Session-8
ML Regression and Classification

Agenda

Lecture -8: Building Machine Learning Model with Python

❑ Building Regression Models – Final Part

❑ Building classification models

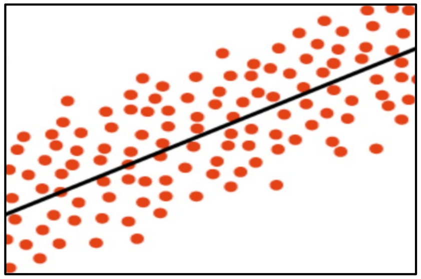
❑ Review assignment solutions

❑ Guidelines on the final exam

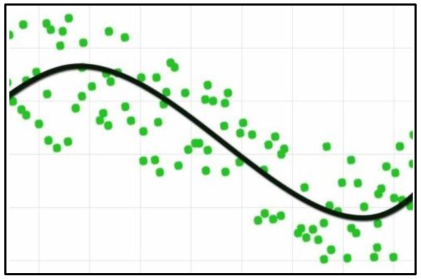
Machine Learning - Classification

- With Classification algorithm a program learns from the existing dataset or observations and then classifies new observation into a number of classes or groups like - Spam email or Not Spam, Churn or not a churn, etc.
- With Regression, we try to predict an amount or a number like the home price, stock price, or sales amount etc.
- With Classification, we try to predict things like whether a loan will be paid or not; or we try to detect things like whether an email is a spam or not.

Regression

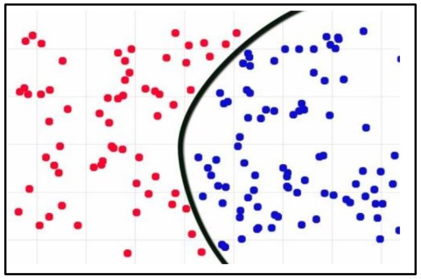


Linear

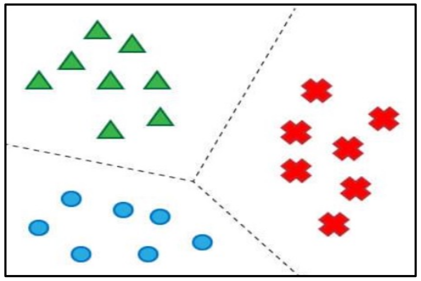


Non-Linear

Classification



Binary



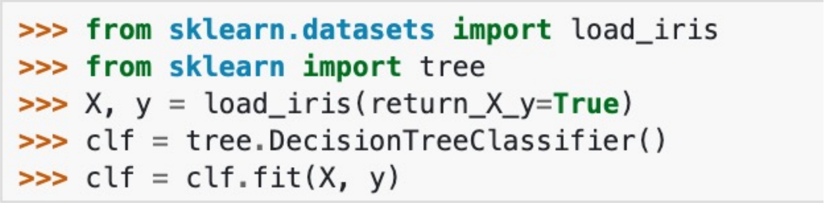
Multi-class

Classification Algorithm – Decision Tree

- ❑ **Decision Trees** are a non-parametric supervised learning method used for classification and regression

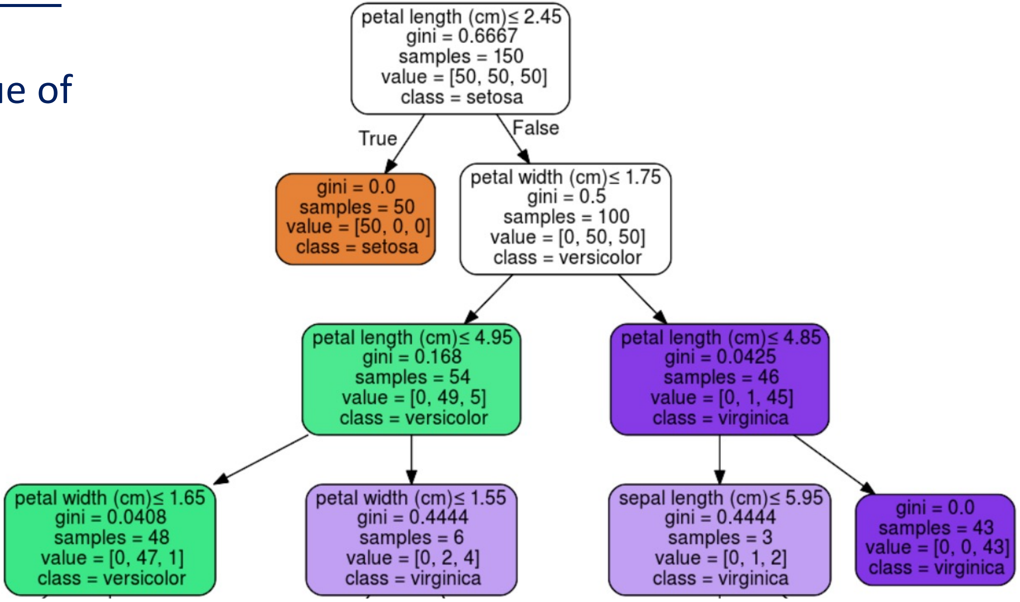
- ❑ The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

