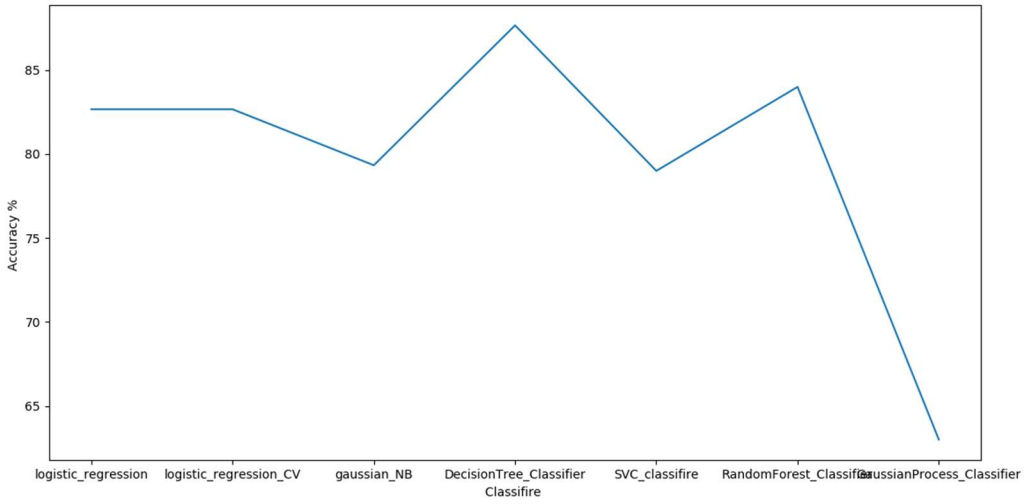
Classifier used

- LogisticRegressionCV
- DecisionTreeClassifier
- LogisticRegression
- GaussianNB
- RandomForestClassifier
- SVC
- GaussianProcessClassifier
- ANN

Accuracy graph:



With ANN we got atmax 76 %. I used Tensorflow grid search for getting accuracy for different hyper-parameters.

Due to lack of time and hardware availability I wasn't able to try many permutations in Grid Serch.

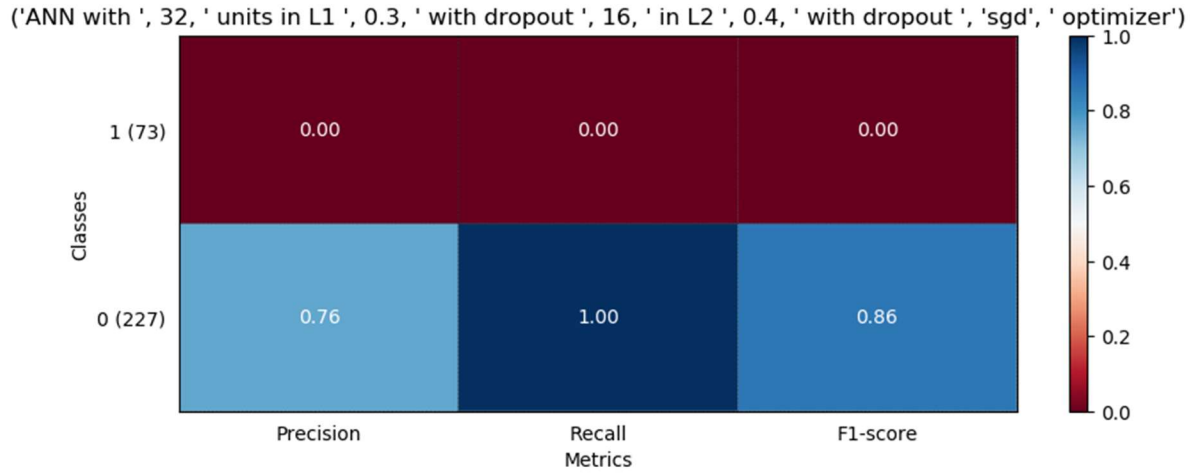
```
def ANN_with_gridSearch(X_train, X_test, y_train, y_test):
    HP_NUM_UNITS_1 = hp.HParam('num_units_1', hp.Discrete(list(range(32,33))))
    HP_DROPOUT_1 = hp.HParam('dropout_1', hp.RealInterval(0.3, 0.4))
    HP_NUM_UNITS_2 = hp.HParam('num_units_2', hp.Discrete(list(range(16,17))))
    HP_DROPOUT_2 = hp.HParam('dropout_2', hp.RealInterval(0.3, 0.4))
    HP_OPTIMIZER = hp.HParam('optimizer', hp.Discrete(['adam','sgd']))
    METRIC_ACCURACY = 'accuracy'
    hparams=[HP_NUM_UNITS_1, HP_DROPOUT_1,HP_NUM_UNITS_2, HP_DROPOUT_2, HP_OPTIMIZER]
    hparams=[HP_NUM_UNITS_1,HP_NUM_UNITS_2,  HP_OPTIMIZER]
    metrics=[hp.Metric(METRIC_ACCURACY, display_name='Accuracy')]
```

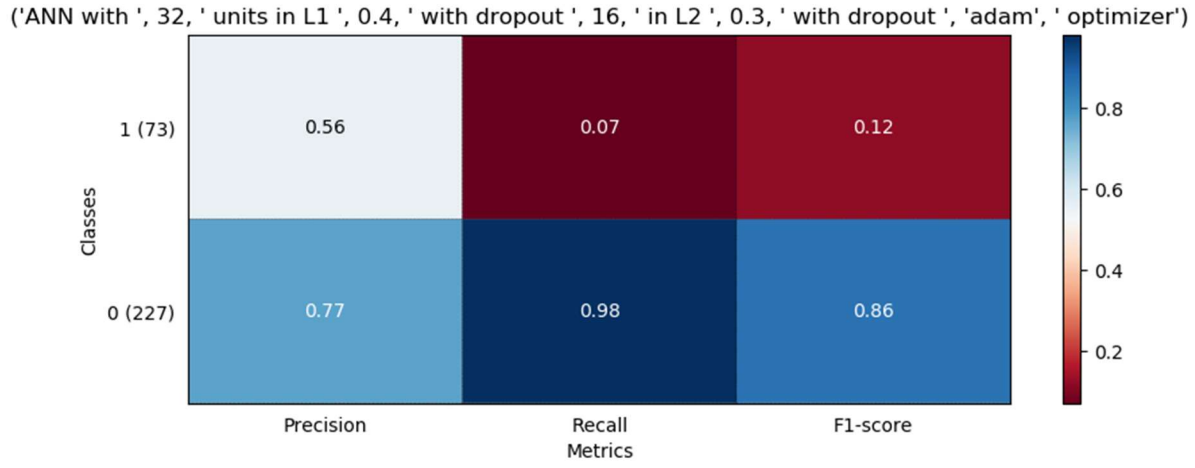
As shown in figure, 2 layers are used and in both layers have used Dropout regularization (instead of L2- Frohenius regularization). Also we can state range of units in hidden layers in it. Have tried it on 2 optimizers and Adams was giving more accuracy.

