VEHİCLE INSURANCE CLAİM FRAUD

Contents

About Us

- Vehicle insurance fraud involves conspiring to make false or exaggerated claims involving property damage or personal injuries following an accident. Some common examples include staged accidents where fraudsters deliberately “arrange” for accidents to occur; the use of phantom passengers where people who were not even at the scene of the accident claim to have suffered grievous injury, and make false personal injury claims where personal injuries are grossly exaggerated.

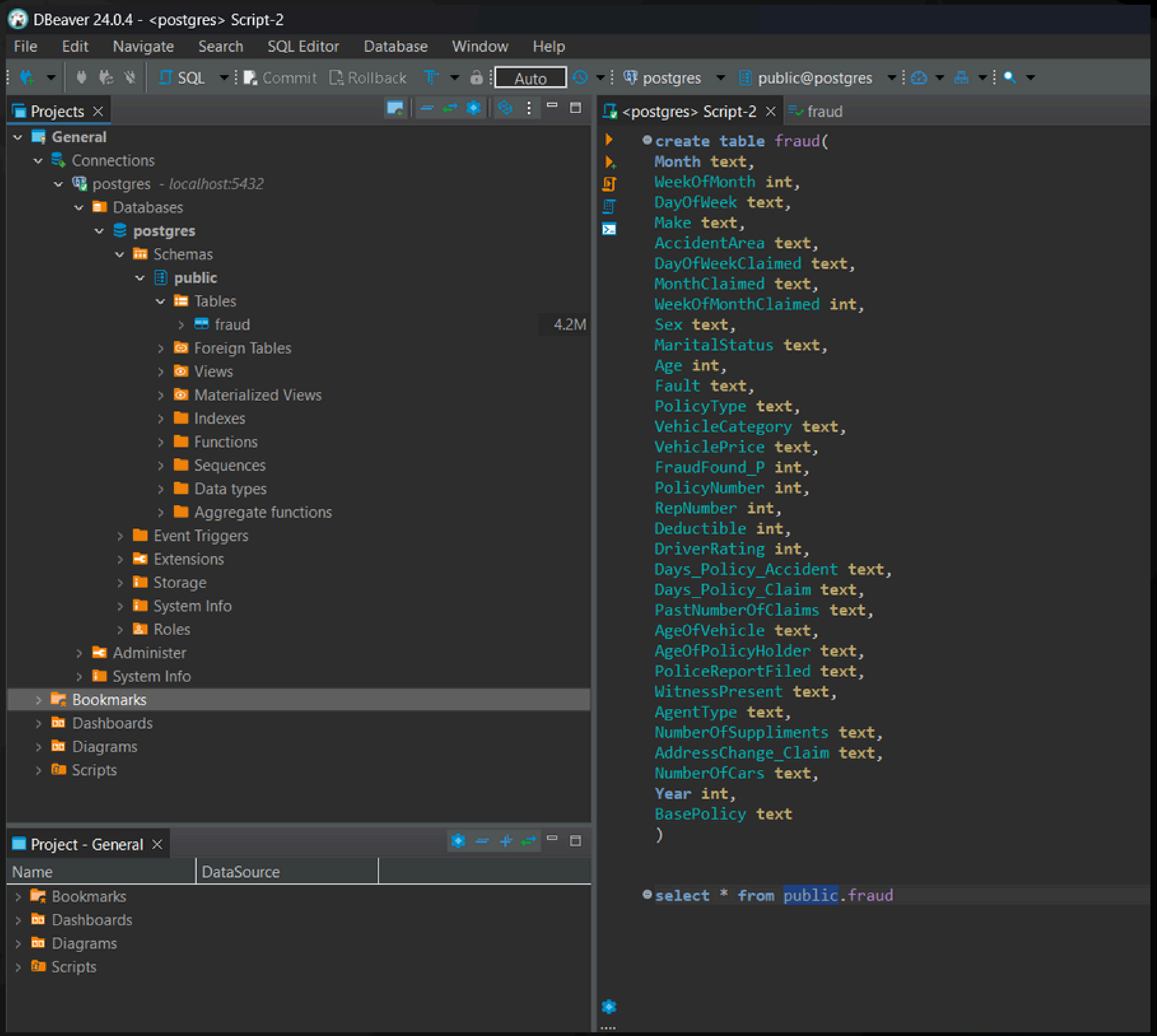
Aims

Problem Statement

- *The problem of fraudulent insurance claims is a major concern for the vehicle insurance industry.*
- *Traditional fraud detection methods such as manual review and investigation are often time-consuming and labor-intensive, making them impractical for detecting fraud at scale. Many new technologies, such as Machine Learning and Deep Learning, are being implemented so that it is easier to detect fraud.*
- *In this project, we present a machine learning model; Vehicle Insurance Claim Fraud Detection system*
- *We have also made some visuals.*
  Trained model with the following algorithms:

  - *KNN Classifier*
  - *Decision Tree Classifier*
  - *Random Forest Classifier*
  - *XGBoost Classifier*
  - *Logistic Regression*

We get our data set from the database using PostgreSQL

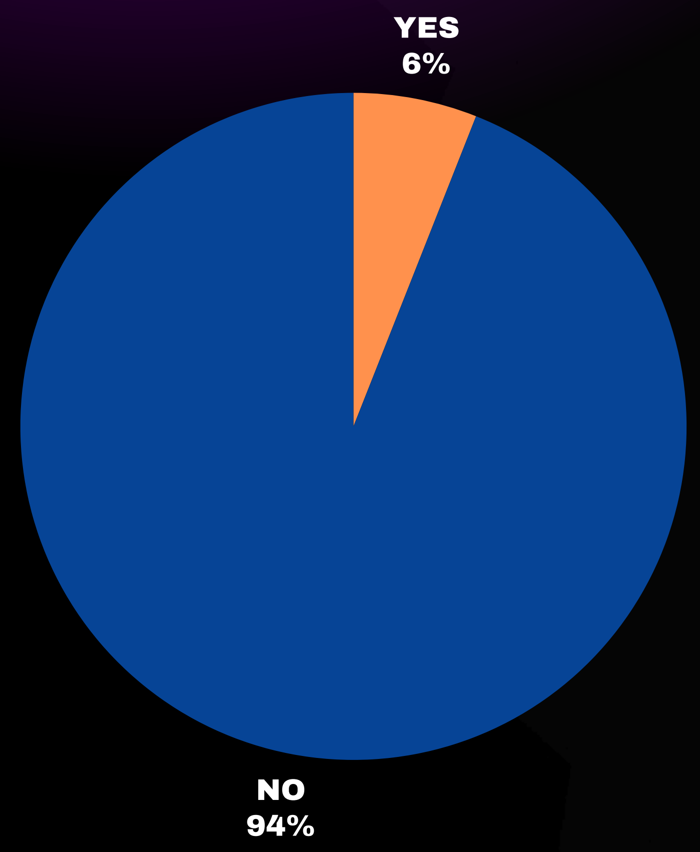


	month	weekofmonth	dayofweek	make	accidentarea	dayofweekclaimed	monthclaimed	weekofmonthclaimed	sex	maritalstatus	...	ageofvehicle
0	Dec	5	Wednesday	Honda	Urban	Tuesday	Jan	1	Female	Single	...	3 years
1	Jan	3	Wednesday	Honda	Urban	Monday	Jan	4	Male	Single	...	6 years
2	Oct	5	Friday	Honda	Urban	Thursday	Nov	2	Male	Married	...	7 years
3	Jun	2	Saturday	Toyota	Rural	Friday	Jul	1	Male	Married	...	more than 7
4	Jan	5	Monday	Honda	Urban	Tuesday	Feb	2	Female	Single	...	5 years
...	...	...	...	...	...	...	...	...	...	...	...	...
15415	Nov	4	Friday	Toyota	Urban	Tuesday	Nov	5	Male	Married	...	6 years
15416	Nov	5	Thursday	Pontiac	Urban	Friday	Dec	1	Male	Married	...	6 years
15417	Nov	5	Thursday	Toyota	Rural	Friday	Dec	1	Male	Single	...	5 years
15418	Dec	1	Monday	Toyota	Urban	Thursday	Dec	2	Female	Married	...	2 years
15419	Dec	2	Wednesday	Toyota	Urban	Thursday	Dec	3	Male	Single	...	5 years