- ❑ A tree can be seen as a piecewise constant approximation.



```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> X, y = load_iris(return_X_y=True)
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(X, y)
```

Decision Tree – Classification Example

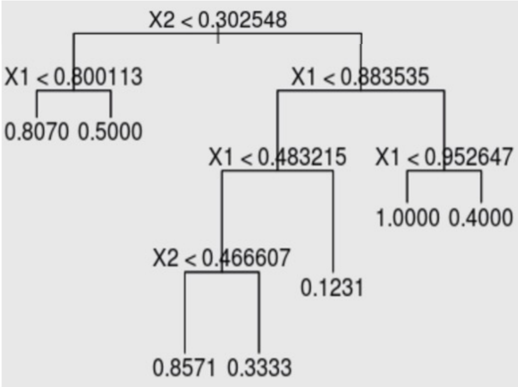


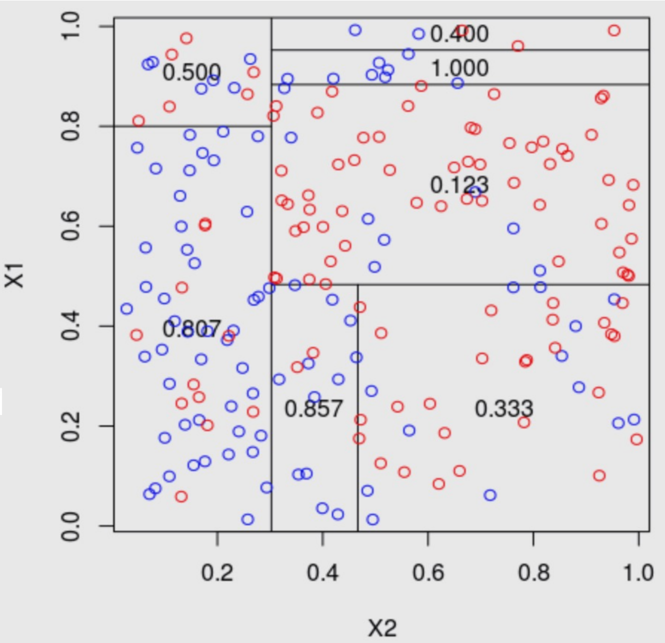
Decision Tree Regressor

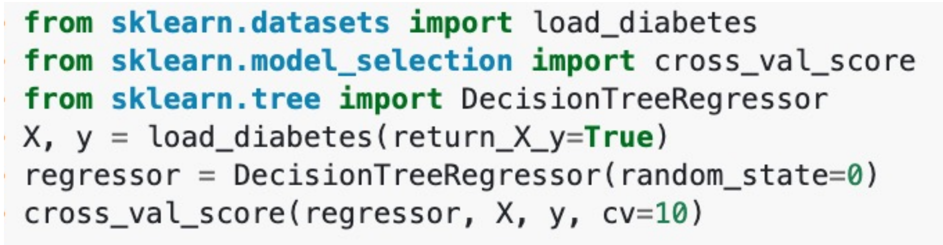
- ❑ Decision tree regression trains a model in the structure of a tree to predict data points like the probable profit from the sale of a product

- ❑ A 2D example of Decision tree regressor can be visualized like the diagram on the right

