UNIVERSITY OF BOLOGNA

BDA Course Project

—

Covertype Classification

ARTIFICIAL INTELLIGENCE

Overview

- Covertype dataset
- Data exploration
- Pre-processing
- Train / Test split and Evaluation
- ML Algorithms: Naive Bayes, Random Forest, Logistic Regression
- Possible improvements

Covertype Dataset

- From UC Irvine Machine Learning Repository
- Forest type on 30x30 m cells in Roosevelt National Forest of Northern Colorado
- Task: predict forest covertype from cartographic variables.
- 581.012 instances and 54 features
- Target class: Cover_Type (integer) – 7 classes

1 - SPRUCE - FIR

2. LODGEPOLE PINE

3 - PONDEROSA PINE

4 - COTTONWOOD / WILLOW

5 - ASPEN

6 - DOUGLAS - FIR

7 - KRUMMHOLZ

Dataset Exploration - pt.1

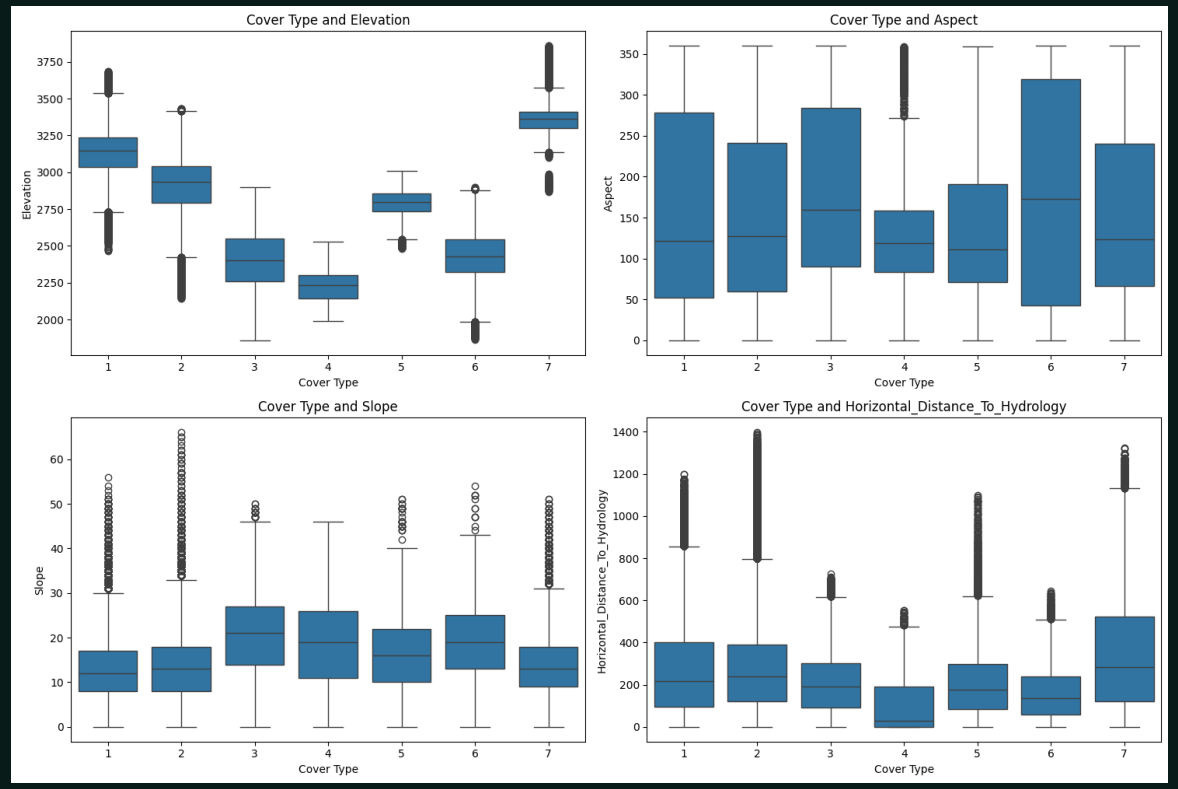
- The dataset does not contain a header
  - Elevation (meters)
  - Aspect (azimuth)
  - Slope (degrees)
  - …
  - Soil Type (binary)
  - Wilderness Area (binary)
- No null entry
- Distribution of target class values `Cover_Type`