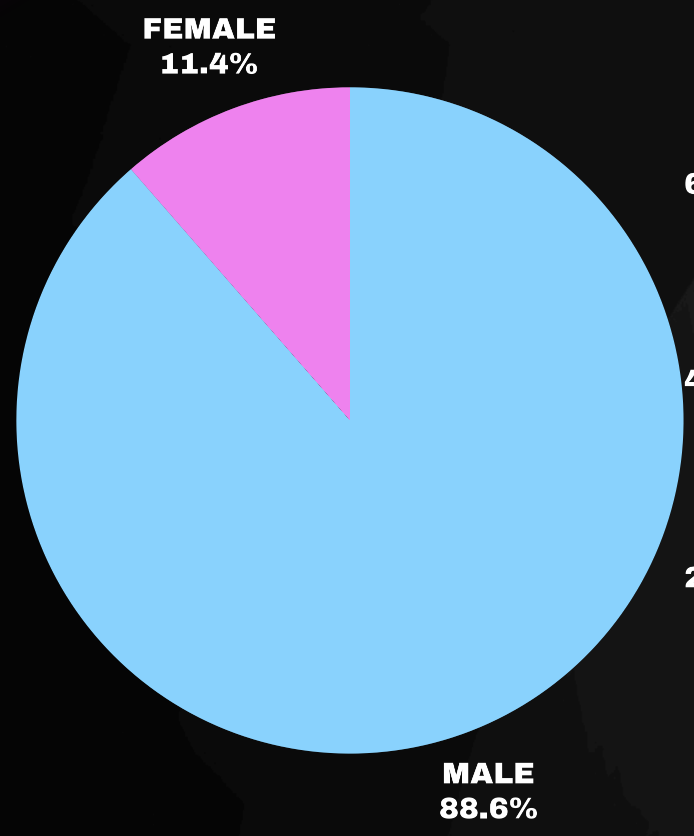
15420 rows × 33 columns

Project Timeline

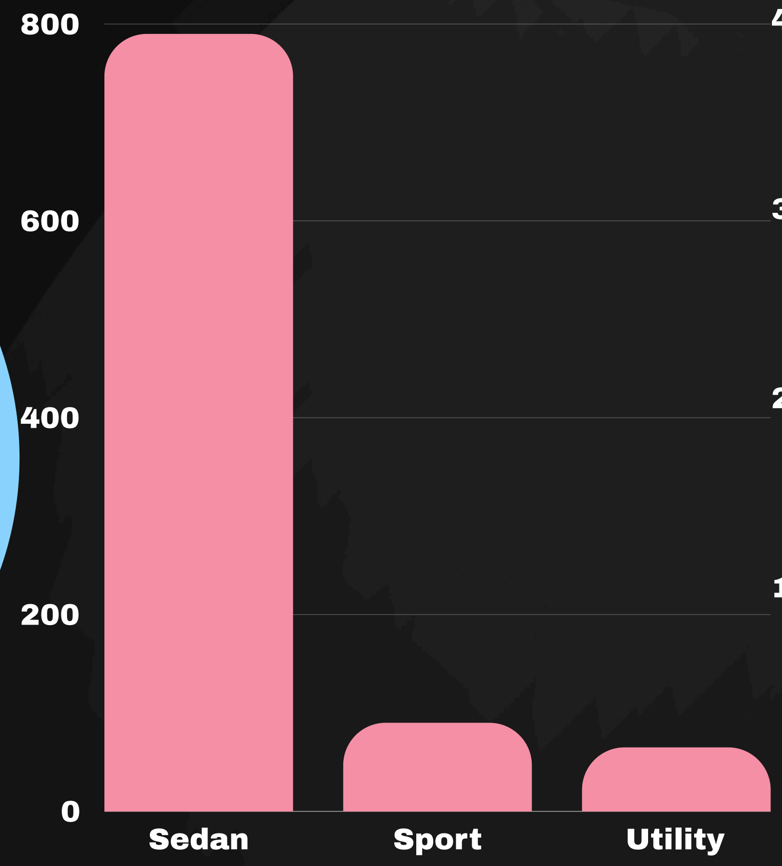
Fraund Found Distribution



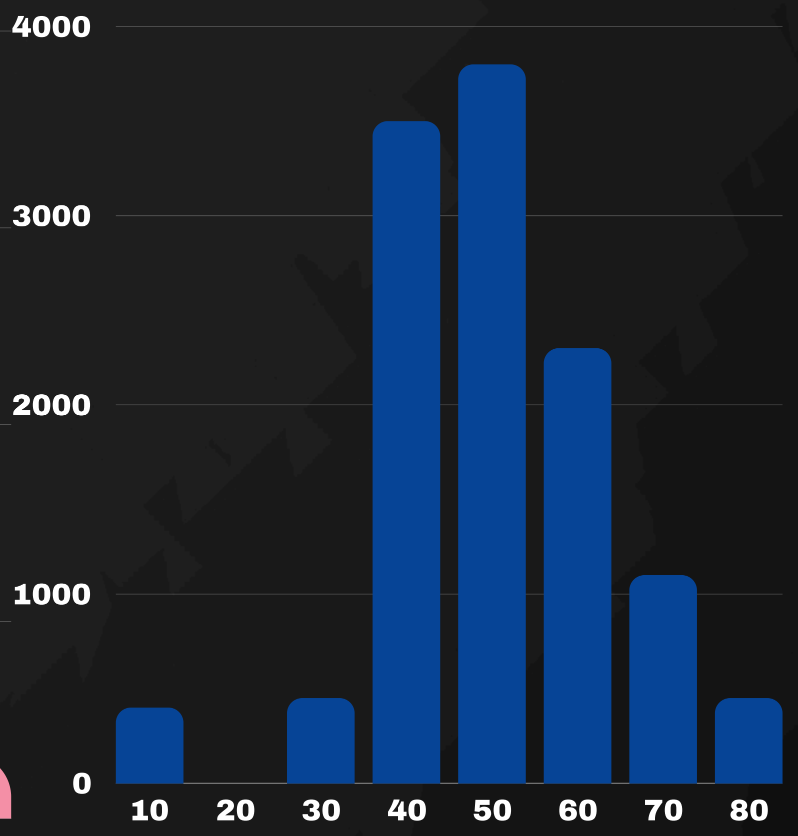
Fraud Cases by Sex



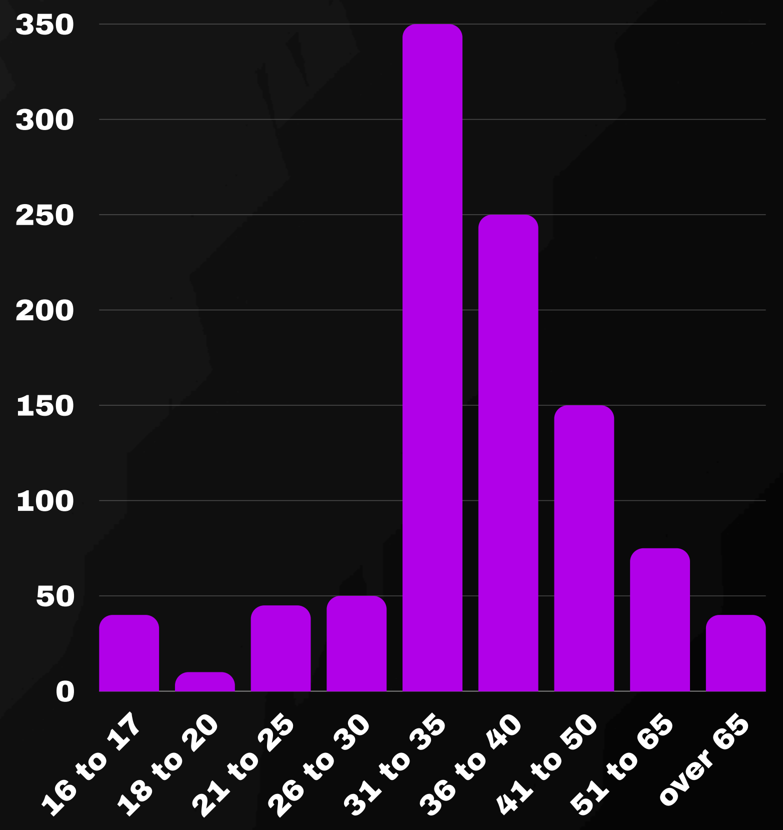
Vehicle Category

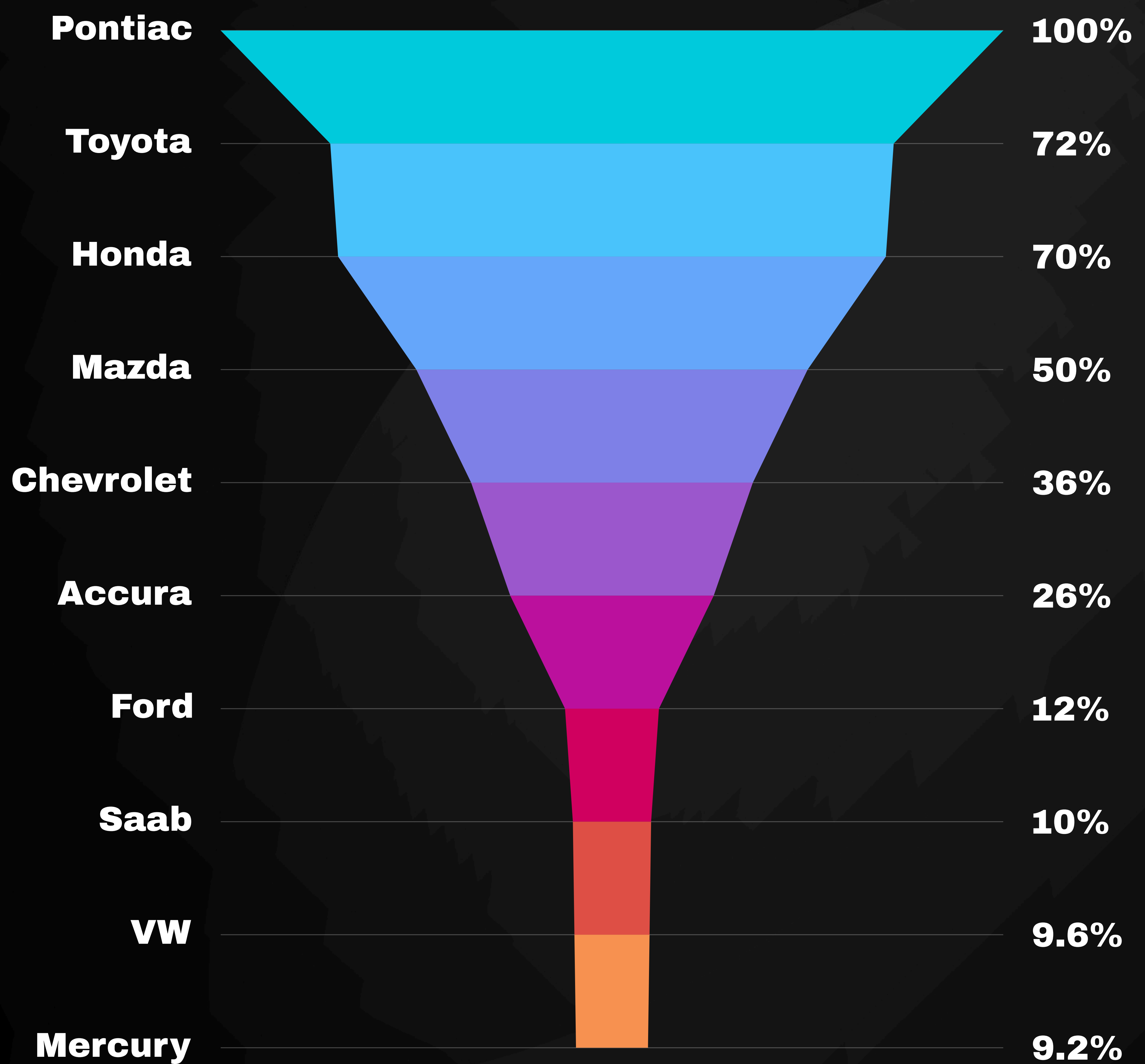


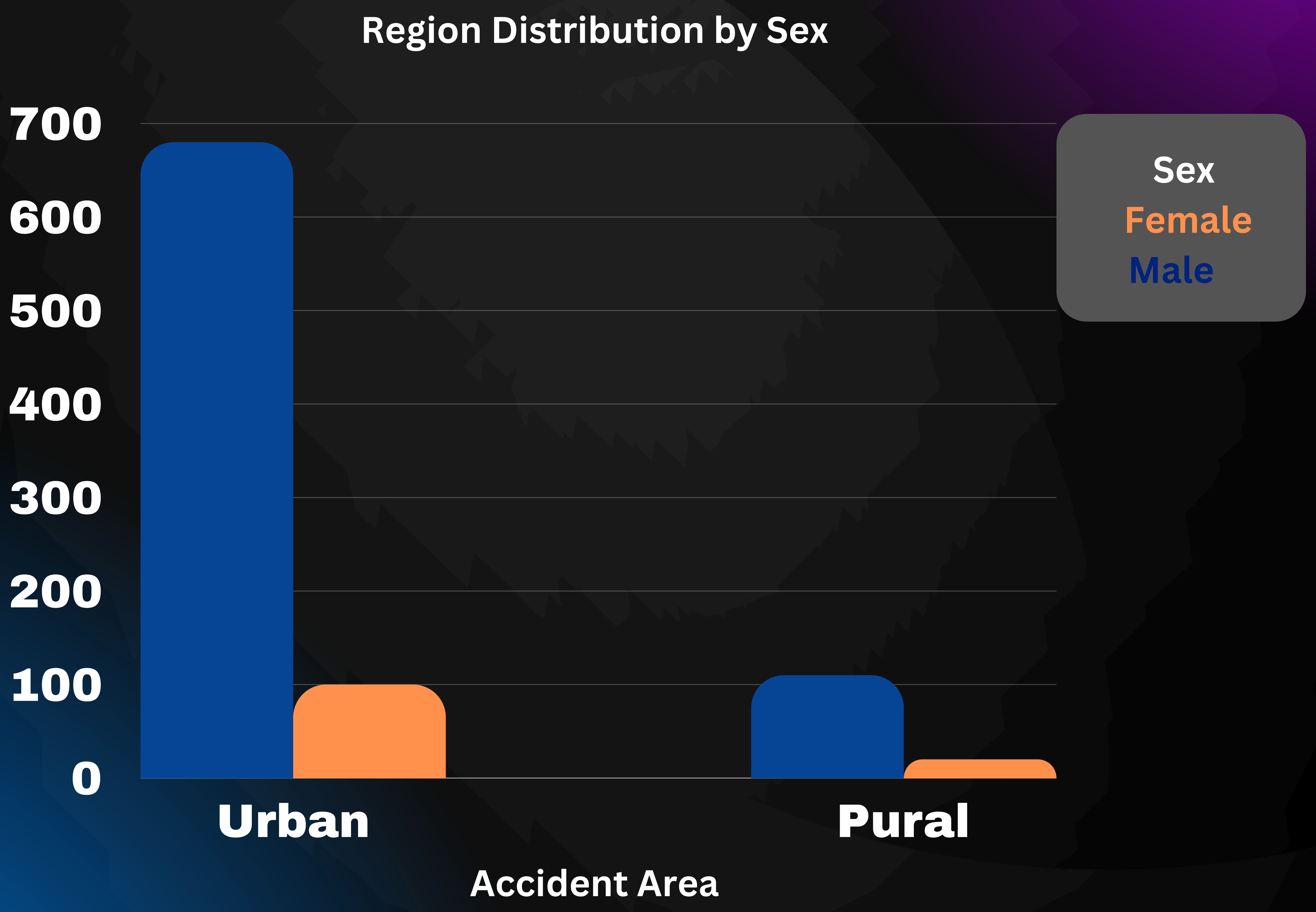
Age



Frauds Per Policy Holder Age Group




Pontiac
100%
Toyota
72%
Honda
70%
Mazda
50%
Chevrolet
36%
Accura
26%
Ford
12%
Saab
10%
VW
9.6%
Mercury
9.2%



Region Distribution by Sex
700
600
500
400
300
200
100
0
Urban
Pural
Accident Area
Sex
Female
Male

	Column Name	Unique Values
0	Month	Dec, Jan, Oct, Jun, Feb, Nov, Apr, Mar, Aug, Jul, May, Sep
1	WeekOfMonth	5, 3, 2, 4, 1
2	DayOfWeek	Wednesday, Friday, Saturday, Monday, Tuesday, Sunday, Thursday
3	Make	Honda, Toyota, Ford, Mazda, Chevrolet, Pontiac, Accura, Dodge, Mercury, Jaguar, Nisson, VW, Saab, Saturn, Porche, BMW, Mecedes, Ferrari, Lexus
4	AccidentArea	Urban, Rural