Decision Tree – Regression Example



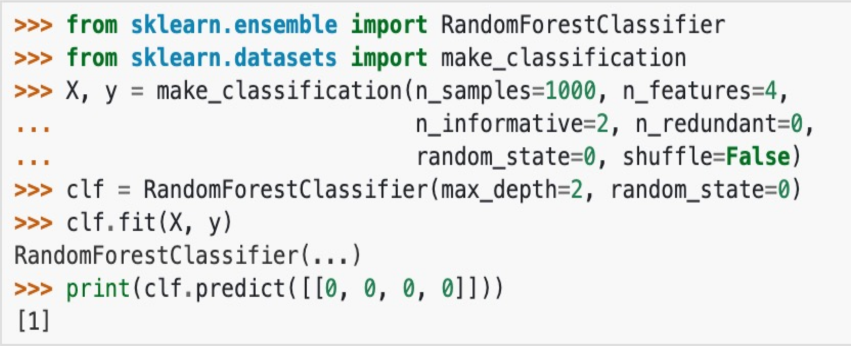




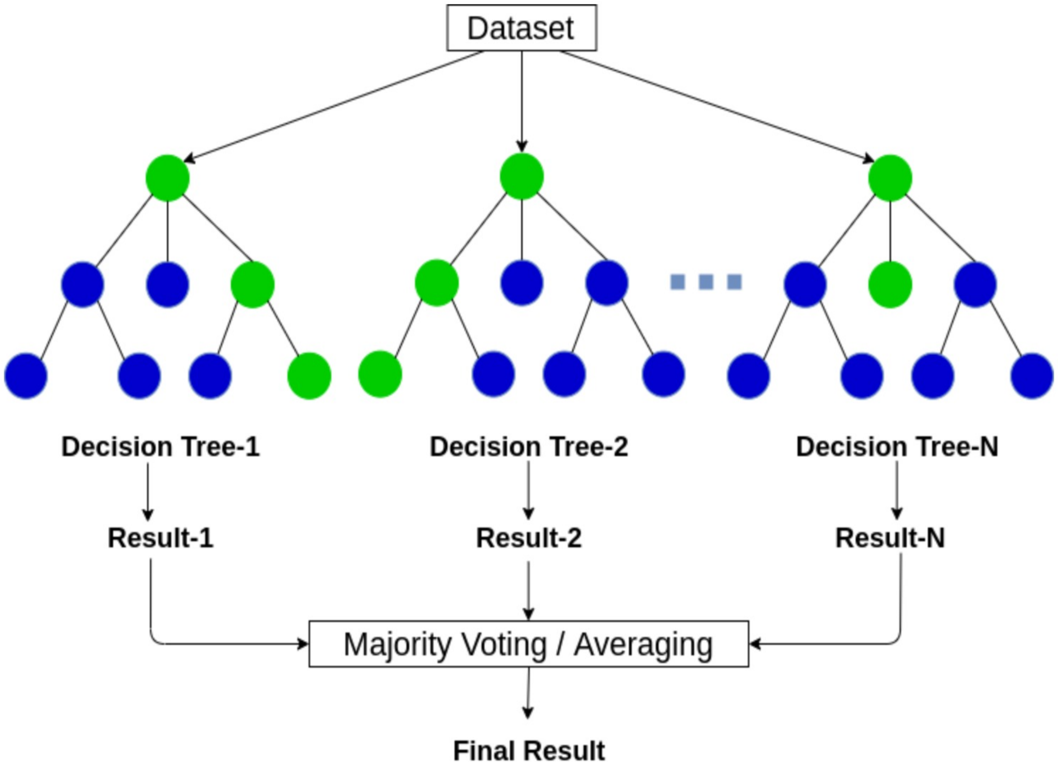
```
from sklearn.datasets import load_diabetes
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeRegressor
X, y = load_diabetes(return_X_y=True)
regressor = DecisionTreeRegressor(random_state=0)
cross_val_score(regressor, X, y, cv=10)
```

Classification - Random Forest

- ❑ A random forest fits a number of decision tree classifiers on various sub-samples of the dataset, and
- ❑ Uses averaging to improve the predictive accuracy and control over-fitting.

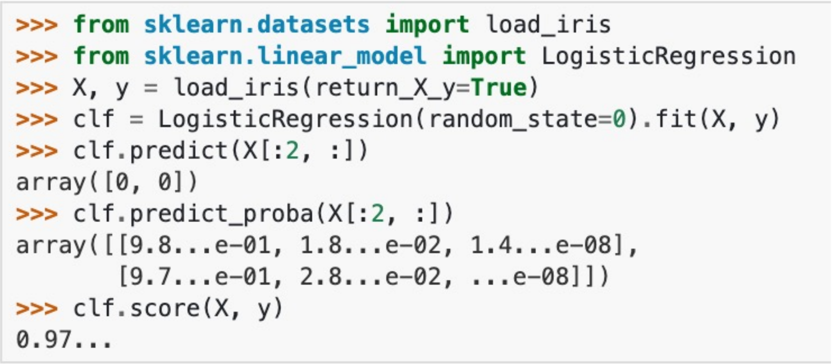


```
>>> from sklearn.ensemble import RandomForestClassifier
>>> from sklearn.datasets import make_classification
>>> X, y = make_classification(n_samples=1000, n_features=4,
...                            n_informative=2, n_redundant=0,
...                            random_state=0, shuffle=False)
>>> clf = RandomForestClassifier(max_depth=2, random_state=0)
>>> clf.fit(X, y)
RandomForestClassifier(...)
>>> print(clf.predict([[0, 0, 0, 0]]))
[1]
```

