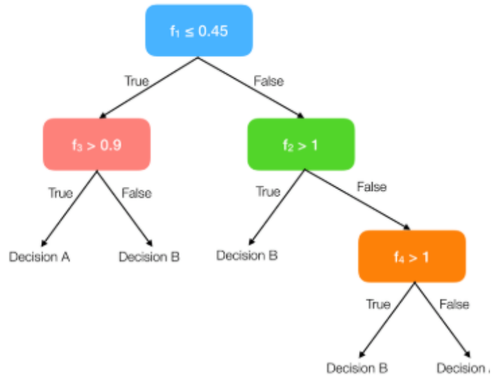
A gradient boosting algorithm that handles categorical variables automatically and reduces overfitting.

CatBoost achieved its highest test accuracy of 89.82% in Model 10, matching XGBoost in minority class F1-score (0.658), and proved highly effective for imbalanced classification tasks involving categorical data.

Model	Iterations	Learning Rate	Depth	L2 Leaf Reg	Accuracy	F1 Score (Class 1)	Precision (Class 1)	Recall (Class 1)
1	300	0.1	4	3	0.883911	0.57416	0.677056	0.498413
2	500	0.05	6	5	0.890694	0.608886	0.694836	0.54186
3	400	0.15	3	2	0.884984	0.581975	0.677807	0.509885
4	600	0.08	5	4	0.891461	0.612691	0.696734	0.546742
5	350	0.2	4	6	0.89445	0.636484	0.693015	0.588479
6	450	0.07	7	1	0.892764	0.622198	0.696283	0.562363
7	500	0.03	6	3	0.852445	0.205858	0.664447	0.121796
8	550	0.09	5	4	0.895409	0.638016	0.698722	0.587015
9	300	0.25	3	2	0.888165	0.610726	0.673433	0.558701
10	400	0.1	8	5	0.898168	0.658439	0.695546	0.625092



5.Decision Tree Classifier



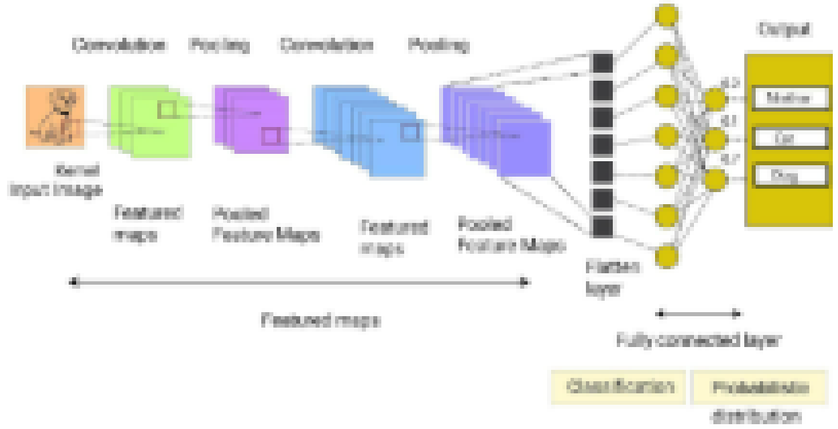
Model	Accuracy	Precision (Class 0)	Recall (Class 0)	F1-score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Macro Avg F1	Weighted Avg F1
Decision Tree 1	0.84	0.93	0.88	0.9	0.5	0.65	0.57	0.73	0.85
Decision Tree 2	0.84	0.89	0.93	0.91	0.5	0.38	0.43	0.67	0.83
Decision Tree 3	0.83	0.89	0.91	0.9	0.46	0.39	0.42	0.66	0.82
Decision Tree 4	0.83	0.87	0.93	0.9	0.45	0.29	0.35	0.63	0.81
Decision Tree 5	0.84	0.93	0.88	0.9	0.52	0.68	0.59	0.75	0.85
Decision Tree 6	0.84	0.88	0.93	0.91	0.51	0.37	0.43	0.67	0.83
Decision Tree 7	0.83	0.89	0.91	0.9	0.46	0.39	0.42	0.66	0.82
Decision Tree 8	0.83	0.87	0.93	0.9	0.45	0.29	0.35	0.63	0.81

Splits data into branches based on feature values to make predictions using a tree-like model.