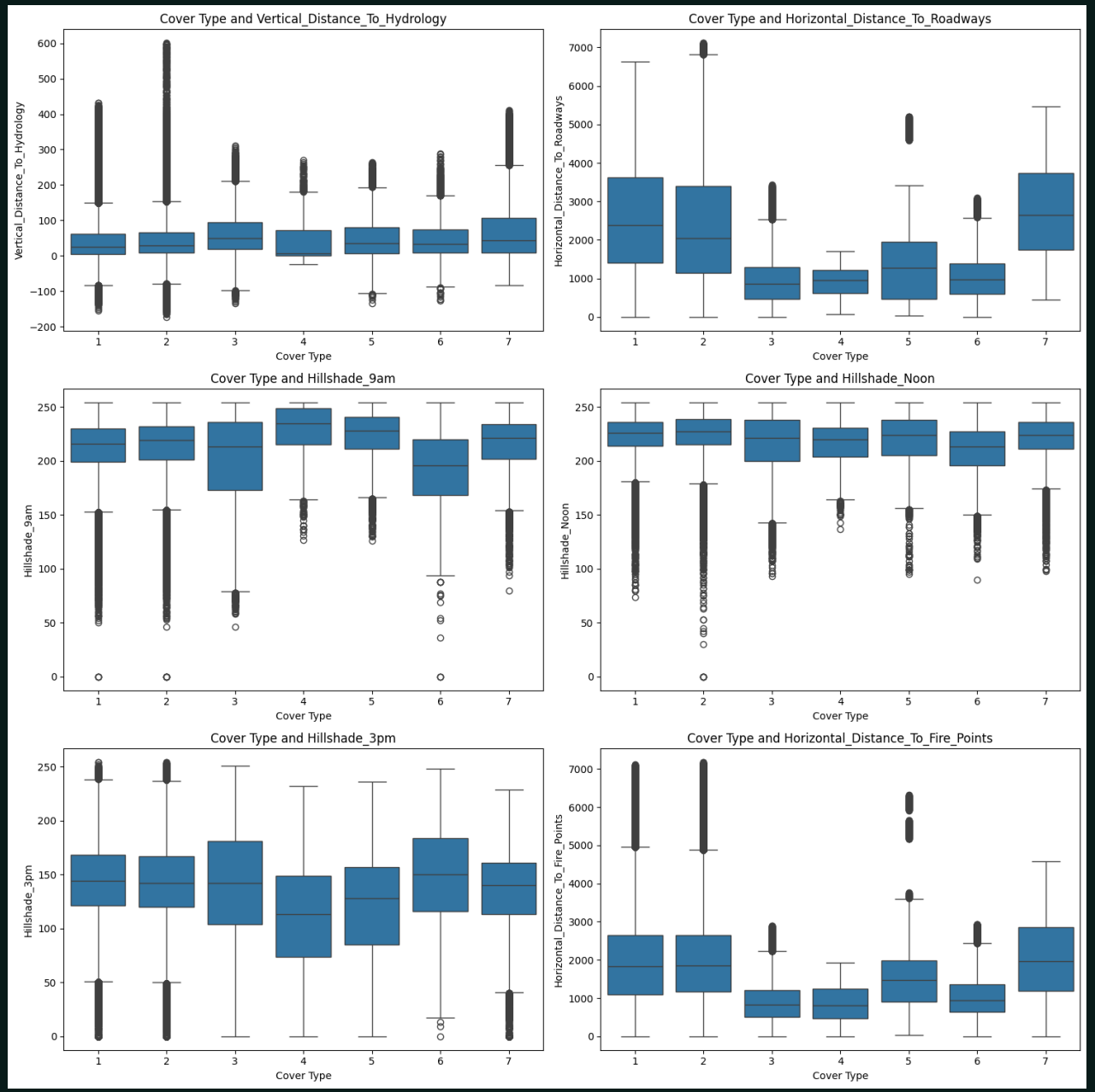
```
schema = StructType([
    StructField("Elevation", IntegerType(), True),
    StructField("Aspect", IntegerType(), True),
    StructField("Slope", IntegerType(), True),
```

```
df.groupBy("Cover_Type").count().orderBy("count").show()
```