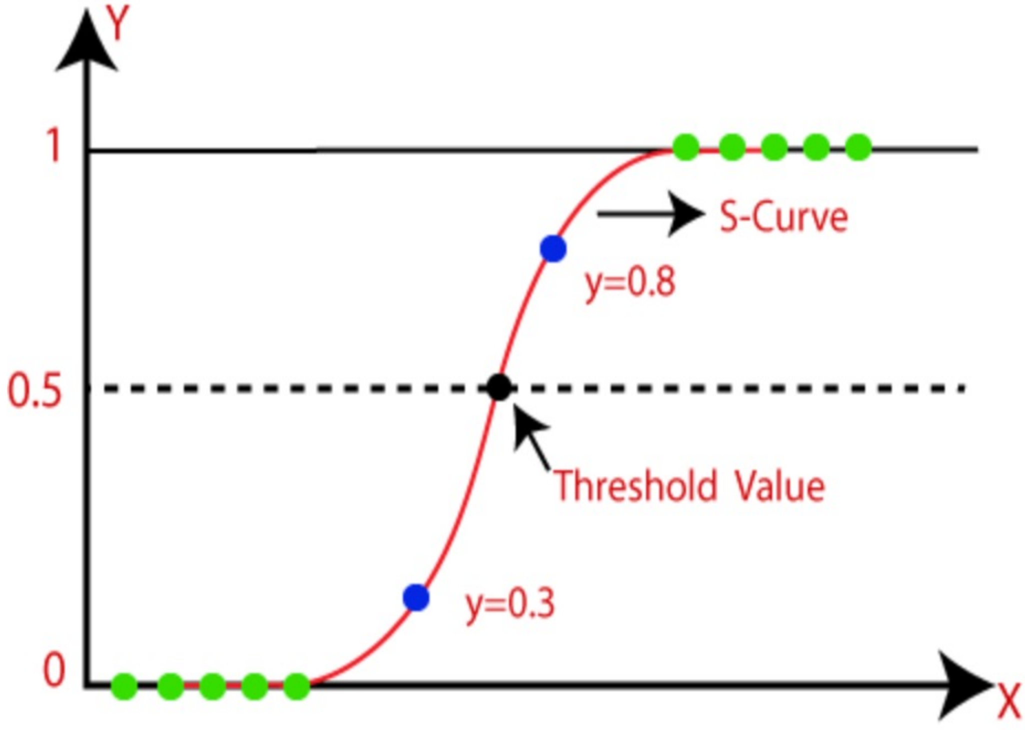


Classification Algorithm- Logistic Regression

- ❑ Logistic Regression – A simple binary classification model
- ❑ Gives the probabilistic values which lie between 0 and 1



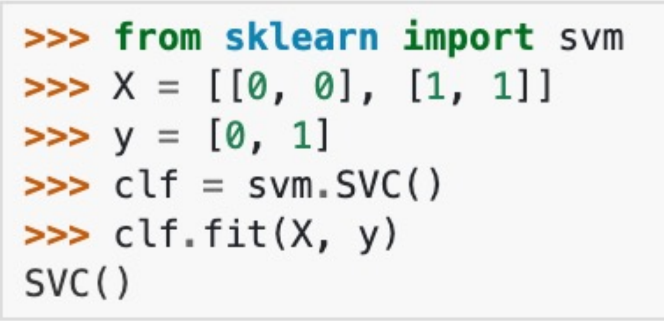
```
>>> from sklearn.datasets import load_iris
>>> from sklearn.linear_model import LogisticRegression
>>> X, y = load_iris(return_X_y=True)
>>> clf = LogisticRegression(random_state=0).fit(X, y)
>>> clf.predict(X[:2, :])
array([0, 0])
>>> clf.predict_proba(X[:2, :])
array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])
>>> clf.score(X, y)
0.97...
```