Decision Tree models achieved 83–84% overall accuracy across all configurations, but those with lower minimum impurity decrease (e.g., 0.0) provided better recall for the minority class, highlighting the trade-off between model flexibility and minority class sensitivity in imbalanced classification tasks.

6.CNN

Architecture	Output Activation	Hidden Activation	Accuracy	F1-Score (Weighted Avg)
1	sigmoid	relu	0.88118964	0.87
2	sigmoid	relu	0.87969493	0.87
3	sigmoid	relu	0.87287291	0.87
1	sigmoid	swish	0.87934999	0.87
2	sigmoid	swish	0.87858347	0.87
3	sigmoid	swish	0.87544075	0.87
1	softmax	relu	0.88226276	0.87
2	softmax	relu	0.88161122	0.87
3	softmax	relu	0.86524605	0.86
1	softmax	swish	0.87739537	0.87
2	softmax	swish	0.87843017	0.87
3	softmax	swish	0.87551740	0.87
1	tanh	relu	0.56415760	0.62
2	tanh	relu	0.84297869	0.77
3	tanh	relu	0.39475701	0.44
1	tanh	swish	0.18622566	0.11
2	tanh	swish	0.51697838	0.58
3	tanh	swish	0.82431397	0.82

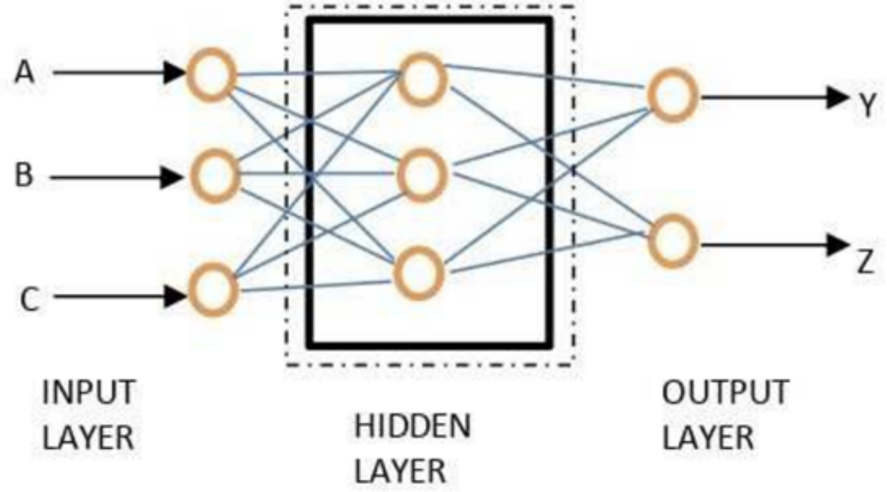


CNN is a deep learning model especially effective for spatial data classification.

Among the evaluated CNN models, Architecture 1 with ReLU activation and a softmax output achieved the highest accuracy of 88.2%, demonstrating the effectiveness of a simpler architecture with appropriate activation choices for binary classification on structured crime data.

7.MLP

MLP is an Artificial Neural Network that maps inputs to outputs through multiple layers of nodes. It follows a feedforward architecture where each node in one layer connects to each node in the successive layer



Model ID	Activation	Solver	Neurons	Accuracy	Precision	Recall	F1 Score
1	logistic	adam	200	0.876782	0.59	0.71	0.65
2	logistic	adam	300	0.880193	0.61	0.66	0.63
3	logistic	adam	800	0.879657	0.61	0.66	0.63
4	identity	adam	200	0.878507	0.6	0.68	0.64
5	identity	adam	300	0.877165	0.59	0.69	0.65
6	identity	adam	800	0.876552	0.59	0.68	0.64
7	Tahn	adam	200	0.875977	0.59	0.7	0.64
8	Tahn	adam	300	0.876591	0.56	0.69	0.64
9	Tahn	adam	800	0.878009	0.6	0.67	0.63
10	Relu	adam	200	0.888587	0.63	0.71	0.67
11	Relu	adam	300	0.890235	0.64	0.73	0.68
12	Relu	adam	800	0.888511	0.63	0.7	0.66