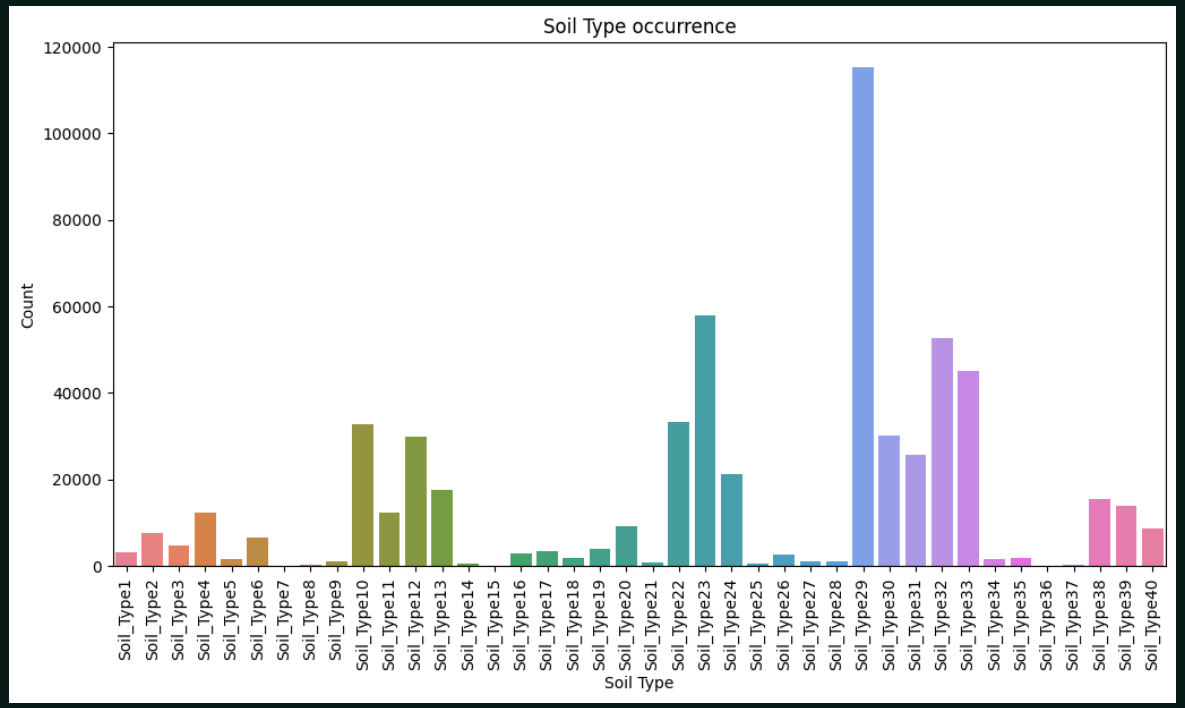
```
+----------+------+
|Cover_Type| count|
+----------+------+
|         4|  2747|
|         5|  9493|
|         6| 17367|
|         7| 20510|
|         3| 35754|
|         1|211840|
|         2|283301|
+----------+------+
```