5	DayOfWeekClaimed	Tuesday, Monday, Thursday, Friday, Wednesday, Saturday, Sunday, 0
6	MonthClaimed	Jan, Nov, Jul, Feb, Mar, Dec, Apr, Aug, May, Jun, Sep, Oct, 0
7	WeekOfMonthClaimed	1, 4, 2, 3, 5
8	Sex	Female, Male
9	MaritalStatus	Single, Married, Widow, Divorced
10	Age	21, 34, 47, 65, 27, 20, 36, 0, 30, 42, 71, 52, 28, 61, 38, 41, 32, 40, 63, 31, 45, 60, 39, 55, 35, 44, 72, 29, 37, 59, 49, 50, 26, 48, 64, 33, 74, 23, 25, 56, 16, 68, 18, 51, 22, 53, 46, 43, 57, 54, 69, 67, 19, 78, 77, 75, 80, 58, 73, 24, 76, 62, 79, 70, 17, 66
11	Fault	Policy Holder, Third Party
12	PolicyType	Sport - Liability, Sport - Collision, Sedan - Liability, Utility - All Perils, Sedan - All Perils, Sedan - Collision, Utility - Collision, Utility - Liability, Sport - All Perils
13	VehicleCategory	Sport, Utility, Sedan
14	VehiclePrice	more than 69000, 20000 to 29000, 30000 to 39000, less than 20000, 40000 to 59000, 60000 to 69000
15	FraudFound	0, 1
16	RepNumber	12, 15, 7, 4, 3, 14, 1, 13, 11, 16, 6, 2, 8, 5, 9, 10