Among all tested configurations, the MLP classifier achieved its best performance using the ReLU activation function with a single hidden layer of 300 neurons, offering the ideal balance between high accuracy and computational efficiency.

8.ExtraTreesClassifier

Extremely Randomized Trees Classifier is an ensemble model that builds multiple decision trees and votes with their predictions for classification. It's really similar to Random Forest, but it is even more random.

The Extra Trees Classifier achieved its highest efficiency with 100 trees, max features set to 0.8, and no depth limit, reaching an accuracy of 0.89 . We can see that whenever the max depth isn't limited we get a higher accuracy.

Model ID	Estimators	max_features	Min Samples split	Min samples leaf	Max depth	Accuracy	Precision	Recall	F1 Score
1	200	log2	5	3	5	0.836312	0.49	0.7	0.57
2	100	log2	5	3	15	0.842979	0.53	0.68	0.59
3	50	log2	5	3	10	0.696497	0.5	0.73	0.61
4	100	log2	10	5	None	0.872183	0.57	0.77	0.65
5	300	log2	5	3	10	0.736279	0.46	0.84	0.61
6	100	sqrt	2	1	10	0.842185	0.56	0.72	0.68
7	100	sqrt	5	3	20	0.780163	0.44	0.73	0.59
8	200	log2	10	5	10	0.853442	0.53	0.69	0.6
9	100	0.8	5	3	None	0.898015	0.68	0.65	0.67
10	300	0.8	10	5	10	0.733673	0.45	0.84	0.6

9.AdaBoost Classifier

A boosting method that combines weak classifiers sequentially to form a strong classifier by focusing on misclassified examples.

AdaBoost reached its highest test accuracy of 85.16% in Model 8 using 180 estimators, a 0.15 learning rate, and max depth of 4, performing well on non-violent crimes but struggling with violent crime recall due to class imbalance sensitivity.

Model	n_estimato	learning_ra	max_depth	Test_Accuracy	Precision_0	Recall_0	F1_0	Precision_1	Recall_1	F1_1	Macro_Precision	Macro_Recall	Macro_F1	Weighted_Precision	Weighted_Recall	Weighted_F1
1	50	1	1	0.842979	0.842979	1	0.9148	0	0	0	0.421489	0.5	0.4574	0.710613	0.842979	0.771157
2	100	0.5	2	0.847999	0.861346	0.976949	0.915513	0.557205	0.155724	0.243419	0.709276	0.566337	0.57947	0.813589	0.847999	0.80998
3	150	0.1	3	0.848728	0.851758	0.993453	0.917165	0.671233	0.07176	0.129658	0.761495	0.532606	0.52341	0.823412	0.848728	0.79351
4	200	0.05	4	0.848574	0.851051	0.994408	0.917161	0.686224	0.065658	0.119849	0.768638	0.530033	0.51851	0.825169	0.848574	0.791966
5	75	0.3	2	0.842979	0.842979	1	0.9148	0	0	0	0.421489	0.5	0.4574	0.710613	0.842979	0.771157
6	120	0.2	3	0.849187	0.853425	0.991362	0.917236	0.649446	0.085917	0.151757	0.751436	0.538639	0.5345	0.821396	0.849187	0.79704
7	80	0.8	1	0.842979	0.842979	1	0.9148	0	0	0	0.421489	0.5	0.4574	0.710613	0.842979	0.771157
8	180	0.15	4	0.85164	0.856968	0.989088	0.918301	0.660057	0.113742	0.194045	0.758513	0.551415	0.55617	0.826049	0.85164	0.804577
9	60	0.4	2	0.842979	0.842979	1	0.9148	0	0	0	0.421489	0.5	0.4574	0.710613	0.842979	0.771157
10	140	0.25	3	0.851027	0.855924	0.989907	0.918053	0.68055	0.105443	0.181856	0.758237	0.547675	0.54996	0.825246	0.851027	0.802454

10.LSTM Classifier

Used for classifying sequence data while retaining long-range dependencies.

