5️⃣

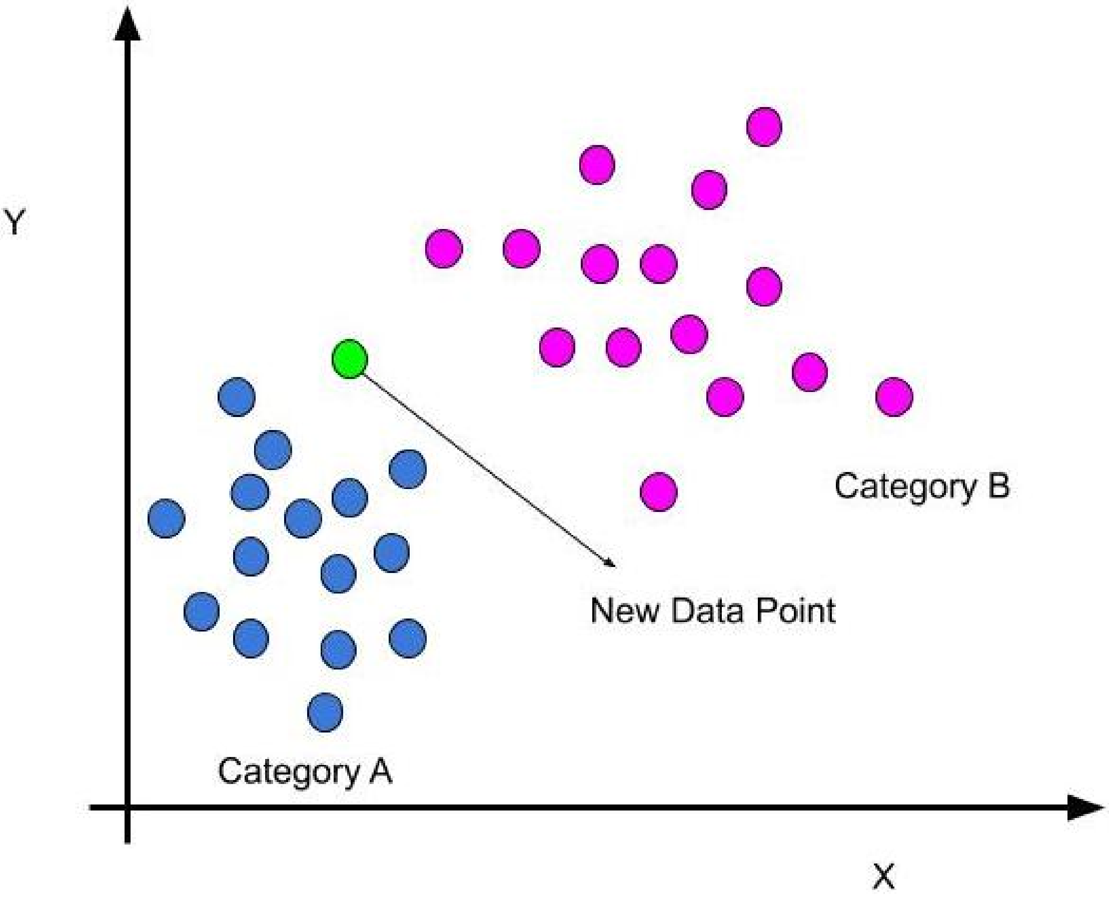
ML-Classification

Brainstorming:

- ☑ ~~Benchmark between algorithms.~~
- ☑ ~~Different algorithms used in classification.~~
- ☑ ~~We will cover why classification is used, its benefits.~~
- ☑ ~~Project: Insurance binary classification from end to end.~~