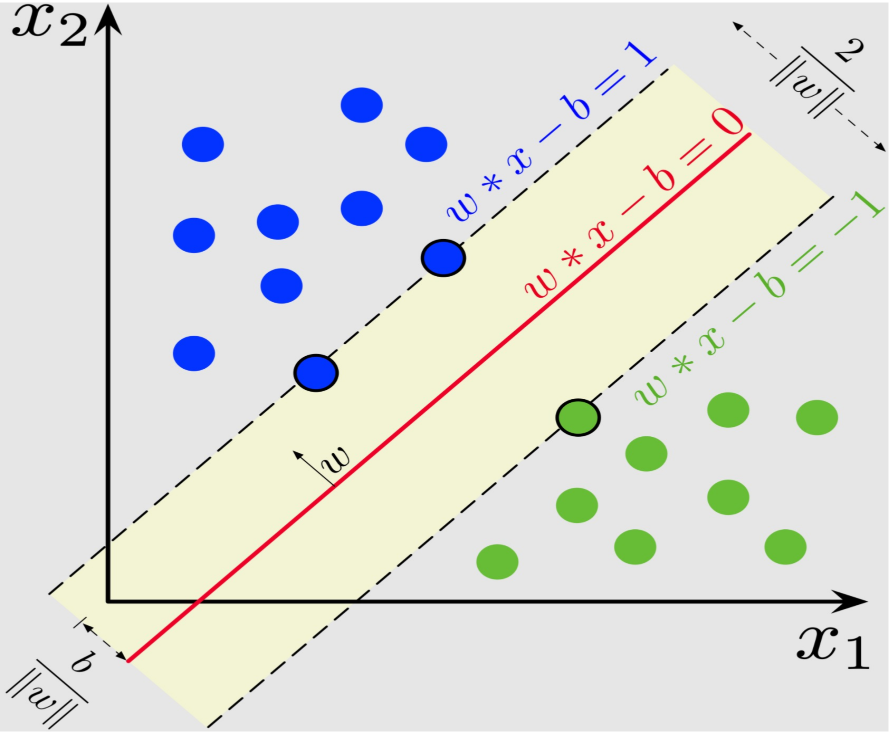
Classification Algorithm - Support Vector Machine

- **Support vector machines (SVMs)** are a set of supervised learning methods used for classification, regression and outliers detection.

- Effective in high dimensional spaces.



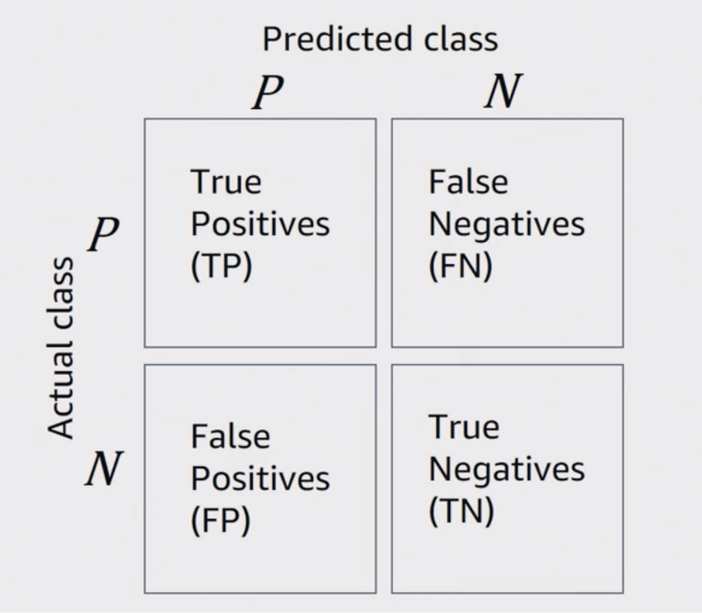
```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC()
```