17	Deductible	300, 400, 500, 700
18	DriverRating	1, 4, 3, 2
19	Days_Policy_Accident	more than 30, 15 to 30, none, 1 to 7, 8 to 15
20	Days_Policy_Claim	more than 30, 15 to 30, 8 to 15, none
21	PastNumberOfClaims	none, 1, 2 to 4, more than 4
22	AgeOfVehicle	3 years, 6 years, 7 years, more than 7, 5 years, new, 4 years, 2 years
23	AgeOfPolicyHolder	26 to 30, 31 to 35, 41 to 50, 51 to 65, 21 to 25, 36 to 40, 16 to 17, over 65, 18 to 20
24	PoliceReportFiled	No, Yes
25	WitnessPresent	No, Yes
26	AgentType	External, Internal
27	NumberOfSuppliments	none, more than 5, 3 to 5, 1 to 2
28	AddressChange_Claim	1 year, no change, 4 to 8 years, 2 to 3 years, under 6 months
29	NumberOfCars	3 to 4, 1 vehicle, 2 vehicles, 5 to 8, more than 8
30	Year	1994, 1995, 1996
31	BasePolicy	Liability, Collision, All Perils

The Age columns contain 0.
Let's examine these.

- For the age column: we detect values of 0 and we see that this indicates <u>under the age of 18.</u>