LSTM achieved its highest accuracy of 82.49% in Model 1 using 64 hidden units, 2 layers, and a 0.001 learning rate, demonstrating strong generalization and effective sequence learning for crime classification tasks.

Model	Hidden_Size	Num_Layers	Learning_Rate	Batch_Size	Dropout	Weight_Decay	Accuracy	Precision_0	Recall_0	F1_0	Precision_1	Recall_1	F1_1	Macro_Precision	Macro_Recall	Macro_F1	Weighted_	Weighted_Recall	Weighted_F1
1	64	2	0.001	64	0.3	0	0.824894	0.82847	0.990652	0.902231	0.677316	0.087315	0.154688	0.752893	0.538983	0.52851	0.800734	0.824894	0.765143
2	128	1	0.0005	32	0.2	0	0.824894	0.828877	0.989911	0.902265	0.668693	0.09061	0.159594	0.748785	0.54026	0.530929	0.799484	0.824894	0.765989
3	256	2	0.0001	128	0.4	0	0.819302	0.8203	0.997131	0.900113	0.686869	0.028007	0.053819	0.753584	0.512569	0.476966	0.795816	0.819302	0.744822
4	64	3	0.002	64	0.3	0.001	0.816505	0.816505	1	0.898985	0	0	0	0.408253	0.5	0.449492	0.666681	0.816505	0.734026
5	32	1	0.005	16	0.1	0	0.816505	0.816553	0.999907	0.898976	0.5	0.000412	0.000823	0.658277	0.50016	0.4499	0.758467	0.816505	0.73417
6	128	2	0.0002	256	0.5	0	0.818319	0.819829	0.996483	0.899566	0.62	0.025535	0.049051	0.719915	0.511009	0.474308	0.783162	0.818319	0.743501
7	512	1	0.0001	64	0.3	0	0.823761	0.831819	0.982877	0.901061	0.603004	0.115733	0.194195	0.717412	0.549305	0.547628	0.789633	0.823761	0.771355
8	64	2	0.001	32	0.2	0.0001	0.816505	0.816505	1	0.898985	0	0	0	0.408253	0.5	0.449492	0.666681	0.816505	0.734026
9	256	3	0.0005	128	0.4	0	0.821644	0.845612	0.956127	0.89748	0.533465	0.223229	0.31475	0.689538	0.589678	0.606115	0.788335	0.821644	0.790553
10	128	1	0.002	64	0.3	0	0.817337	0.817762	0.998889	0.899296	0.657143	0.009473	0.018676	0.737452	0.504181	0.458986	0.788289	0.817337	0.737707

11. Logistic Regression classifier

model	penalty	c	solver	acc	f1-score
1	l2	1	lbfgs	0.74963	0.34142
2	l2	10	lbfgs	0.74982	0.34158
3	l2	0.1	lbfgs	0.74981	0.34155
4	l1	1	liblinear	0.74926	0.34109
5	elesticnet	1	saga	0.74945	0.34125

A linear model used for binary or multi-class classification by modeling the probability of class membership.

Logistic Regression achieved its best performance with L2 regularization (C=10 or 0.1), reaching an accuracy of ~74.9% and a recall of 81.86% for violent crimes, effectively identifying most true cases despite a low precision of ~21.5%.

12.kNN classifier

Predicts class labels based on the majority label among the k-nearest training samples in the feature space.