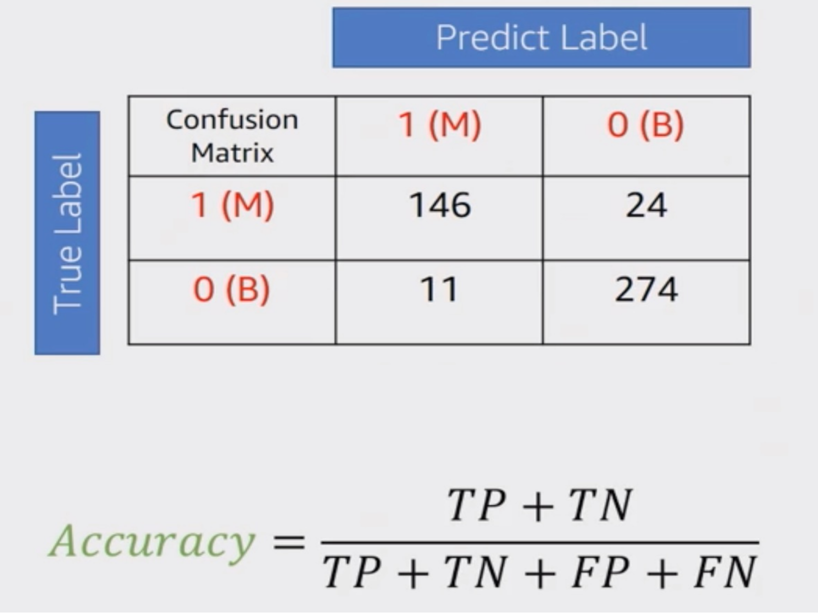
Model Evaluation – Binary Classification

Model Evaluation Metrics - Classification

A confusion matrix is computed to evaluate the accuracy of a classification model.



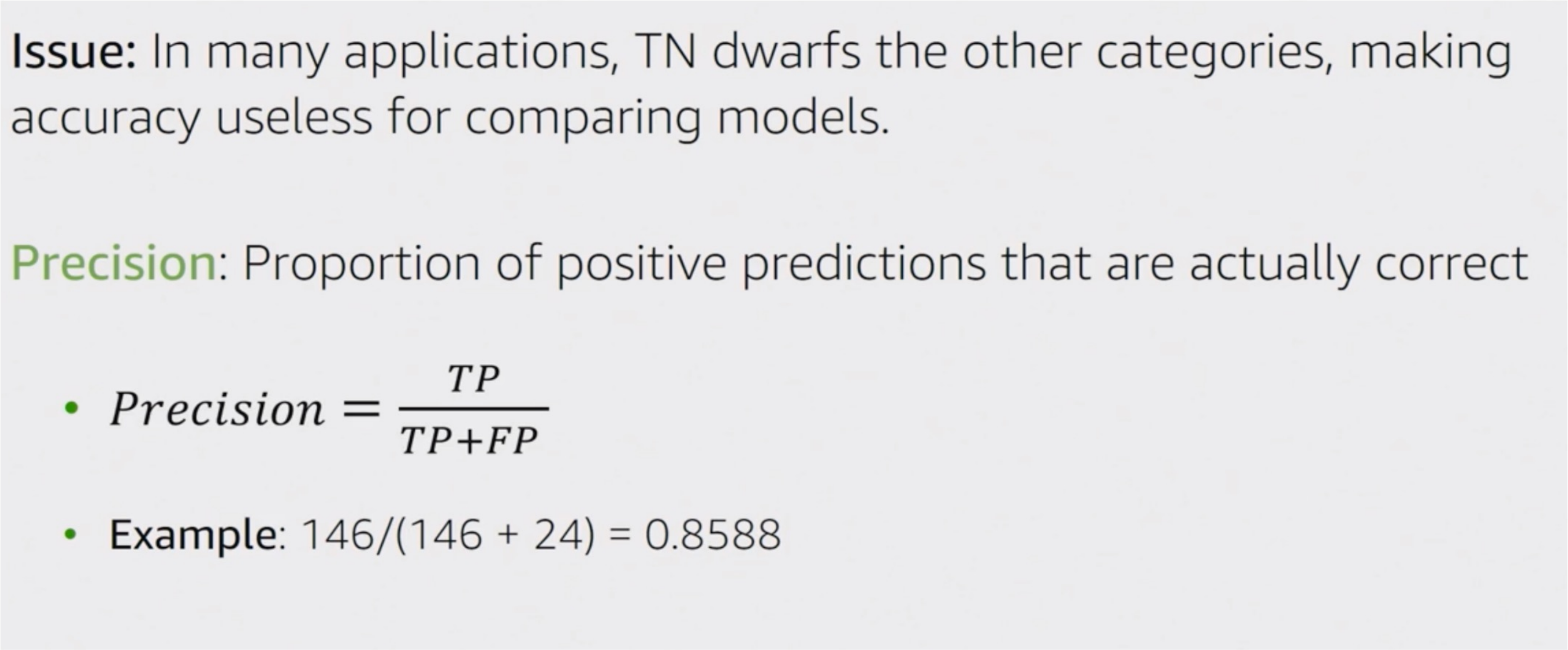
		Predicted class	
		P	N
Actual class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)