Exploration - pt.2

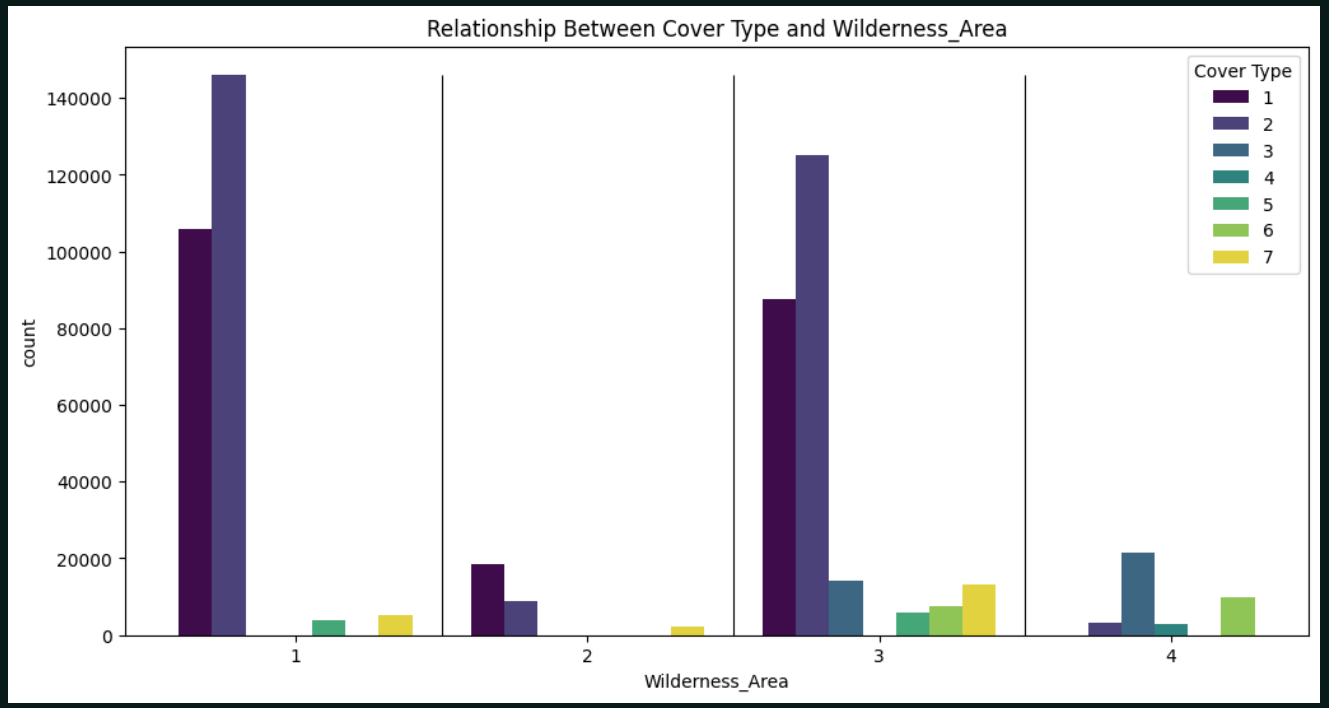




Exploration - pt.3



40 Soil Types



4 Wilderness Areas

Pre-processing

- Re-assign target class values [0,6]
- `VectorAssembler`
- `MinMaxScaler` (for some features)
  - Recompose the df
- Sparse represantation

```
assembler = VectorAssembler(inputCols=input_columns, outputCol="features")
df_assembled = assembler.transform(df)
```

```
minmax_scaler = MinMaxScaler(inputCol="features", outputCol="scaled_features")
minmax_model = minmax_scaler.fit(df_assembled)
```

```
df_mm_final.select("final_features").show(5, truncate=False)
```

```
+--------------------------------------------------------------------------------
|final_features
+--------------------------------------------------------------------------------
|(54,[0,1,2,3,4,5,6,7,8,9,38,50],[0.36868434217108553,0.14166666666666666,0.045454
|(54,[0,1,2,3,4,5,6,7,8,9,38,50],[0.36568284142071034,0.15555555555555556,0.030303
|(54,[0,1,2,3,4,5,6,7,8,9,21,50],[0.47273636818409204,0.3861111111111111,0.136363
|(54,[0,1,2,3,4,5,6,7,8,9,39,50],[0.4632316158079039,0.4305555555555556,0.27272727
|(54,[0,1,2,3,4,5,6,7,8,9,38,50],[0.368184092046023,0.125,0.030303030303030304,0.1
+--------------------------------------------------------------------------------
only showing top 5 rows
```