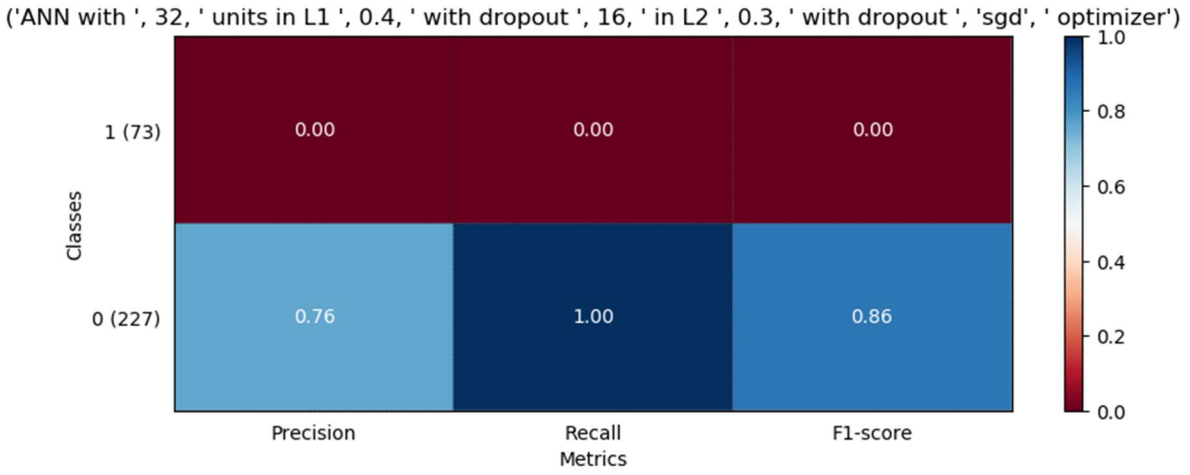
```python
def train_test_model(hparams,X_train,X_test, y_train,  y_test):
  model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(hparams['num_units_1'], activation=tf.nn.relu),
    tf.keras.layers.Dropout(hparams['dropout_1']),
    #tf.keras.layers.Dense(hparams['num_units_2'],kernel_regularizer=keras.regularizers.l2(0.1), activation=tf.nn.relu),
    tf.keras.layers.Dense(hparams['num_units_2'],activation=tf.nn.relu),
    tf.keras.layers.Dropout(hparams['dropout_2']),
    tf.keras.layers.Dense(1, activation=tf.nn.sigmoid),
    tf.keras.layers.Flatten()
  ])
```

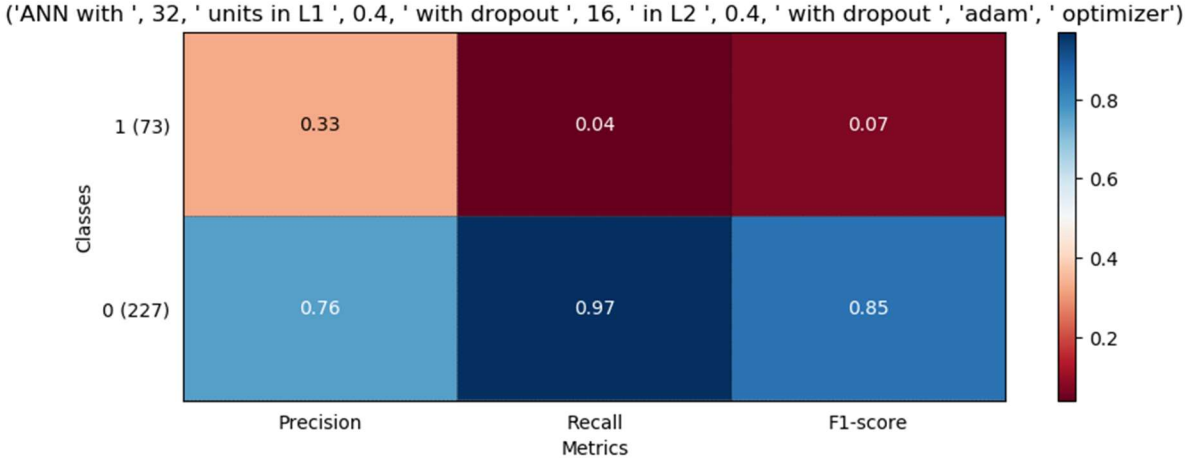
As shown in above figure, we can also use other regularization technique. (commented out lines)

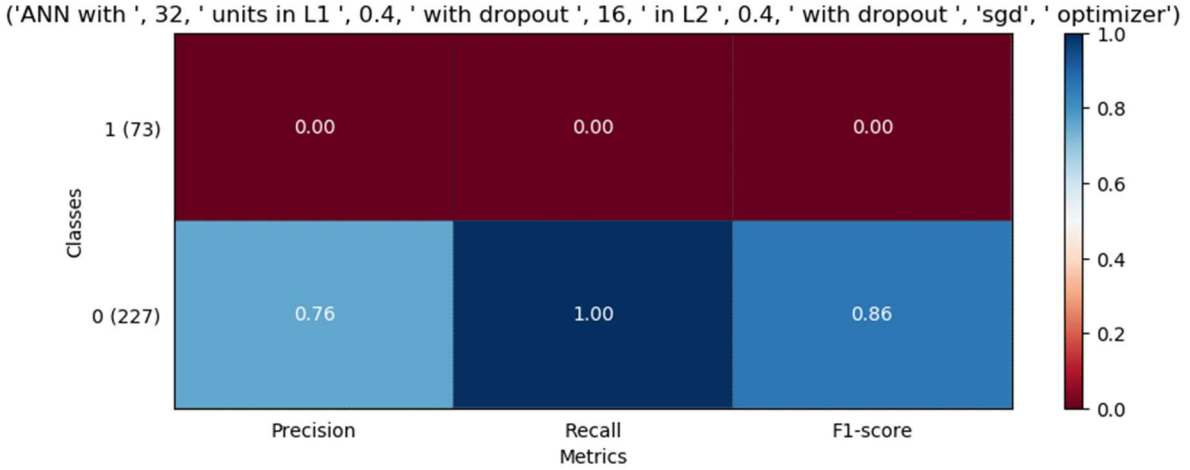
Classification report for different Classifiers:

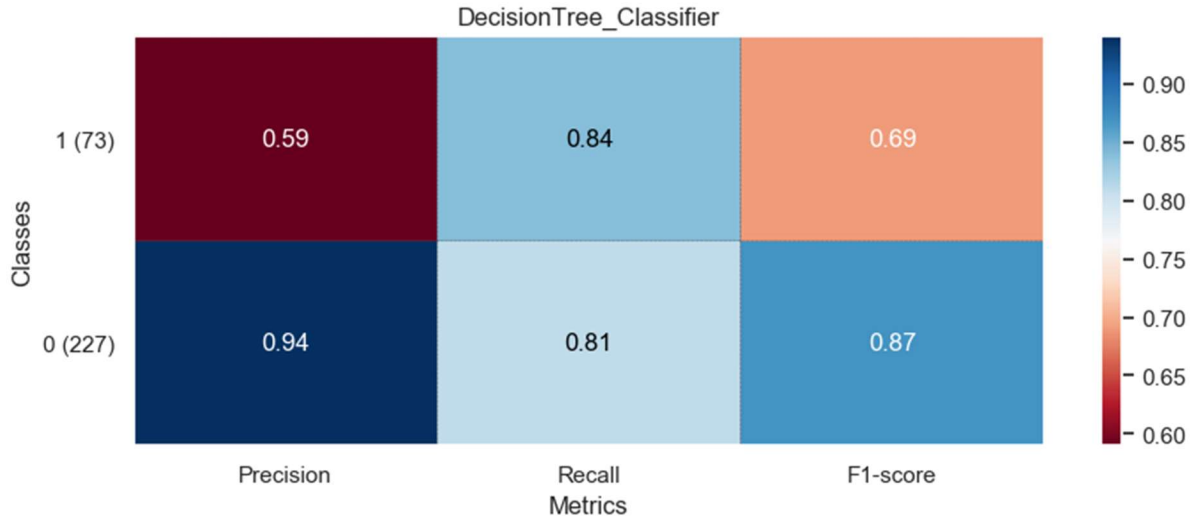


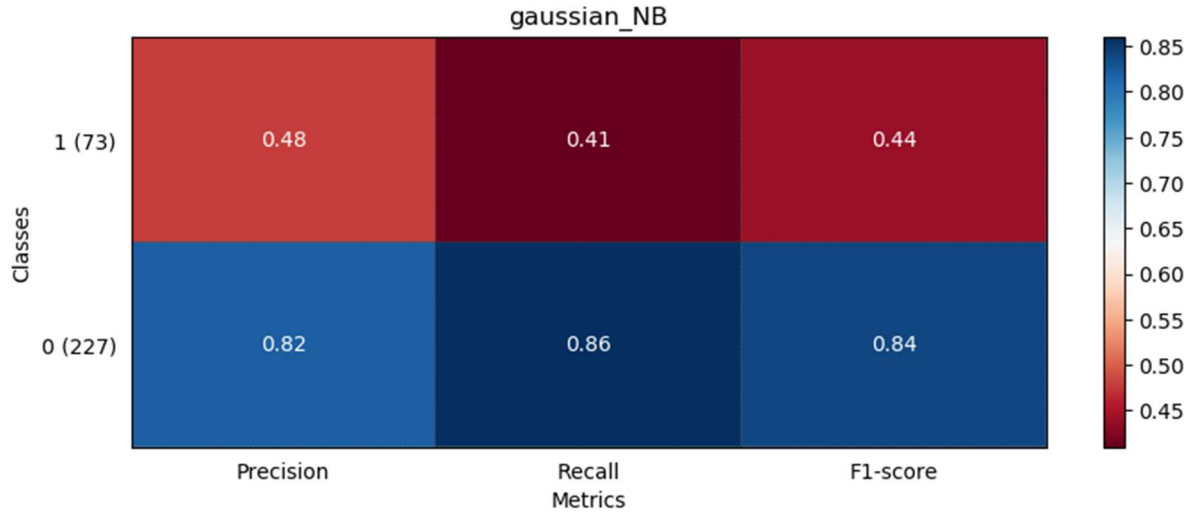


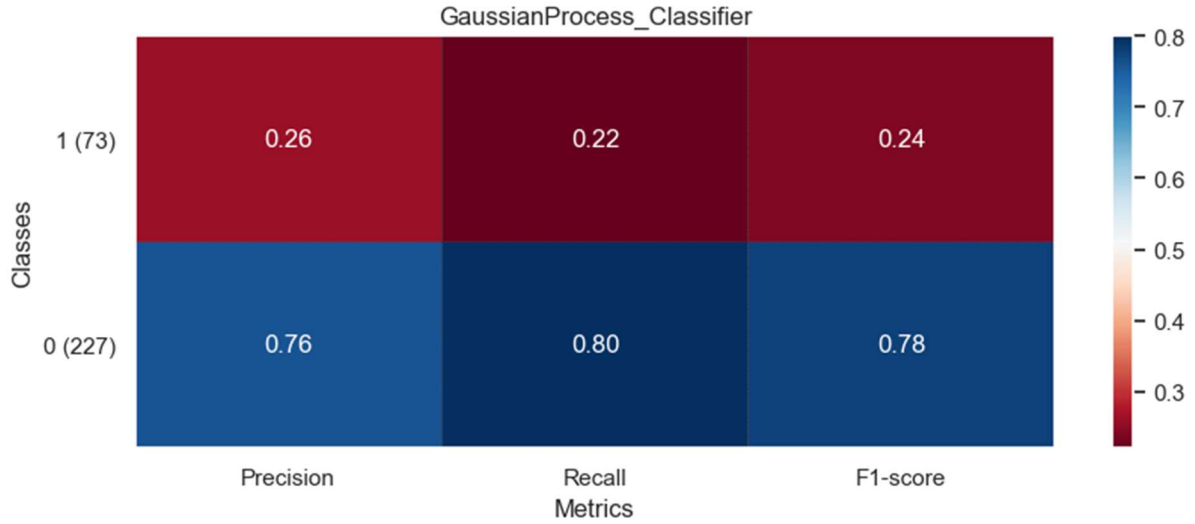

('ANN with ', 32, ' units in L1 ', 0.4, ' with dropout ', 16, ' in L2 ', 0.3, ' with dropout ', 'sgd', ' optimizer')
1 (73)
0 (227)
Classes
0.00
0.00
0.00
0.76
1.00
0.86
Precision
Recall
F1-score
Metrics
1.0
0.8
0.6
0.4
0.2
0.0

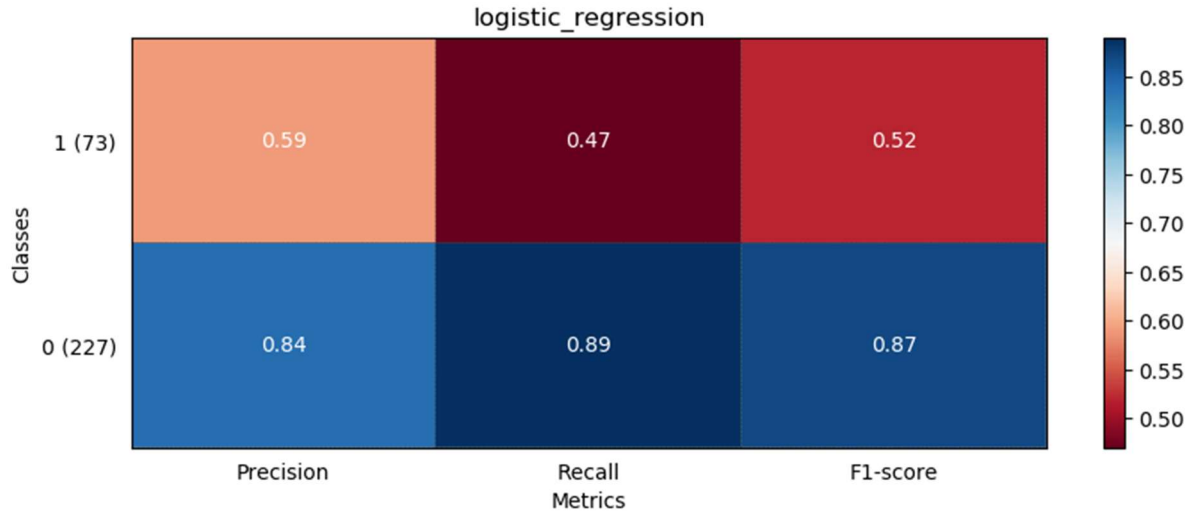

('ANN with ', 32, ' units in L1 ', 0.4, ' with dropout ', 16, ' in L2 ', 0.4, ' with dropout ', 'adam', ' optimizer')
1 (73)
0 (227)
Classes
0.33
0.04
0.07
0.76
0.97
0.85
Precision
Recall
F1-score
Metrics
0.8
0.6
0.4
0.2