		Predict Label	
	Confusion Matrix	1 (M)	0 (B)
True Label	1 (M)	146	24
	0 (B)	11	274

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Model Evaluation - Precision

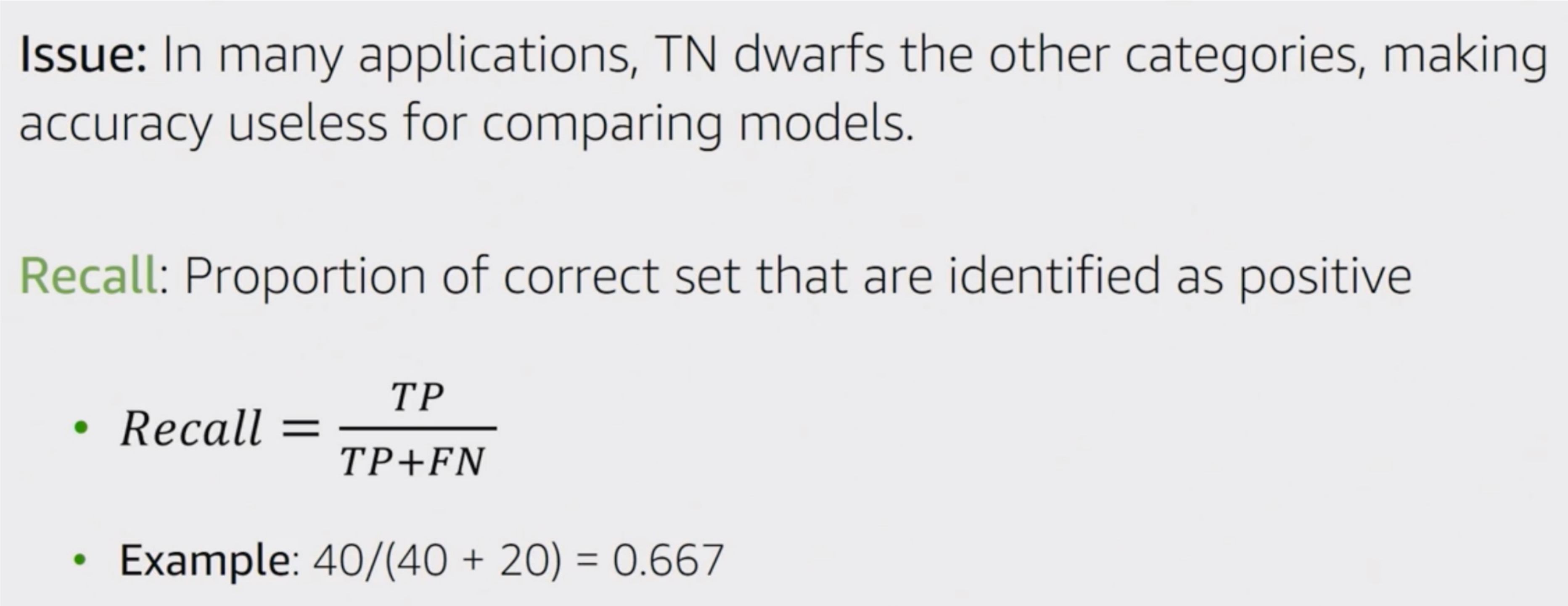


Issue: In many applications, TN dwarfs the other categories, making accuracy useless for comparing models.