Train/Test, Evaluation, ML Algorithms

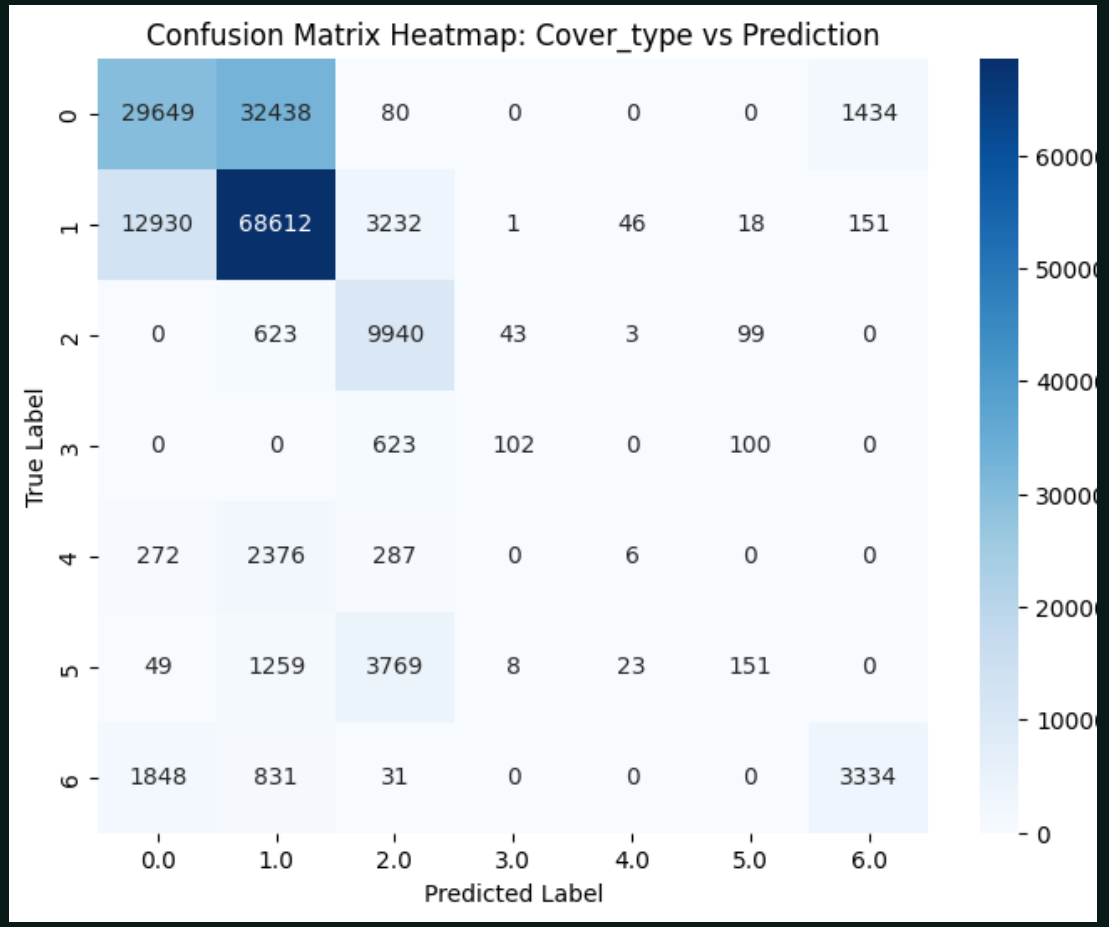
- 70/30 ratio, seed = 42
  - Train split: 406.644 elem
  - Test split: 174.368 elem

```
(trainSplit_mm, testSplit_mm) = df_mm_final.randomSplit([0.7, 0.3], seed=42)
```

- `MulticlassClassificationEvaluator` used for eval
- 3 ML Algorithms:
  - Naive Bayes
  - Random Forest
  - Logistic Regression