The KNN classifier achieved its best performance with k=25k = 25, yielding an accuracy of 84.0% and a high specificity of ~95%, though it struggled with minority class detection, reaching only 35% recall.

model	n	acc	f1-score
1	25	0.84008	0.44694
2	19	0.83714	0.45096
3	15	0.836	0.45559
4	11	0.83389	0.46075
5	9	0.8317	0.46531
6	7	0.82777	0.46515
7	5	0.82013	0.45687
8	3	0.81061	0.45616
9	2	0.78703	0.45218

13.Random Forest Regressor

Model	n_estimators	max_depth	Predicted Crime Count	MSE
1	100	5	29.75005	1.067199
2	200	6	86.16193	1.029573
3	150	7	84.22557	0.982184
4	100	8	83.8957	0.966816
5	120	4	16.23323	1.07593
6	180	9	83.23774	0.944612
7	160	6	85.75667	1.030184
8	130	5	30.32586	1.065247
9	110	7	83.62072	0.983548
10	140	6	85.64563	1.02938

Uses an ensemble of decision trees to make robust and accurate regression predictions.

The Random Forest Regressor effectively forecasted 2024 crime counts with a low MSE of 0.944, accurately identifying zone 992 as the most crime-prone area, and demonstrated strong potential for guiding strategic, data-driven crime prevention efforts.

14. Hist GradientBoosting Regressor

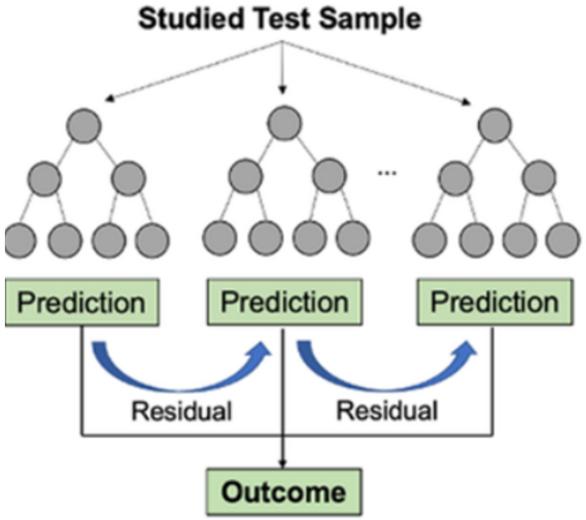
Model	Max Iter	Learning Rate	Max Depth	Top Zone Predicted (Rpt Dist No)	Predicted Crime Count	MSE
1	100	0.1	3	989	30.592858	1.039997
2	200	0.05	4	989	38.567744	1.045511
3	300	0.01	6	992	33.011724	1.025054
4	150	0.15	3	991	38.622426	1.04553
5	120	0.1	5	992	43.439997	1.058323
6	250	0.05	7	989	44.23285	1.067813
7	180	0.08	4	991	41.760491	1.051946
8	300	0.1	6	991	43.661748	1.068149
9	200	0.02	8	989	37.407431	1.040924
10	100	0.2	2	990	32.154313	1.051424

A histogram-based gradient boosting regressor that is faster and more efficient on large datasets.

The HistGradientBoosting Regressor achieved its best performance in Model 3 with an MSE of 1.03, accurately predicting zone 992 as the most high-risk area with 33.01 crimes, and offered interpretable, zone-level forecasts valuable for targeted crime prevention strategies.

15.XGBoost Regressor

The configuration with 300 estimators, 'hist' tree method, max depth of 7, and a learning rate of 0.1 (Model 2) achieved the lowest MSE of 0.9874, identifying district 990 as the most dangerous zone with a predicted crime count of 70.9798. Configurations with higher numbers of estimators, like Model 3 and 8 (500 estimators), resulted in a worse MSE of 0.9277 and 0.988, respectively. And also in increased computational demand.