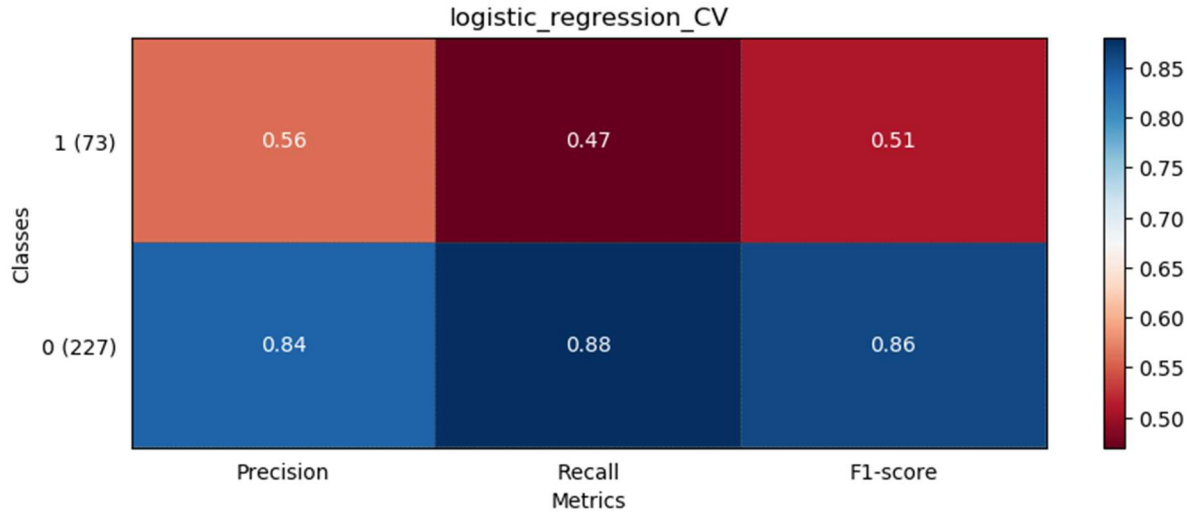

('ANN with ', 32, ' units in L1 ', 0.4, ' with dropout ', 16, ' in L2 ', 0.4, ' with dropout ', 'sgd', ' optimizer')
1 (73)
0 (227)
Classes
0.00
0.00
0.00
0.76
1.00
0.86
Precision
Recall
F1-score
Metrics
1.0
0.8
0.6
0.4
0.2
0.0

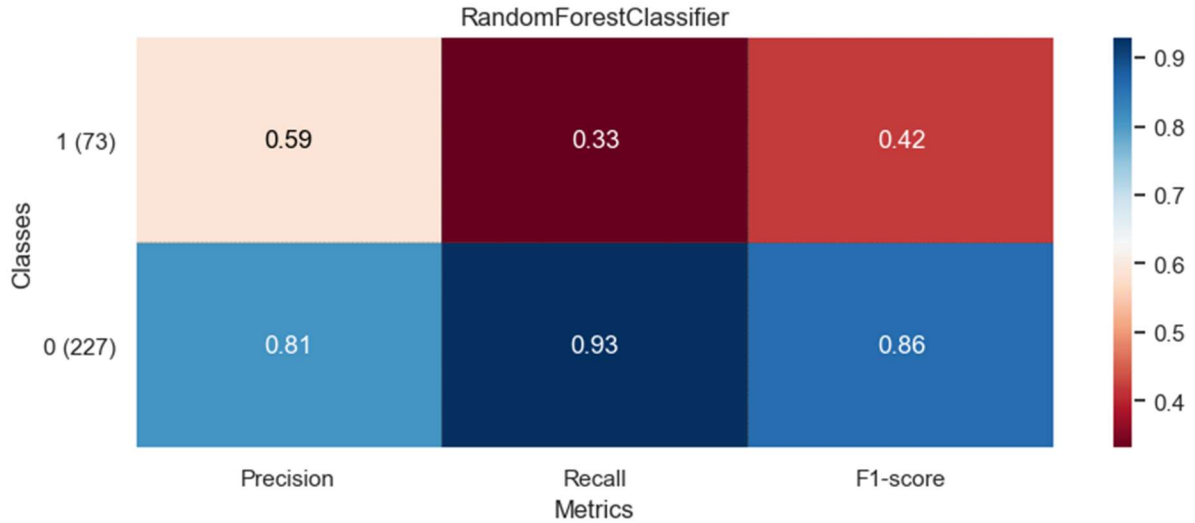

DecisionTree_Classifier
1 (73)
0 (227)
Classes
0.59
0.84
0.69
0.94
0.81
0.87
Precision
Recall
F1-score
Metrics
0.90
0.85
0.80
0.75
0.70
0.65
0.60

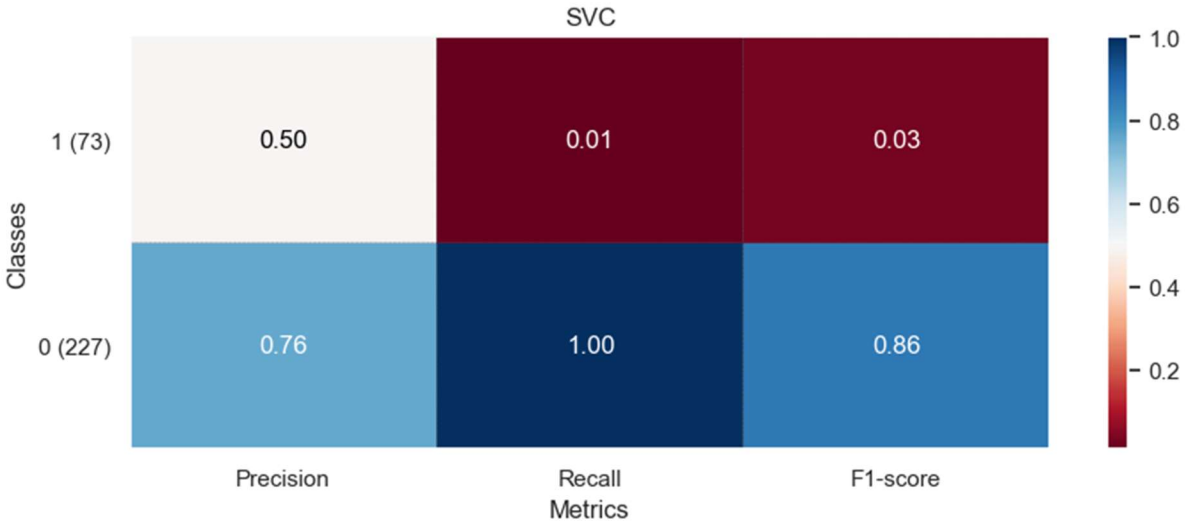

gaussian_NB
1 (73)
0 (227)
Classes
0.48
0.41
0.44
0.82
0.86
0.84
Precision
Recall
F1-score
Metrics
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45


GaussianProcess_Classifier
1 (73)
0 (227)
Classes
0.26
0.22
0.24
0.76
0.80
0.78
Precision
Recall
F1-score
Metrics
0.8
0.7
0.6
0.5
0.4
0.3

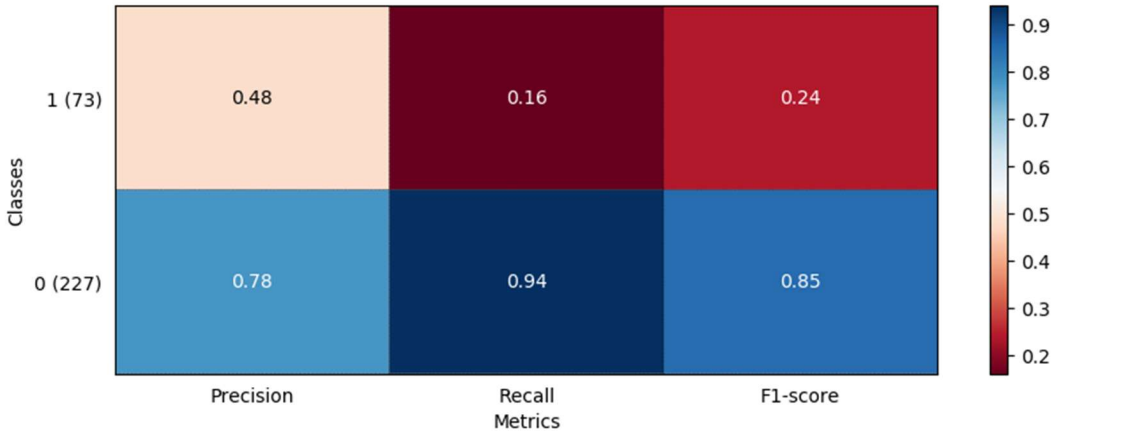

logistic_regression
1 (73)
0 (227)
Classes
0.59
0.47
0.52
0.84
0.89
0.87
Precision
Recall
F1-score
Metrics
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50