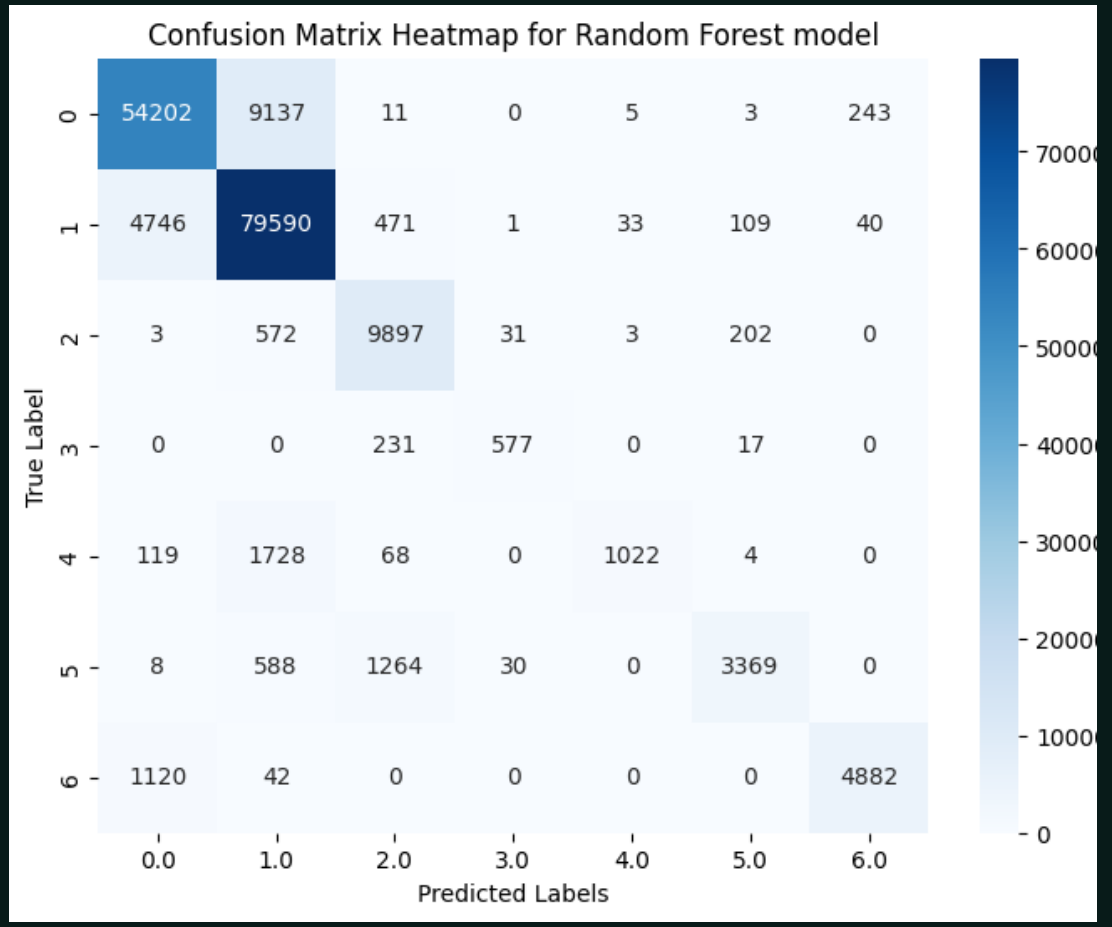
ML Algorithms – Naive Bayes

- Probabilistic classification algorithm
- Based on Bayes' Theorem
- Key assumption: features are *conditionally independent* given the target class
- `modelType=multinomial` – well-suited for multi-class classification problem with categorical outcomes
- `smoothing=1.0` – mitigate issues with assigning zero probabilities
- Accuracy: 64%