Model ID	Estimators	Tree Method	Max depth	Learning rate	Most dangerous zone	Predicted crime count	MSE
1	100	hist	3	0.01	980	14.4563	1.0967
2	300	hist	7	0.1	990	70.9798	0.9874
3	500	hist	3	0.05	992	76.7349	0.9927
4	200	auto	5	0.05	991	90.6151	0.9891
5	100	hist	10	0.2	991	102.747	1.026
6	400	hist	5	0.05	990	94.1408	0.9886
7	200	hist	3	0.05	993	63.0808	1.0215
8	500	hist	7	0.01	991	89.0717	0.988
9	300	auto	5	0.1	992	98.0046	0.9923
10	250	hist	6	0.07	992	99.9561	0.9914

A powerful gradient boosting regressor designed for speed and accuracy using tree-based models.

16. LightGBM

Model	n_estimators	learning_rate	max_depth	num_leaves	min_child_samples	subsample	colsample_bytree	reg_alpha	reg_beta	Rpt Dist No	Predicted Crime Count	MSE
1	100	0.1	5	31	20	0.8	0.8	0	0	991	81.080523	0.988204
2	200	0.05	7	50	10	0.9	0.7	0.1	0.1	989	85.259763	0.97944
3	300	0.01	3	20	30	0.6	0.6	0.5	0.5	989	43.513258	1.05754
4	150	0.2	10	70	15	0.7	0.9	0	0.2	992	102.282902	1.002054
5	400	0.03	6	40	25	0.8	0.8	0.3	0.3	992	85.852243	0.98373
6	250	0.08	4	25	20	0.9	0.7	0.2	0	990	80.29609	0.99263
7	500	0.005	8	60	10	0.6	0.6	0.4	0.4	992	61.190231	0.968675
8	100	0.15	5	35	30	0.7	0.8	0.1	0.1	990	84.489756	0.987348
9	350	0.02	7	45	15	0.8	0.9	0	0.5	991	88.902639	0.988123
10	200	0.1	6	30	20	0.9	0.7	0.3	0.2	992	85.916723	0.983116

Uses LightGBM for predicting numerical values in tabular crime data.

The LightGBM regressor achieved its best performance in Model 7 with an MSE of 0.968675, accurately forecasting crime counts and identifying zone 992 as the most dangerous zone, showcasing its effectiveness in modeling complex crime patterns for precise zone-level prediction.

17.Linear Regression with Random Forest

model	LinType	RF_n	RF_depth	RF_minLeaf	MSE
1	Lin. Regression	300	None	1	0.9448
2	Lin. Regression	500	20	2	0.934
3	Ridge	400	None	1	0.9376
4	Lasso	600	30	1	0.966
5	Lin. Regression	1000	15	1	0.936

A hybrid model that combines the simplicity of linear regression with the non-linear predictive power of random forests to capture both linear trends and complex patterns in data.

The hybrid VotingRegressor model combining Linear Regression and a Random Forest Regressor achieved the best performance with an MSE of 0.934, effectively balancing interpretability and non-linear predictive power for accurate zone-level crime forecasting.

Conclusion

The project successfully tackled the classification of violent crimes and prediction of area danger levels in Los Angeles using LAPD Crime Data (2020–2024) with a variety of models, including traditional algorithms (SVM, Decision Trees, Logistic Regression, MLP), ensemble methods (Random Forest, AdaBoost, XGBoost, CatBoost), and deep learning approaches (CNN, LSTM). In the classification task, ensemble models like XGBoost, ExtraTrees, and CatBoost excelled with accuracies near 90% and a balanced precision-recall for violent crime detection, while a hybrid kNN + Random Forest model achieved a competitive F1-score of 0.58.
Deep learning models like CNN also performed well, unlike simpler models (kNN, SVM, Logistic Regression) which struggled with minority classes. For the regression task predicting crime counts, the Linear Regression with Random Forest Hybrid outperformed others with the lowest MSE, followed closely by Random Forest and LightGBM (MSE 0.93–1.07), with Hist GradientBoosting and XGBoost also identifying high-crime areas effectively, underscoring the superiority of ensemble and deep learning techniques in crime analytics, particularly CatBoost and XGBoost for classification and the hybrid model for regression.