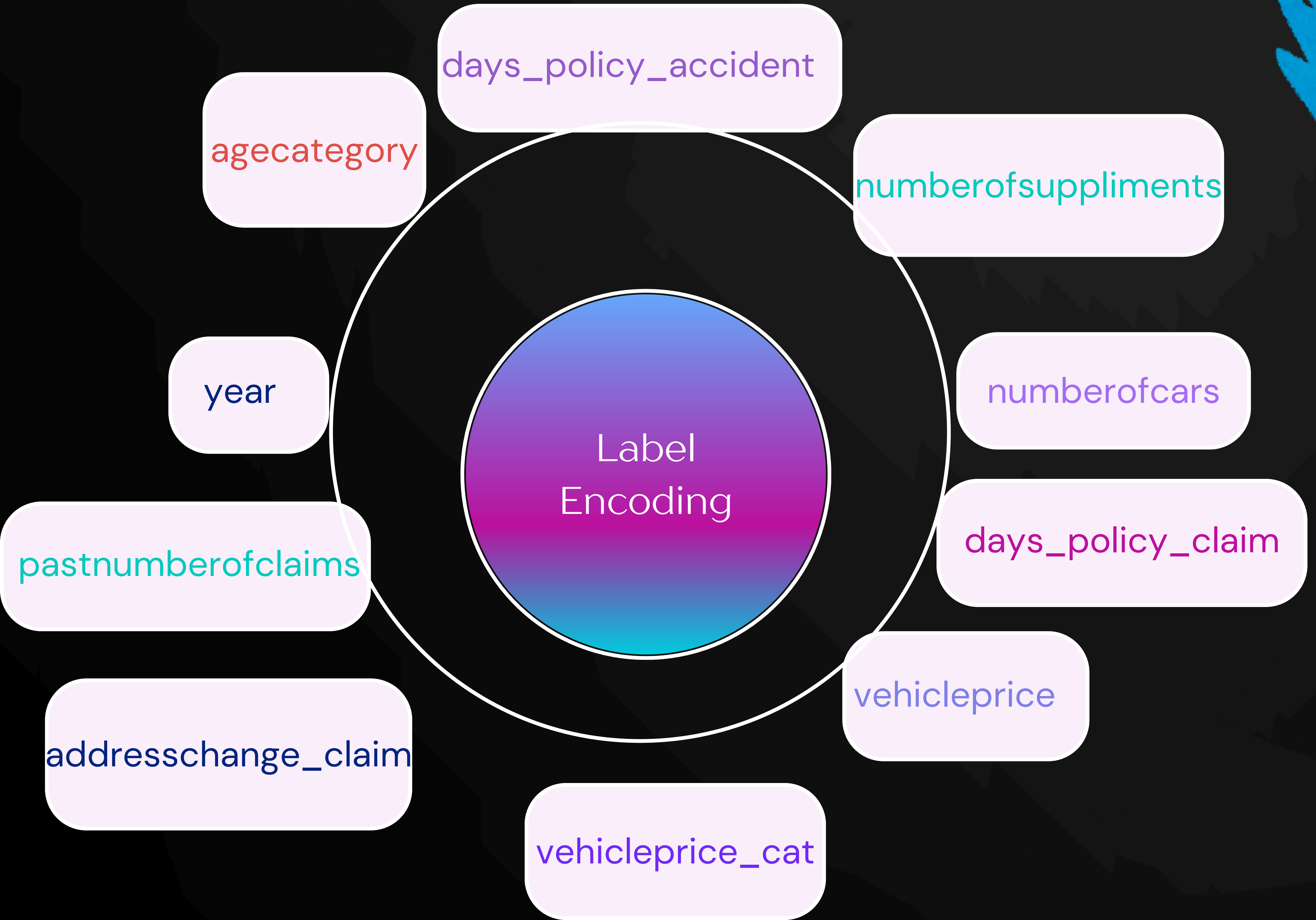
- For the values of 0 here, we randomly distribute 16 and 17.

- Then we classify the ages Decently as young, middle and old among ourselves we also classify the vehicle price_cat column as low, mid, and high.

ageofvehicle

Label Encoding

We convert the Object columns to numerical values with the help of Label Encoding.