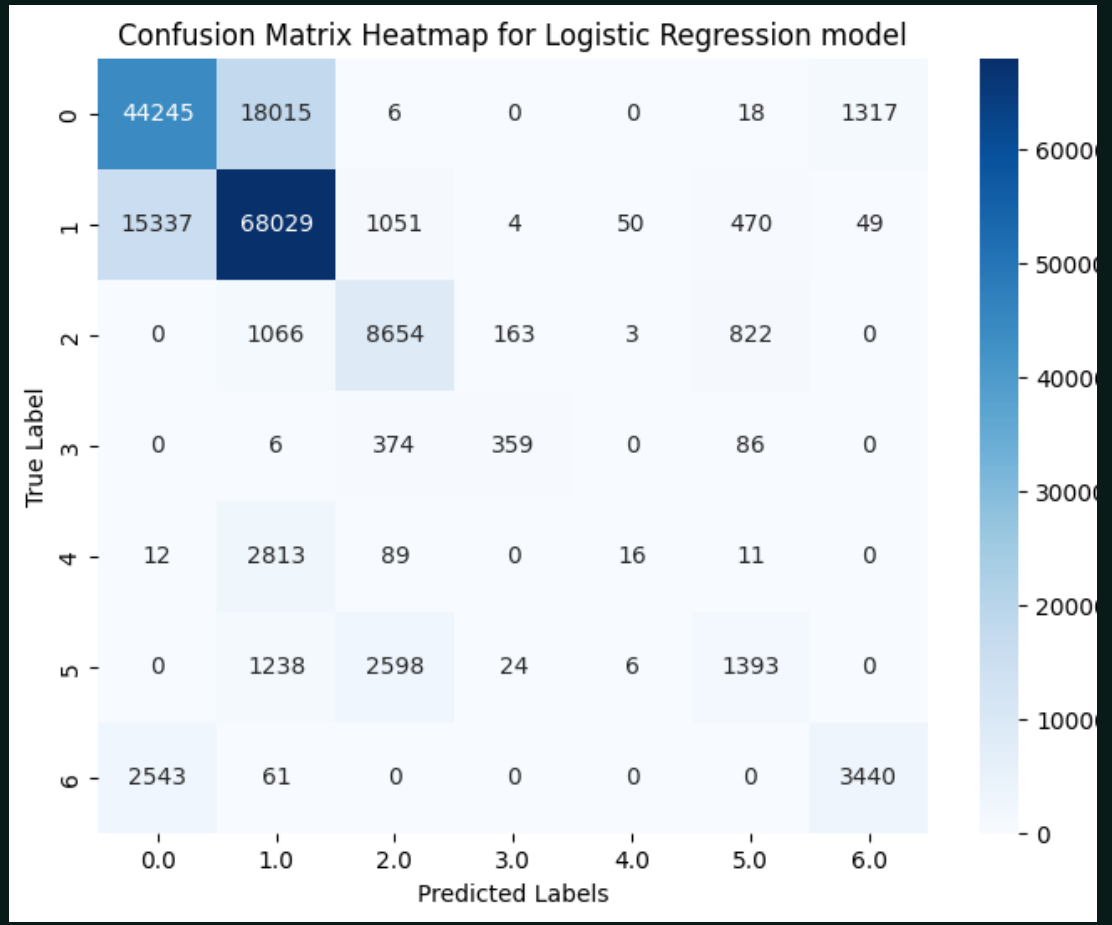
ML Algorithms – Random Forest

- Ensemble learning method
  - combines the predictions of multiple decision trees
  - bagging – subset of data
  - considers random subset of features
- `numTrees=100`
- `maxDepth=30`
- Accuracy: 89%



ML Algorithms – Logistic Regression

- Uses the logistic (sigmoid) function
- Map input features to a probability [0, 1]
- In pyspark extended with One-vs-Rest technique
- `family=multinomial`
- `maxIter=500`
- Accuracy: 72%



Possible improvements

- Applying pre-processing techniques (NB)
  - Oversapling
  - Undersampling
- Standard Scaler, mean=0, variance=1 (RF, LR)
- Feature Selection
- Feature Extraction (with PCA)
- Cross-validation

Thank you

Sara Vorabbi

0001026226

sara.vorabbi@studio.unibo.it