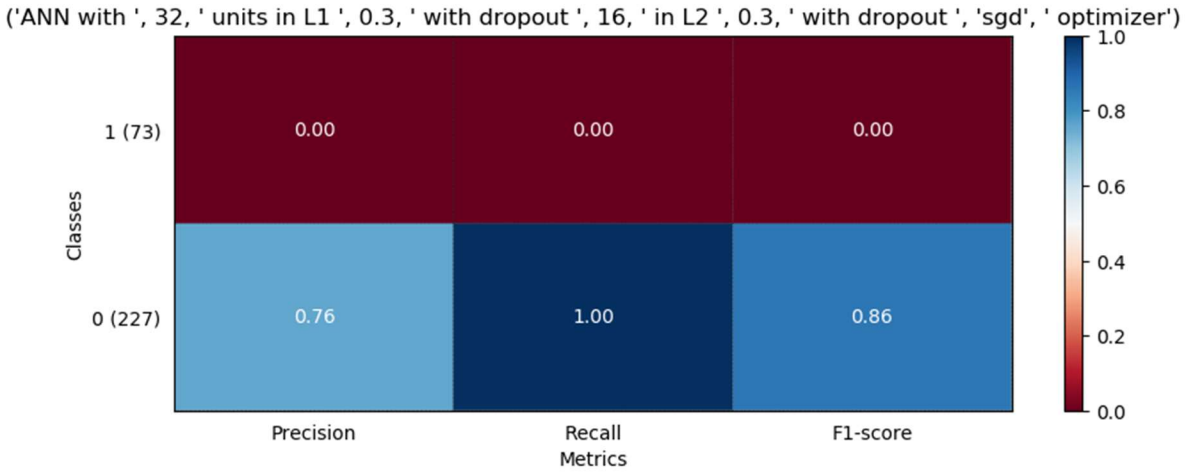

logistic_regression_CV
1 (73)
0 (227)
Classes
0.56
0.47
0.51
0.84
0.88
0.86
Precision
Recall
F1-score
Metrics
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50

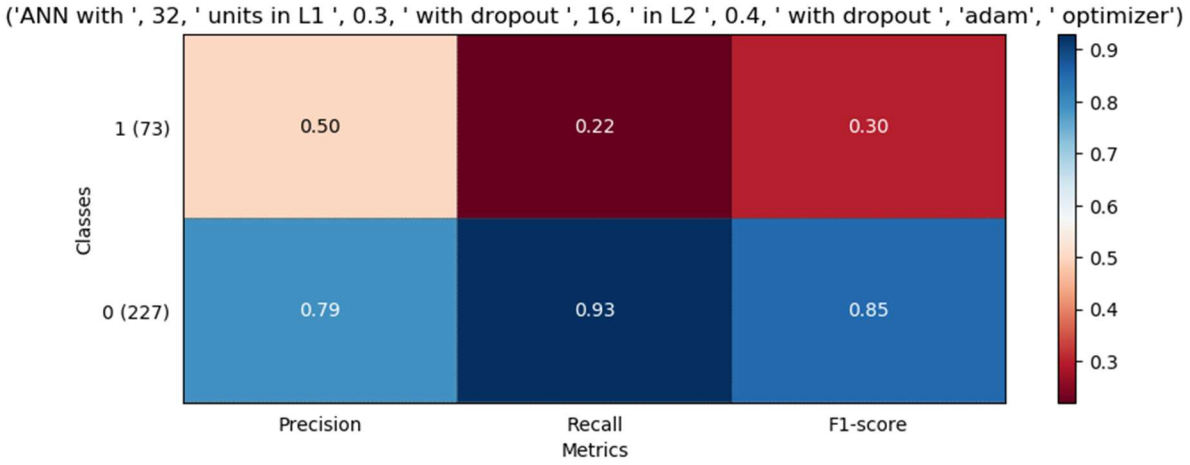

RandomForestClassifier
1 (73)
0 (227)
Classes
0.59
0.33
0.42
0.81
0.93
0.86
Precision
Recall
F1-score
Metrics
0.9
0.8
0.7
0.6
0.5
0.4



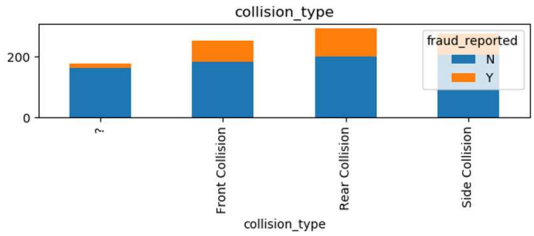
('ANN with ', 32, ' units in L1 ', 0.3, ' with dropout ', 16, ' in L2 ', 0.3, ' with dropout ', 'adam', ' optimizer')

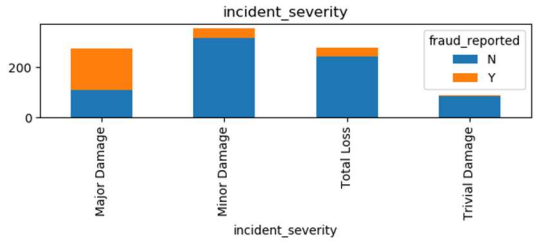


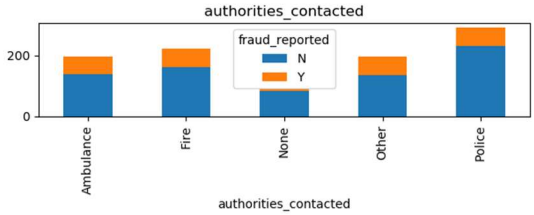


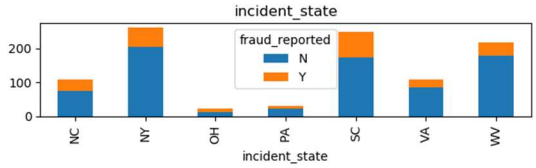


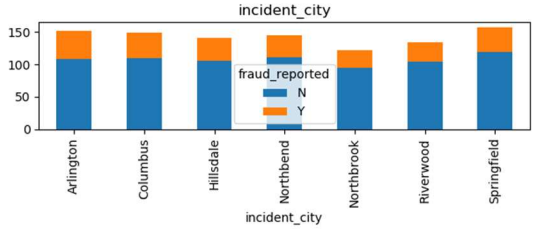
Now data visualization of all features

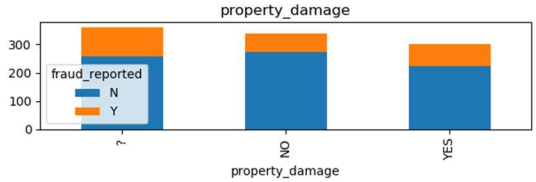

collision_type
fraud_reported
N
Y
200
0
?
Front Collision
Rear Collision
Side Collision
collision_type

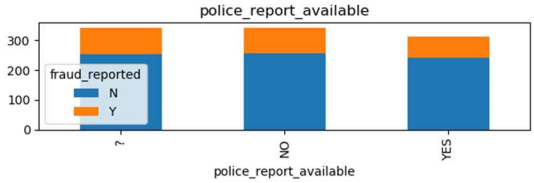

incident_severity
fraud_reported
N
Y
200
0
Major Damage
Minor Damage
Total Loss
Trivial Damage
incident_severity

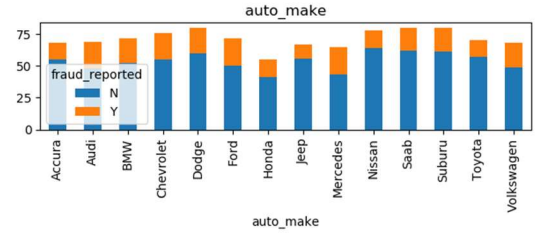

authorities_contacted
fraud_reported
N
Y
200
0
Ambulance
Fire
None
Other
Police
authorities_contacted