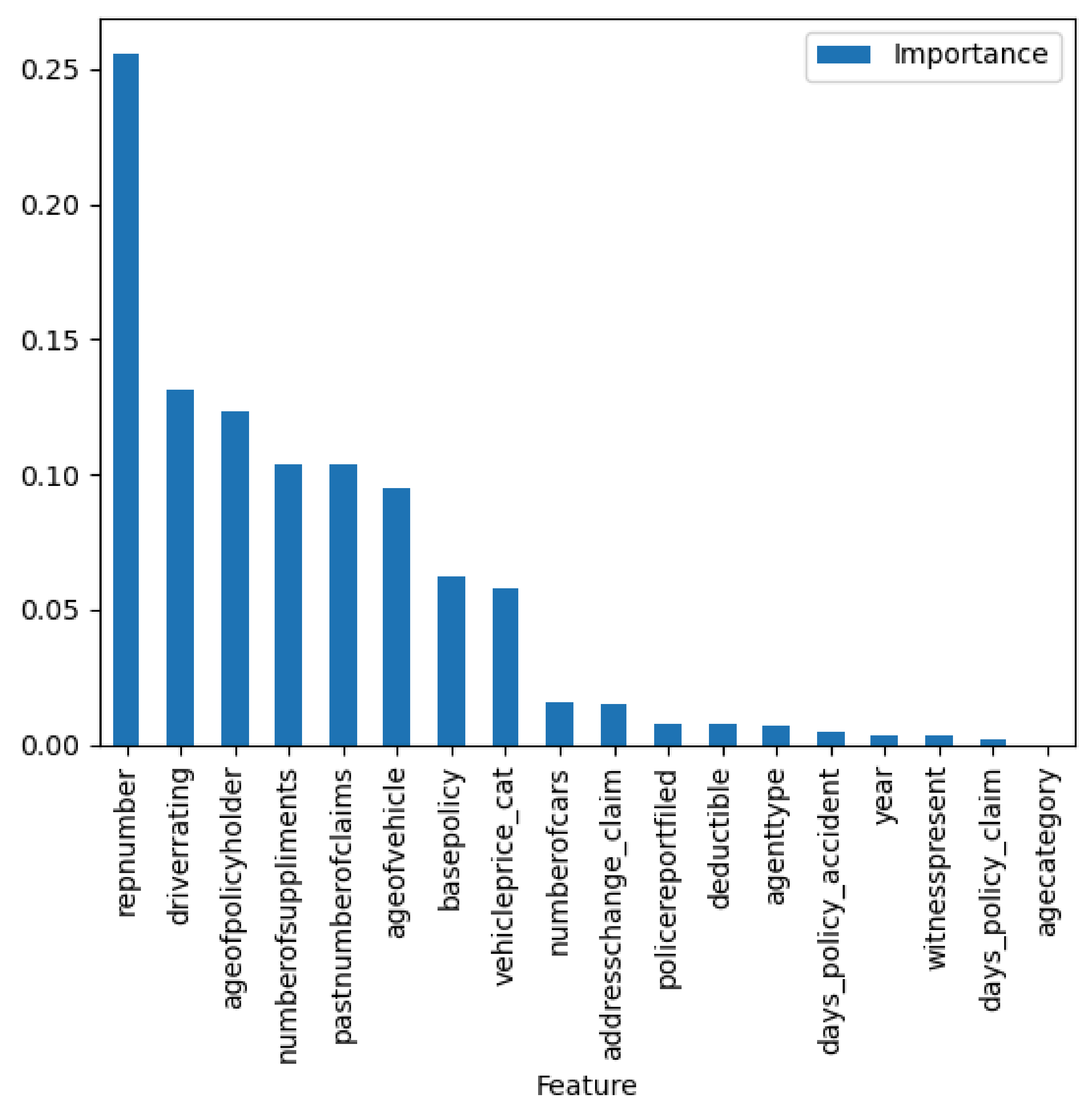


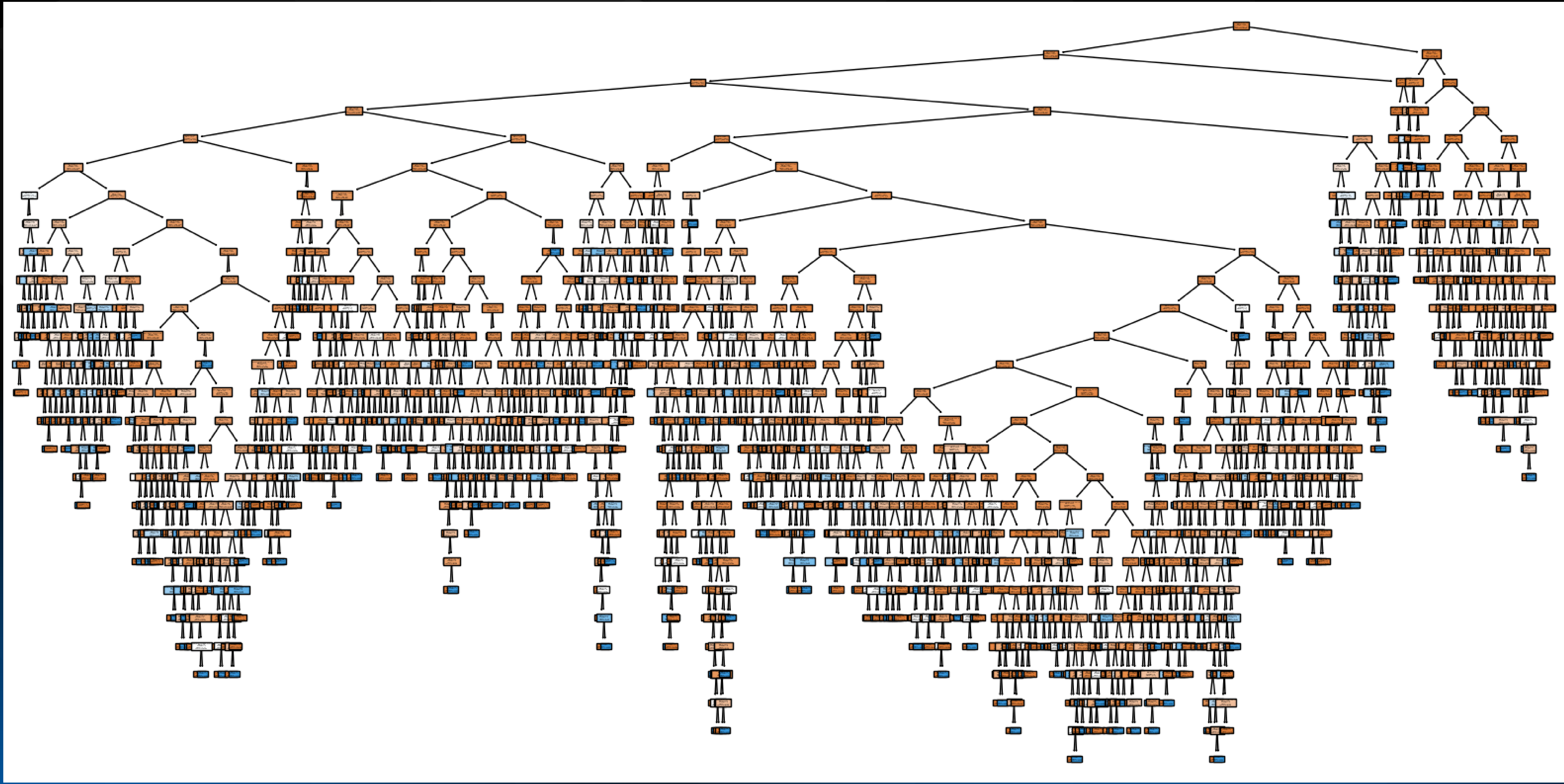
	Column Name	Unique Values
0	Month	2, 4, 10, 6, 3, 9, 0, 7, 1, 5, 8, 11
1	WeekOfMonth	5, 3, 2, 4, 1
2	DayOfWeek	6, 0, 2, 1, 5, 3, 4
3	Make	6, 17, 5, 9, 2, 13, 0, 3, 11, 7, 12, 18, 15, 16, 14, 1, 10, 4, 8
4	AccidentArea	1, 0
5	DayOfWeekClaimed	6, 2, 5, 1, 7, 3, 4, 0
6	MonthClaimed	5, 10, 6, 4, 8, 3, 1, 2, 9, 7, 12, 11, 0
7	WeekOfMonthClaimed	1, 4, 2, 3, 5
8	Sex	0, 1
9	MaritalStatus	2, 1, 3, 0
10	Fault	0, 1
11	VehicleCategory	1, 2, 0
12	VehiclePrice	5, 1, 2, 0, 3, 4
13	FraudFound	0, 1
14	RepNumber	12, 15, 7, 4, 3, 14, 1, 13, 11, 16, 6, 2, 8, 5, 9, 10
15	Deductible	300, 400, 500, 700
16	DriverRating	1, 4, 3, 2
17	Days_Policy_Accident	4, 3, 0, 1, 2
18	Days_Policy_Claim	4, 3, 2, 0
19	PastNumberOfClaims	0, 1, 3, 5
20	AgeOfVehicle	2, 5, 6, 7, 4, 0, 3, 1
21	AgeOfPolicyHolder	3, 4, 6, 7, 2, 5, 0, 8, 1
22	PoliceReportFiled	0, 1
23	WitnessPresent	0, 1
24	AgentType	0, 1
25	NumberOfSuppliments	0, 6, 3, 1
26	AddressChange_Claim	1, 0, 4, 2
27	NumberOfCars	3, 1, 2, 7, 9
28	Year	0, 1, 2
29	BasePolicy	2, 1, 0
30	AgeCategory	0, 1, 2
31	VehiclePrice_Cat	2, 1, 0