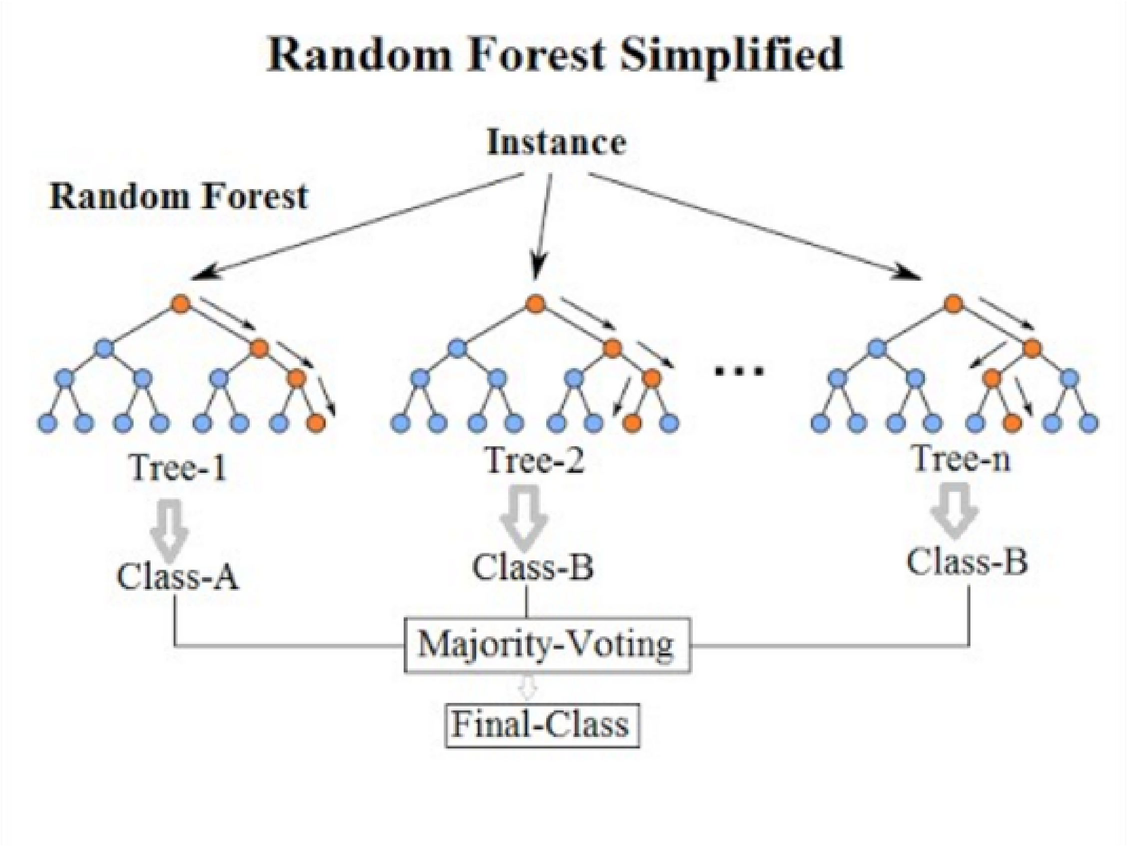
Classification Models:

- K Nearest Neighbor:

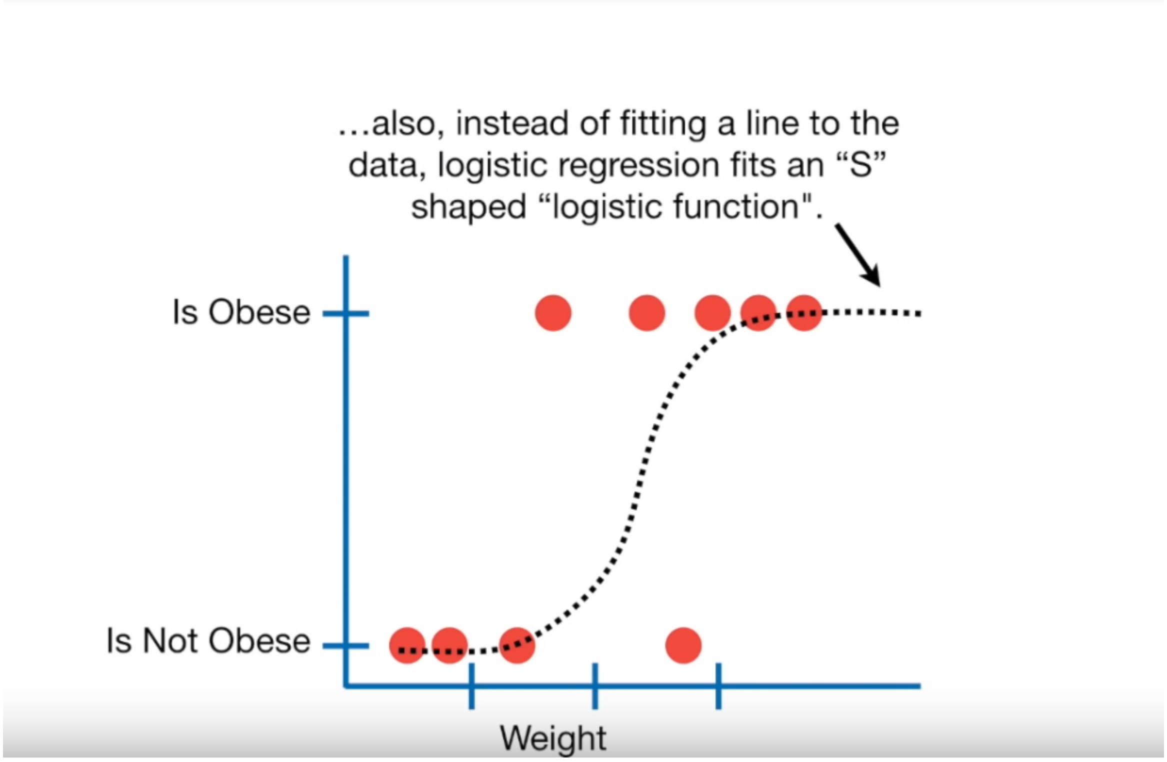


- We usually use KNN is if we are facing issues like (Imbalance data/outliers/categorical data).
- hat is the K?
  - It means how many points I’ll take from my groups to decide where to put my new data.
- How it works?
  - I mainly calculcate the Eucledian Distance between K points and my new data(point in green), then I decide where to integrate mine(based on the priority/short distance).

- Random Forest:



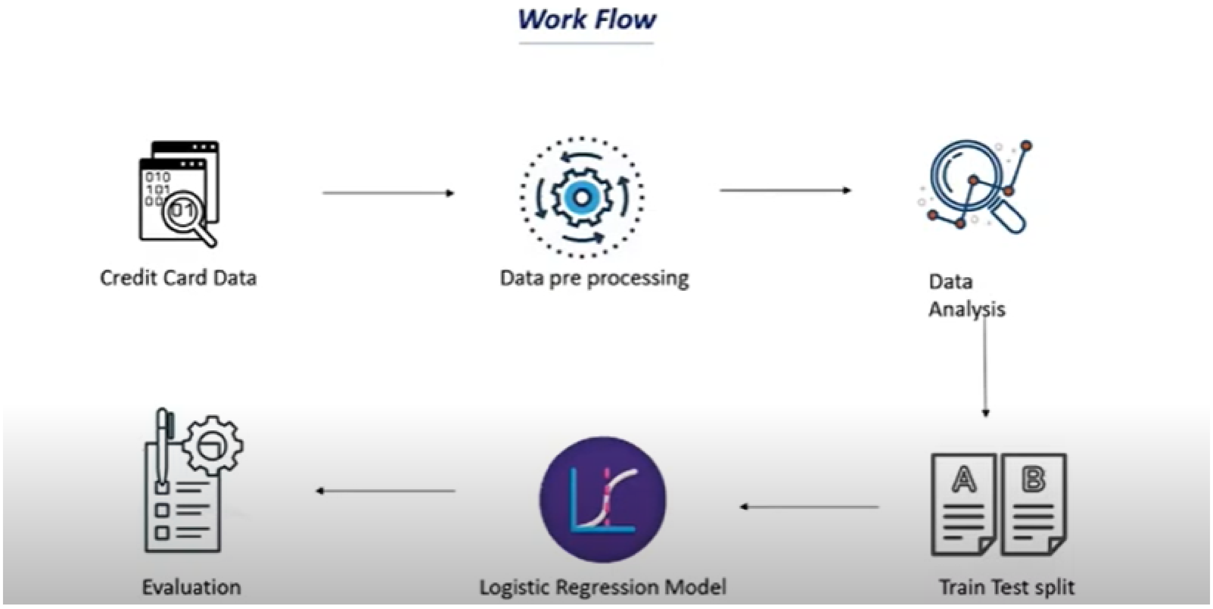
  - Random Forest is a combination of decision trees, and a method called “Bagging” to give one decision.
  - Why Random Forest over A decision tree? The combination increases results.
- Decision Tree(Small part of random forest)
- Logistic Regression.



Project

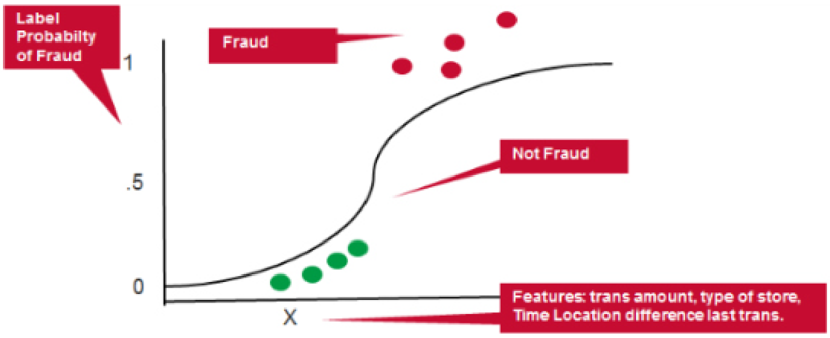
Fraud Detection With Logistic Regression:

- Project Pipeline



Classification methods commonly used for fraud detection

- Logistic regression


Label Probabilty of Fraud
1
Fraud
.5
Not Fraud
0
X
Features: trans amount, type of store, Time Location difference last trans.