Precision: Proportion of positive predictions that are actually correct

- $Precision = \frac{TP}{TP+FP}$
- **Example**: 146/(146 + 24) = 0.8588

Model Evaluation - Recall

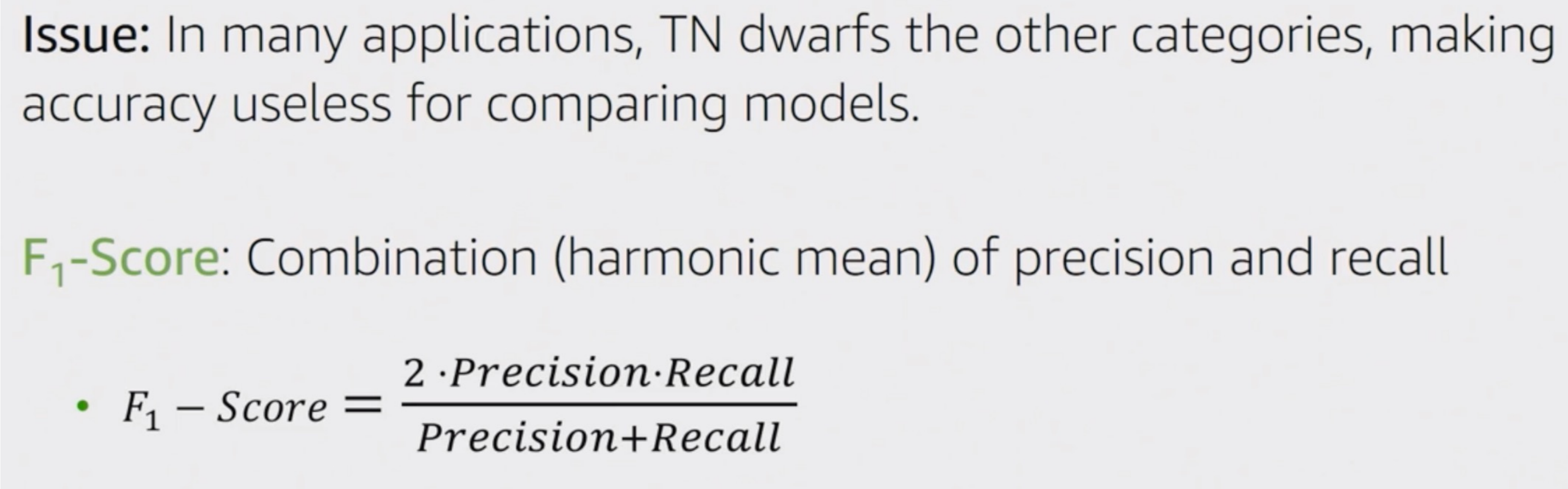


Issue: In many applications, TN dwarfs the other categories, making accuracy useless for comparing models.

Recall: Proportion of correct set that are identified as positive

- $Recall = \frac{TP}{TP+FN}$
- **Example**: 40/(40 + 20) = 0.667

Model Evaluation – F1 Score



Issue: In many applications, TN dwarfs the other categories, making accuracy useless for comparing models.

F_1-Score: Combination (harmonic mean) of precision and recall

- $$F_1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Quiz

A Data Scientist is evaluating different binary classification models. A false positive result is 5 times more expensive (from a business perspective) than a false negative result.
The models should be evaluated based on the following criteria: 1) Must have a recall rate of at least 80% 2) Must have a false positive rate of 10% or less 3) Must minimize business costs After creating each binary classification model, the Data Scientist generates the corresponding confusion matrix. Which confusion matrix represents the model that satisfies the requirements?

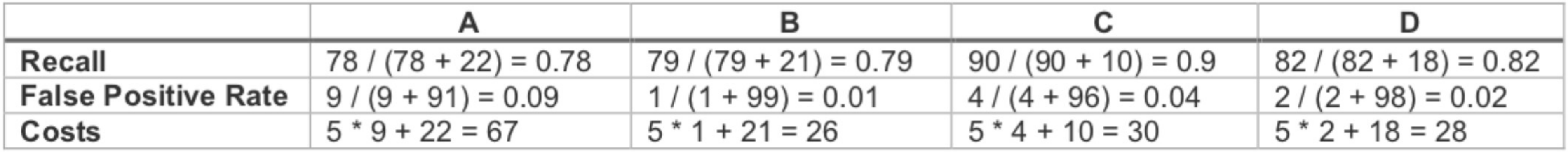
A. TN = 91, FP = 9 FN = 22, TP = 78

B. TN = 99, FP = 1 FN = 21, TP = 79

C. TN = 96, FP = 4 FN = 10, TP = 90

D. TN = 98, FP = 2 FN = 18, TP = 82

Quiz Answer



	A	B	C	D
Recall	78 / (78 + 22) = 0.78	79 / (79 + 21) = 0.79	90 / (90 + 10) = 0.9	82 / (82 + 18) = 0.82
False Positive Rate	9 / (9 + 91) = 0.09	1 / (1 + 99) = 0.01	4 / (4 + 96) = 0.04	2 / (2 + 98) = 0.02
Costs	5 * 9 + 22 = 67	5 * 1 + 21 = 26	5 * 4 + 10 = 30	5 * 2 + 18 = 28

Thank You