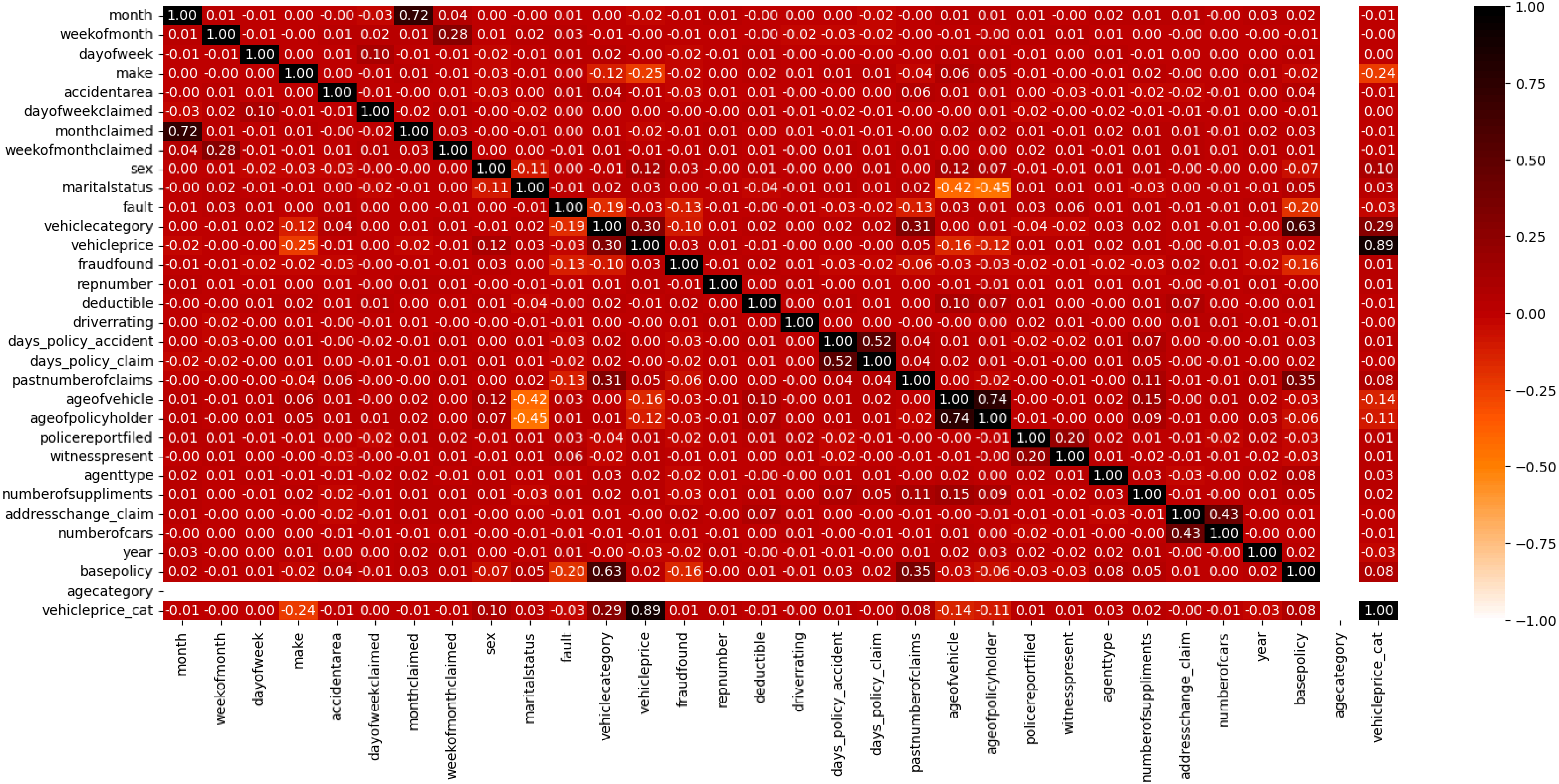
ExplainableAI

Decision Tree

Feature	Importance
repnumber	0.256
driverrating	0.131
ageofpolicyholder	0.123
numberofsuppliments	0.104
pastnumberofclaims	0.104
ageofvehicle	0.095
basepolicy	0.062
vehicleprice_cat	0.058
numberofcars	0.016
addresschange_claim	0.015
policereportfiled	0.008
deductible	0.008
agenttype	0.007
days_policy_accident	0.005
year	0.003
witnesspresent	0.003
days_policy_claim	0.002
agecategory	0.000





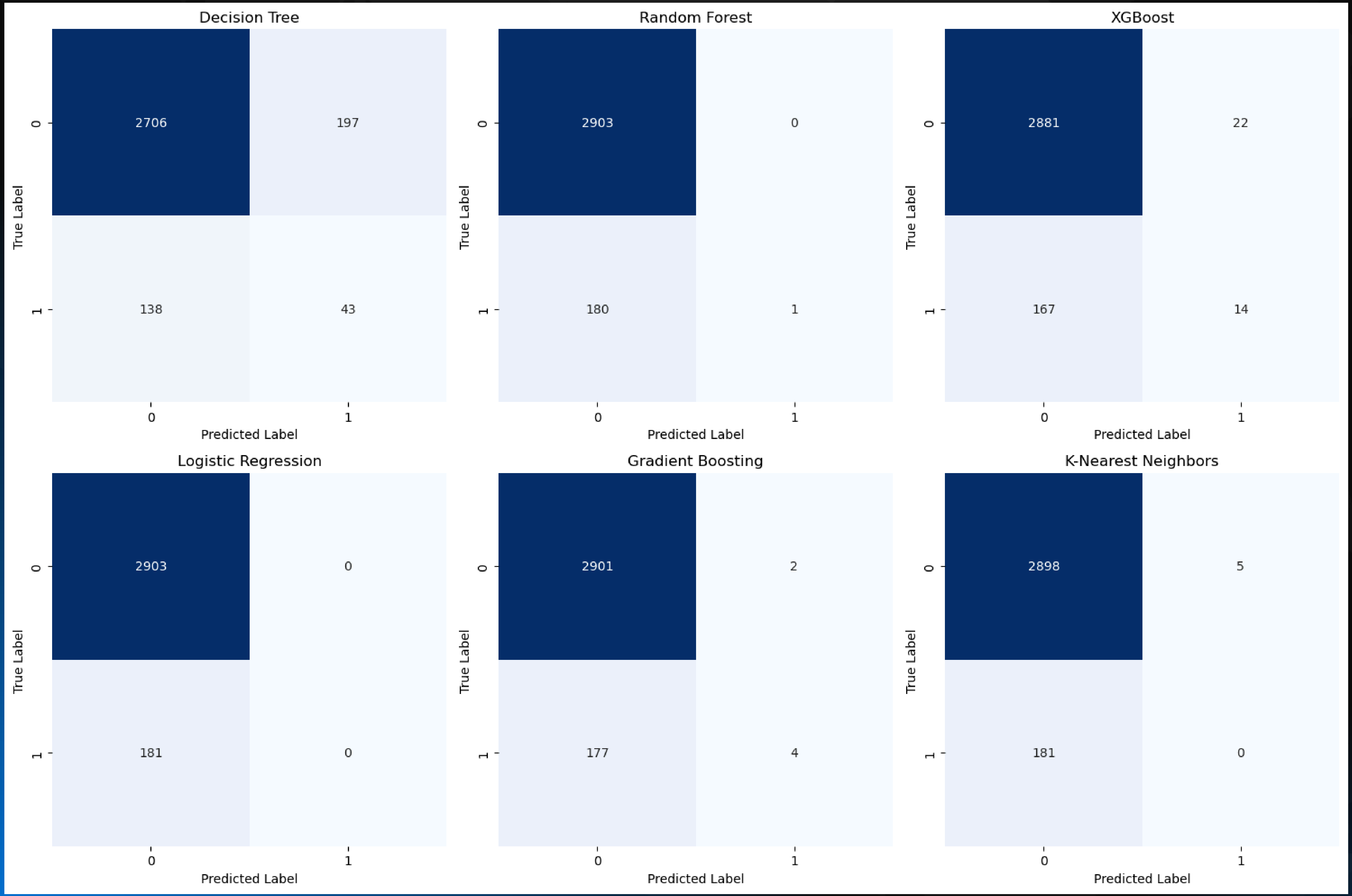

month
weekofmonth
dayofweek
make
accidentarea
dayofweekclaimed
monthclaimed
weekofmonthclaimed
sex
maritalstatus
fault
vehiclecategory
vehicleprice
fraudfound
repnumber
deductible
driverrating
days_policy_accident
days_policy_claim
pastnumberofclaims
ageofvehicle
ageofpolicyholder
policereportfiled
witnesspresent
agenttype
numberofsuppliments
addresschange_claim
numberofcars
year
basepolicy
agecategory
vehicleprice_cat
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