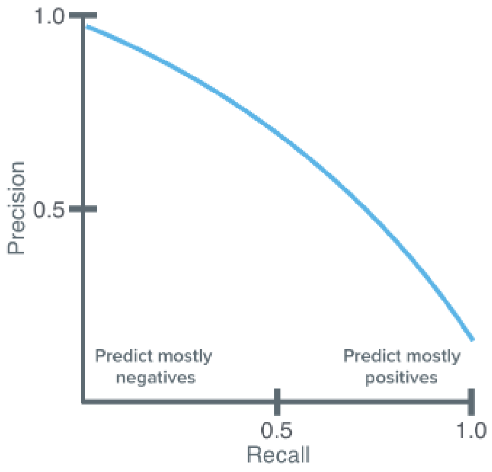
- ML Syntax:

scikit-learn syntax

```
from sklearn.module import Model
model = Model()
model.fit(X, y)
predictions = model.predict(X_new)
print(predictions)
```

- `from sklearn.linear_model LogisticRegression`
- Metrics:
  - F1-score:
  - Recall:
  - Support:
  - Precision:

Precision-recall tradeoff



$$Precision = \frac{\#True\ Positives}{\#True\ Positives + \#False\ Positives}$$

$$Recall = \frac{\#True\ Positives}{\#True\ Positives + \#False\ Negatives}$$

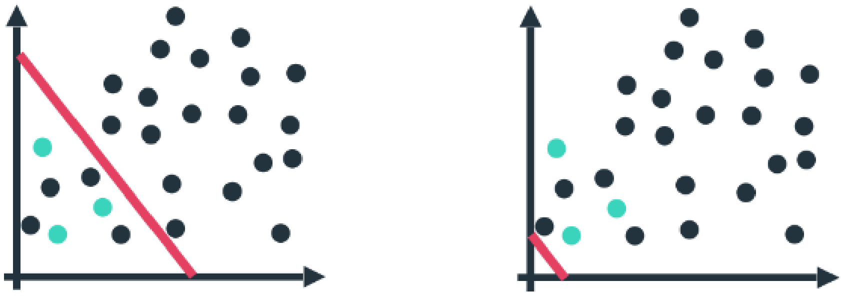
$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$= \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Accuracy:

Accuracy isn't everything

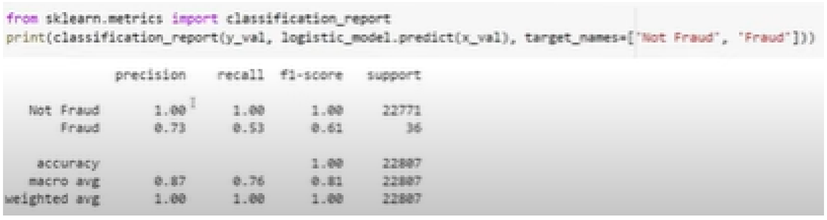
Throw accuracy out of the window when working on fraud detection problems



Accuracy = 93%

Accuracy = 97%

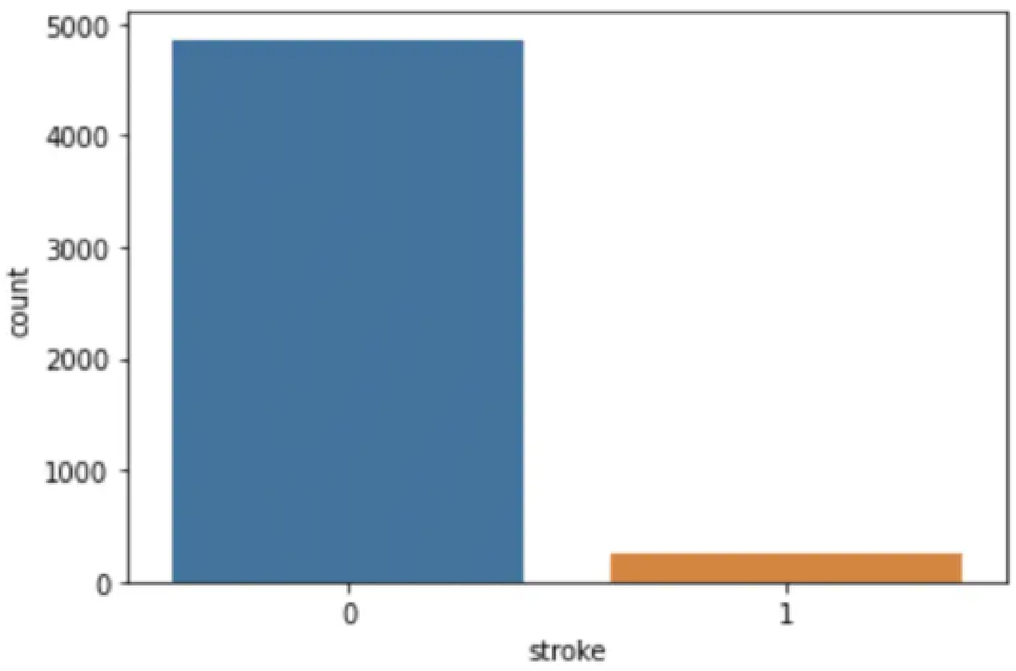
- Imbalance data issue:



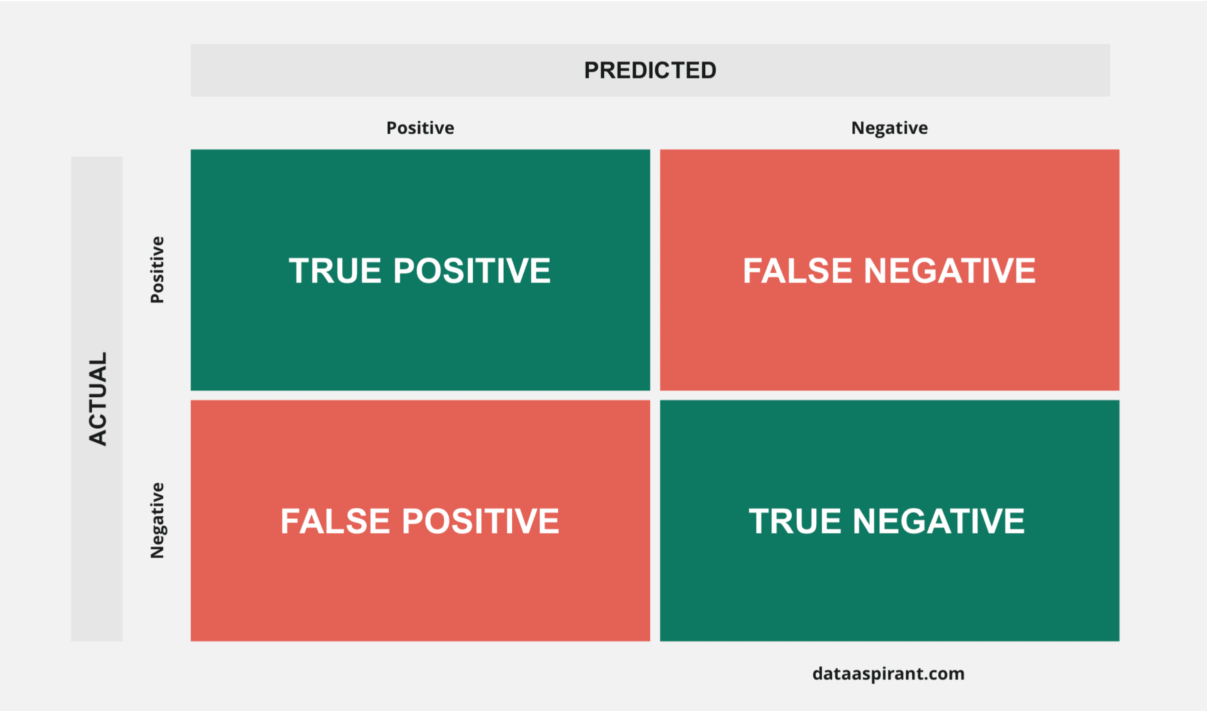
```
from sklearn.metrics import classification_report
print(classification_report(y_val, logistic_model.predict(x_val), target_names=['Not Fraud', 'Fraud']))
```

```
              precision    recall  f1-score   support

   Not Fraud       1.00      1.00      1.00     22771
       Fraud       0.73      0.53      0.61        36

    accuracy                           1.00     22807
   macro avg       0.87      0.76      0.81     22807
weighted avg       1.00      1.00      1.00     22807
```

```
sns.countplot(y)
```



- Confusion Matrix:



		PREDICTED	
		Positive	Negative
ACTUAL	Positive	TRUE POSITIVE	FALSE NEGATIVE
	Negative	FALSE POSITIVE	TRUE NEGATIVE

dataaspirant.com

- Notebook Link: https://colab.research.google.com/drive/1IkgASl8Uy4_J7RWHu_1xDquMRwqAnwIG?usp=sharing