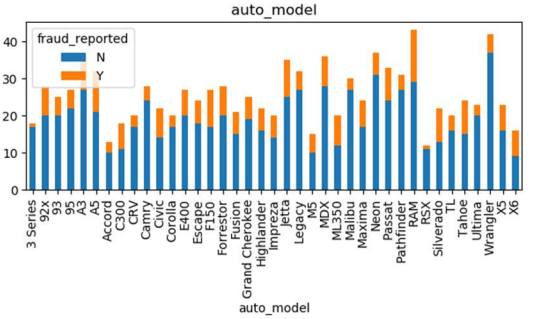

incident_state
fraud_reported
N
Y
200
100
0
NC
NY
OH
PA
SC
VA
WV
incident_state


incident_city
fraud_reported
N
Y
150
100
50
0
Arlington
Columbus
Hillsdale
Northbend
Northbrook
Riverwood
Springfield
incident_city

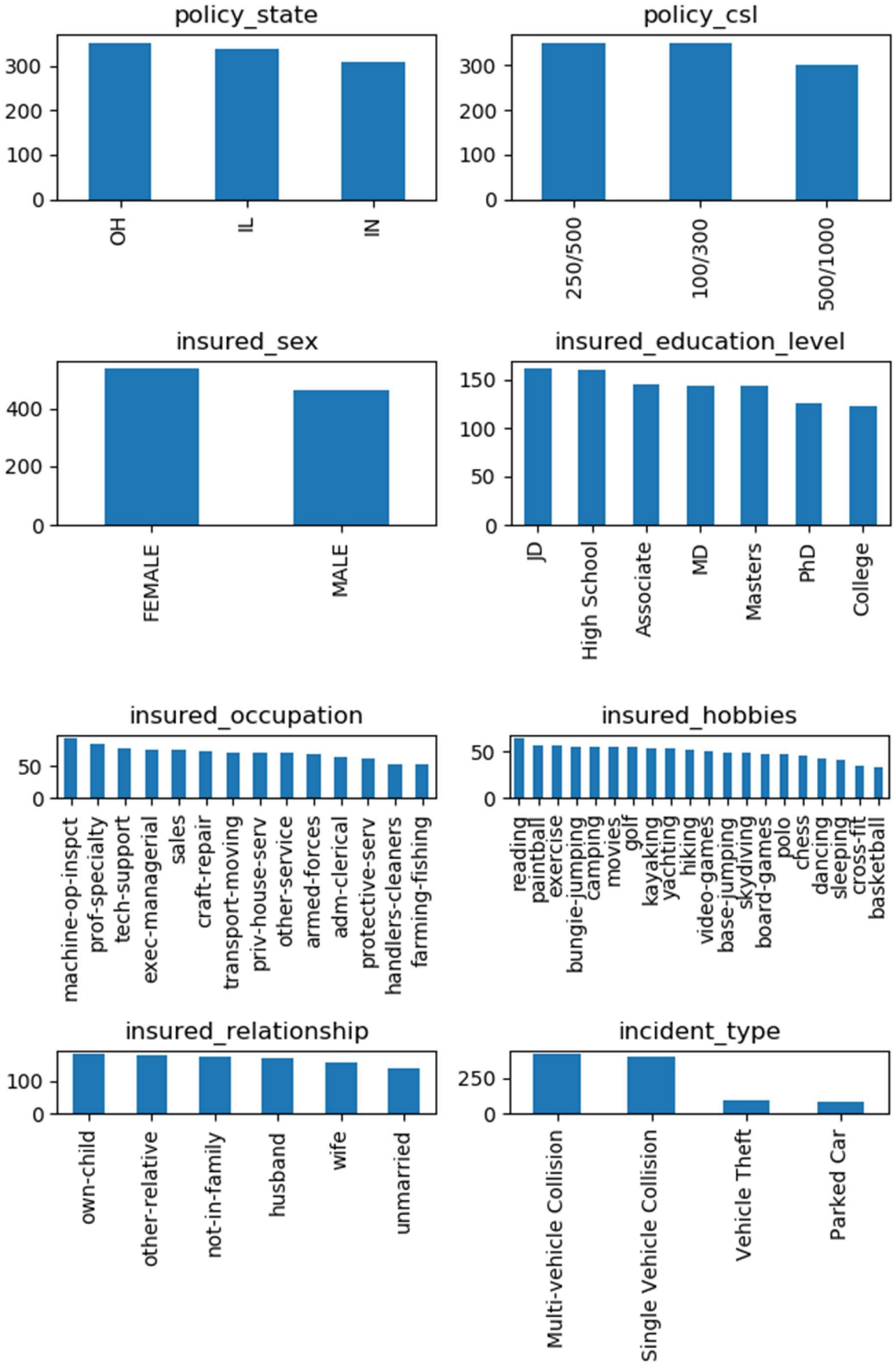

property_damage
fraud_reported
N
Y
300
200
100
0
?
NO
YES
property_damage

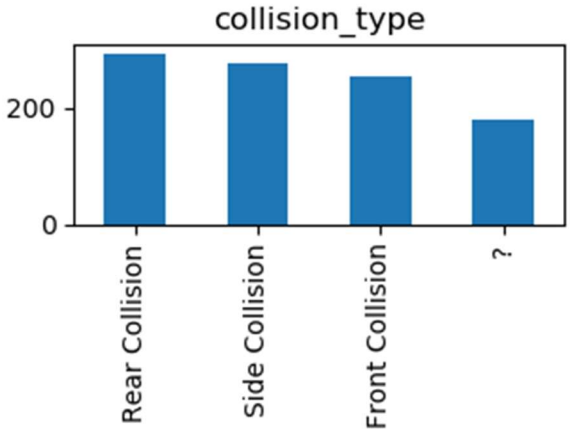

police_report_available
fraud_reported
N
Y
300
200
100
0
?
NO
YES
police_report_available

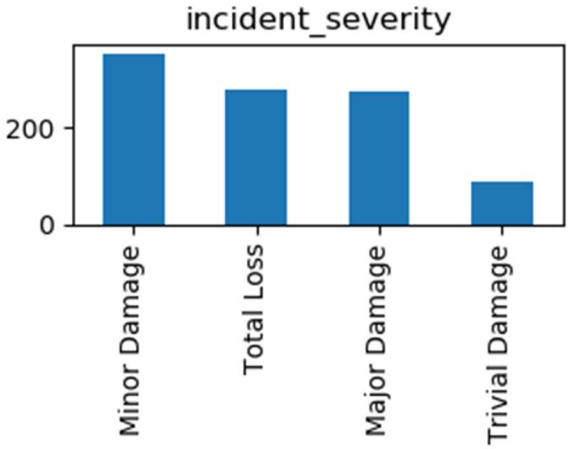

auto_make
fraud_reported
N
Y
75
50
25
0
Accura
Audi
BMW
Chevrolet
Dodge
Ford
Honda
Jeep
Mercedes
Nissan
Saab
Suburu
Toyota
Volkswagen
auto_make