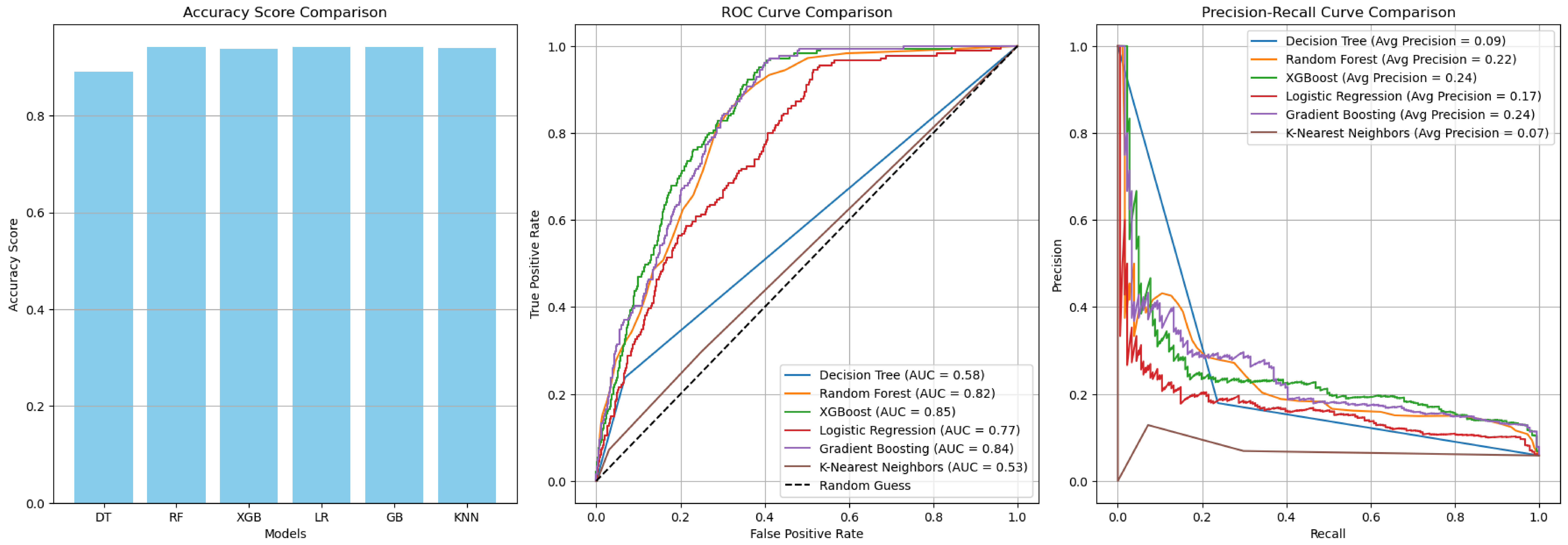
We also apply our model to other algorithms.

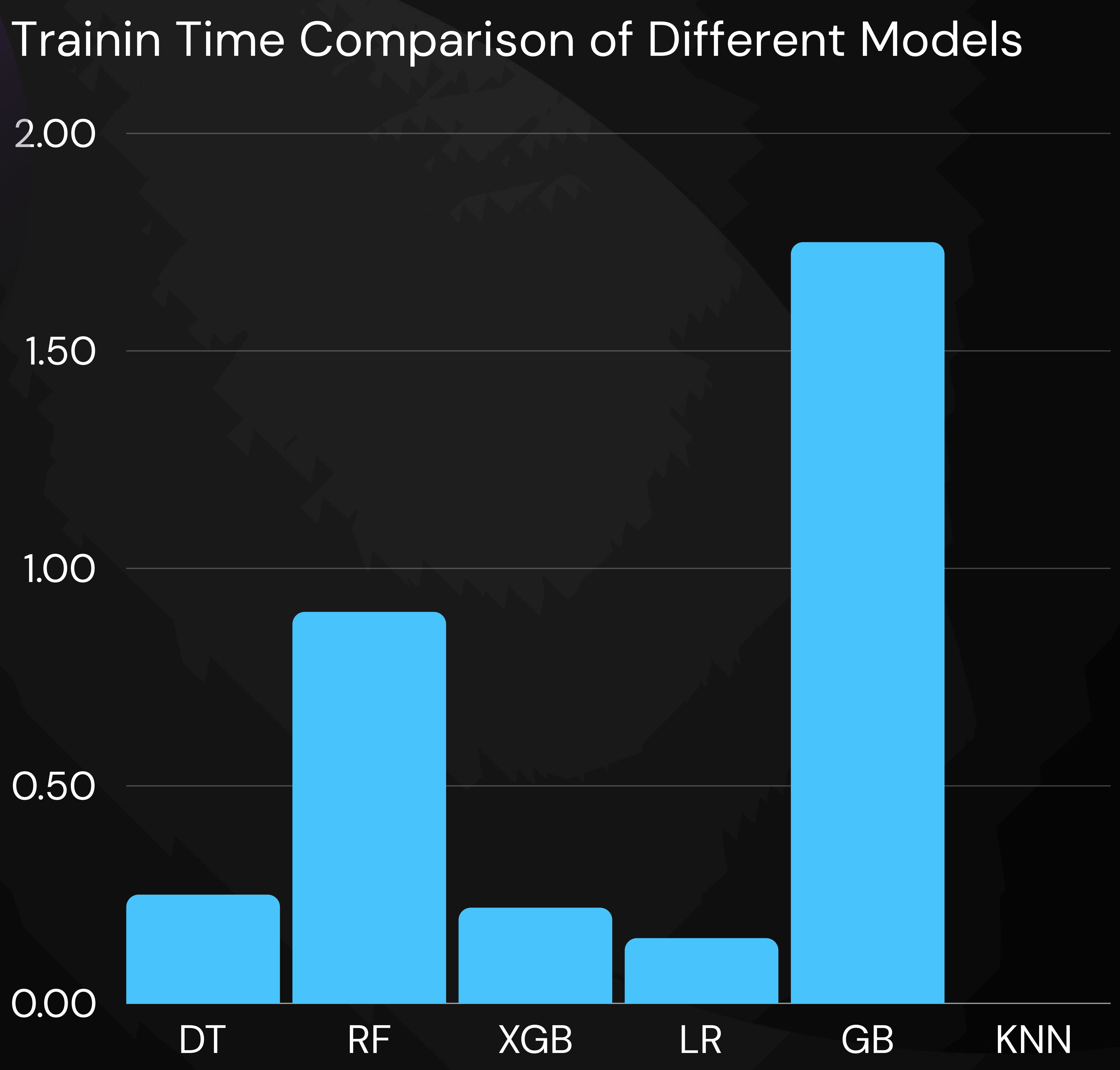
	Accuracy
"Decision Tree": dt_model,	0.8914
"Random Forest": rf_model,	0.9416
"XGBoost": xgb_model,	0.9387
"Logistic Regression": lr_model,	0.9413
"Gradient Boosting": gb_model,	0.9420
"K-Nearest Neighbors": knn_model	0.93974


1.000
0.800
0.600
0.400
0.200
0.000
Decision Tree
Random Forest
XGBoost
Logistic Regression
Gradient Boosting
K-Nearest Neighbors

Confusion Matrix




Accuracy Score Comparison
Accuracy Score
0.0
0.2
0.4
0.6
0.8
DT
RF
XGB
LR
GB
KNN
Models
ROC Curve Comparison
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Decision Tree (AUC = 0.58)
Random Forest (AUC = 0.82)
XGBoost (AUC = 0.85)
Logistic Regression (AUC = 0.77)
Gradient Boosting (AUC = 0.84)
K-Nearest Neighbors (AUC = 0.53)
Random Guess
Precision-Recall Curve Comparison
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Decision Tree (Avg Precision = 0.09)
Random Forest (Avg Precision = 0.22)
XGBoost (Avg Precision = 0.24)
Logistic Regression (Avg Precision = 0.17)
Gradient Boosting (Avg Precision = 0.24)
K-Nearest Neighbors (Avg Precision = 0.07)


Trainin Time Comparison of Different Models
2.00
1.50
1.00
0.50
0.00
DT
RF
XGB
LR
GB
KNN

Finally, we decided to create our model with the Gradient Boosting algorithm.

```
from joblib import dump, load

# Modeli pickle kullanarak kaydetme
dump(gb_model, 'gb_model.pkl')
```

✓ 0.1s

```
['gb_model.pkl']
```

THANK YOU

For listening to us

- *Soner Kocoglu*
- *Azad Halhalli*
- *Huseyin Berke Ilbay*