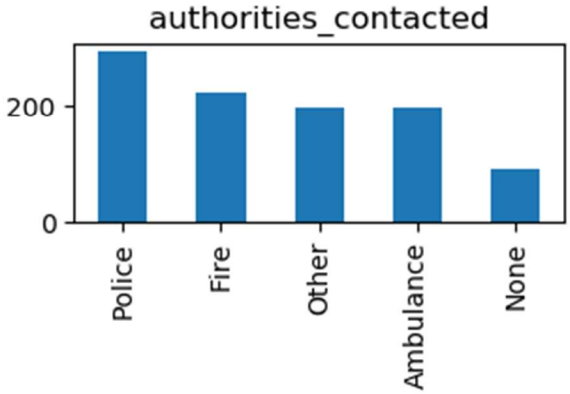

auto_model
fraud_reported
N
Y
40
30
20
10
0
3 Series
92x
93
95
A3
A5
Accord
C300
CRV
Camry
Civic
Corolla
E400
Escape
F150
Forrestor
Fusion
Grand Cherokee
Highlander
Impreza
Jetta
Legacy
M5
MDX
ML350
Malibu
Maxima
Neon
Passat
Pathfinder
RAM
RSX
Silverado
TL
Tahoe
Ultima
Wrangler
X5
X6
auto_model

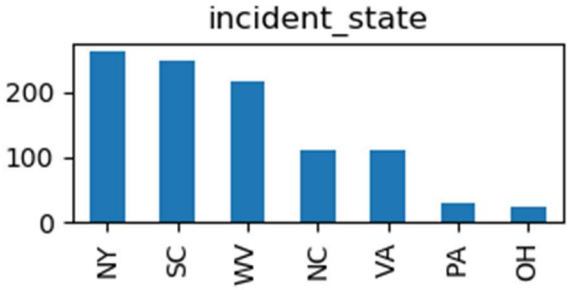

fraud_reported
fraud_reported
N
Y
600
400
200
0
N
Y
fraud_reported


policy_state
300
200
100
0
OH
IL
IN
policy_csl
300
200
100
0
250/500
100/300
500/1000
insured_sex
400
200
0
FEMALE
MALE
insured_education_level
150
100
50
0
JD
High School
Associate
MD
Masters
PhD
College
insured_occupation
50
0
machine-op-inspct
prof-specialty
tech-support
exec-managerial
sales
craft-repair
transport-moving
priv-house-serv
other-service
armed-forces
adm-clerical
protective-serv
handlers-cleaners
farming-fishing
insured_hobbies
50
0
reading
paintball
exercise
bungie-jumping
camping
movies
golf
kayaking
yachting
hiking
video-games
base-jumping
skydiving
board-games
polo
chess
dancing
sleeping
cross-fit
basketball
insured_relationship
100
0
own-child
other-relative
not-in-family
husband
wife
unmarried
incident_type
250
0
Multi-vehicle Collision
Single Vehicle Collision
Vehicle Theft
Parked Car

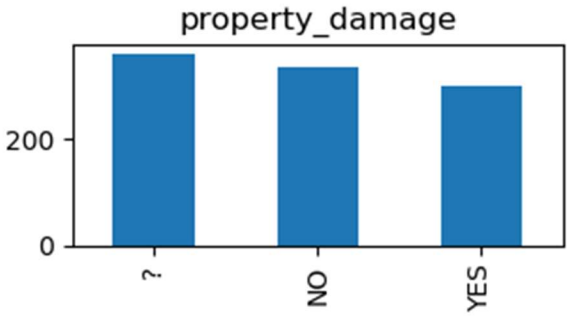

collision_type
200
0
Rear Collision
Side Collision
Front Collision
?

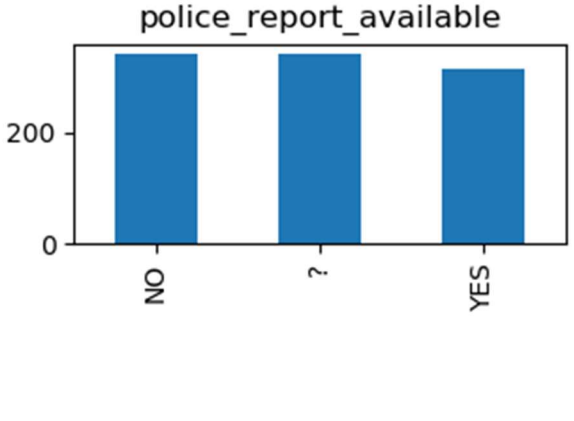

incident_severity
200
0
Minor Damage
Total Loss
Major Damage
Trivial Damage

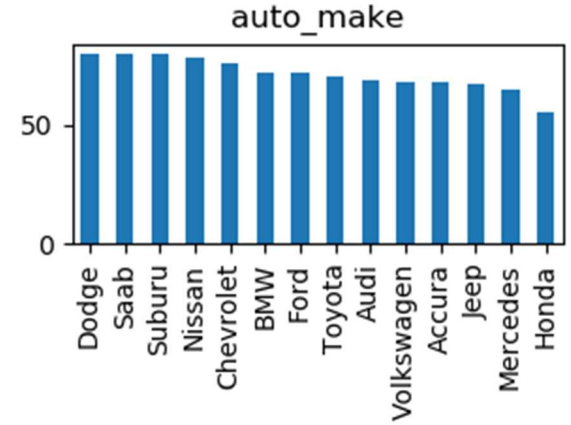

authorities_contacted
200
0
Police
Fire
Other
Ambulance
None

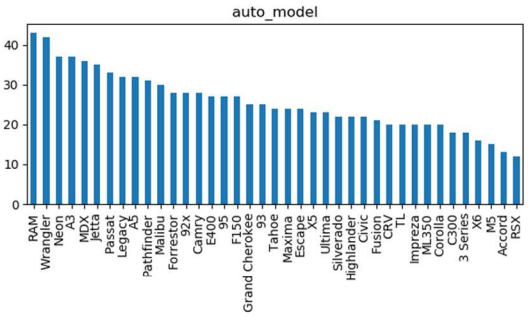

incident_state
200
100
0
NY
SC
WV
NC
VA
PA
OH

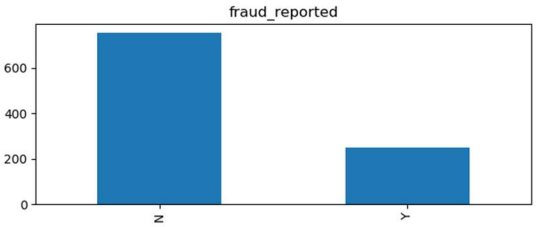

incident_city
100
0
Springfield
Arlington
Columbus
Northbend
Hillsdale
Riverwood
Northbrook

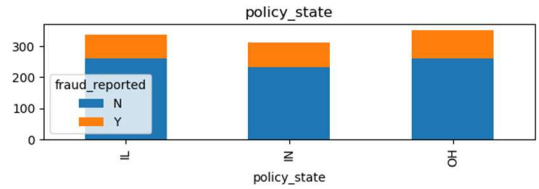

property_damage
200
0
?
NO
YES

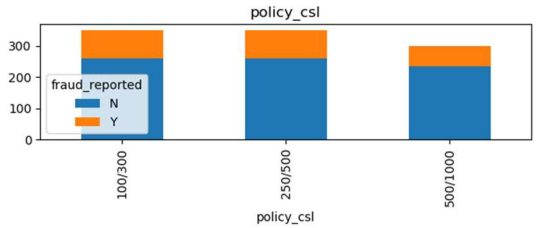

police_report_available
200
0
NO
?
YES


auto_make
50
0
Dodge
Saab
Suburu
Nissan
Chevrolet
BMW
Ford
Toyota
Audi
Volkswagen
Accura
Jeep
Mercedes
Honda

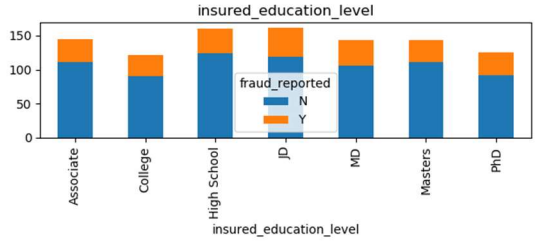

auto_model
RAM
Wrangler
Neon
A3
MDX
Jetta
Passat
Legacy
A5
Pathfinder
Malibu
Forrestor
92x
Camry
E400
95
F150
Grand Cherokee
93
Tahoe
Maxima
Escape
X5
Ultima
Silverado
Highlander
Civic
Fusion
CRV
TL
Impreza
ML350
Corolla
C300
3 Series
X6
M5
Accord
RSX

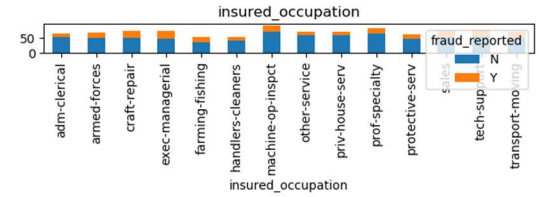

fraud_reported
N
Y

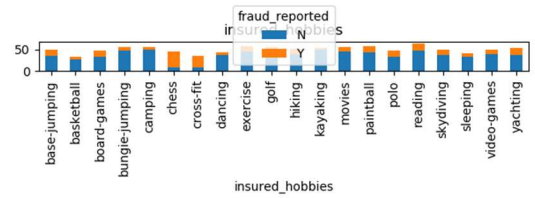

policy_state
fraud_reported
N
Y
IL
IN
OH
policy_state

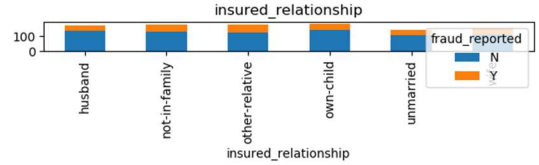

policy_csl
fraud_reported
N
Y
100/300
250/500
500/1000
policy_csl

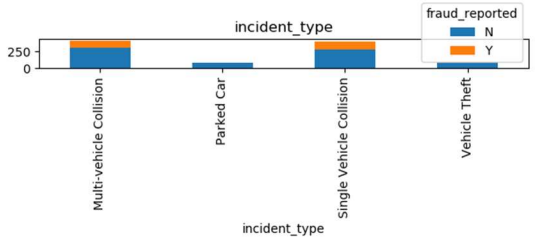

insured_sex
fraud_reported
N
Y
FEMALE
MALE
insured_sex


insured_education_level
fraud_reported
N
Y
Associate
College
High School
JD
MD
Masters
PhD
insured_education_level

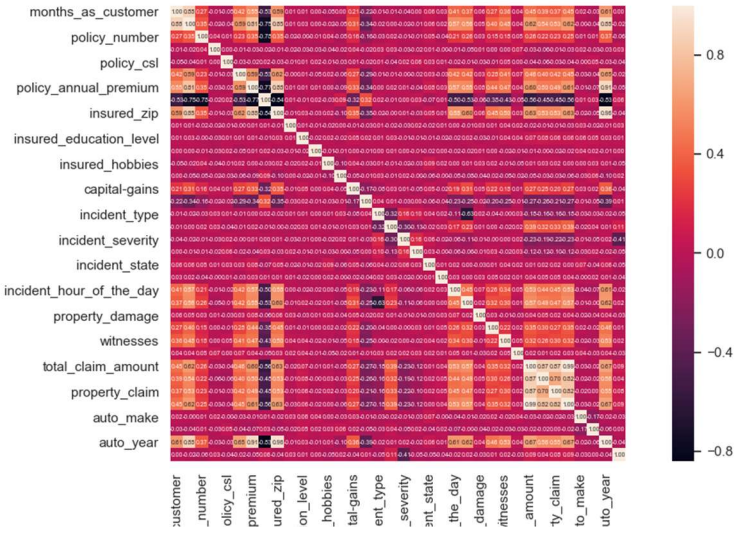

insured_occupation
fraud_reported
N
Y
adm-clerical
armed-forces
craft-repair
exec-managerial
farming-fishing
handlers-cleaners
machine-op-inspct
other-service
priv-house-serv
prof-specialty
protective-serv
tech-sup
transport-mo
insured_occupation


fraud_reported
N
Y
base-jumping
basketball
board-games
bungie-jumping
camping
chess
cross-fit
dancing
exercise
golf
hiking
kayaking
movies
paintball
polo
reading
skydiving
sleeping
video-games
yachting
insured_hobbies

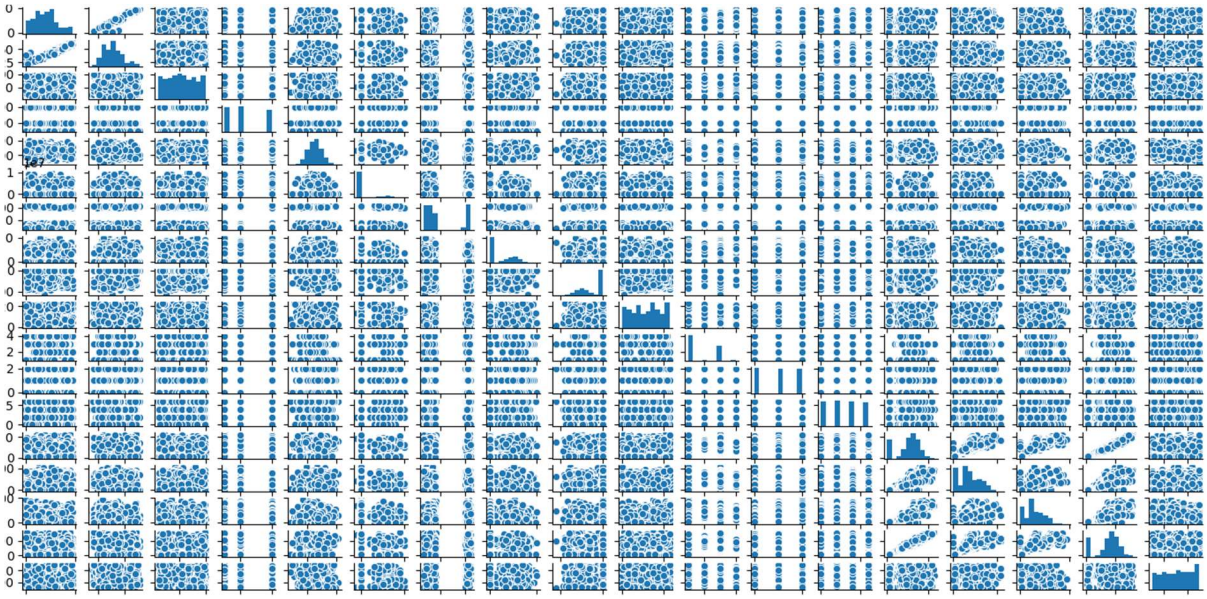

insured_relationship
fraud_reported
N
Y
husband
not-in-family
other-relative
own-child
unmarried
insured_relationship


incident_type
fraud_reported
N
Y
Multi-vehicle Collision
Parked Car
Single Vehicle Collision
Vehicle Theft
incident_type

HEAT MAP



Pairplot



Have used Label encoder to encode features:

```
le = preprocessing.LabelEncoder()
for column in col_ob_dt:
    insurance_csv[column] = le.fit_transform(insurance_csv[column].astype(str))
    print(column,'         ',le.classes_)
```

Normalizing data for greater performance

```
def preprocess_data(insurance_csv,col_ob_dt):
    col_to_normalize = list( set(insurance_csv.columns) -  set(col_ob_dt))
    insurance_csv[col_to_normalize] = preprocessing.normalize(insurance_csv[col_to_normalize])
    insurance_csv.to_csv('Sample1.csv')
    return insurance_csv
```

FUTURE scope:

- To have positive correct matrix
- To use OverSampling since we have less data
- To try more NN schema and hyper parameters with Oversampling