МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

**Обнинский институт атомной энергетики –**

филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»

**(ИАТЭ НИЯУ МИФИ)**

Отделение интеллектуальных кибернетических систем

ЛАБОРАТОРНАЯ РАБОТА №4

«Машинное обучение »

по дисциплине

«Большие данные»

Выполнил студент 1 курса

группы ИВТ-М20

Лискунов Р. Г.

Проверил:

кандидат технических наук

Грицюк С. В.

Обнинск, 2020

# Цель работы

Определить простую задачу машинного обучения и решить ее. Точность на испытательном наборе должна быть не менее 60%.

# Краткая теория

Apache Spark MLlib используется для создания приложения машинного обучения. Приложение выполняет прогнозный анализ на открытом наборе данных. MLlib — это основная библиотека Spark, которая предоставляет множество служебных программ, полезных для задач машинного обучения, таких как:

1. Классификация;
2. Регрессия;
3. Кластеризация;
4. Моделирование сингулярного разложения и анализа по методу главных компонент;
5. Проверки гипотез и статистической выборки.

## Общие сведения о классификации и логистической регрессии

Классификация — это распространенная задача машинного обучения, которая представляет собой процесс сортировки входных данных по категориям. Это задание алгоритма классификации, позволяющее определить, как назначить "метки" входным данным, которые мы предоставляем. Например, можно представить алгоритм машинного обучения, который принимает в качестве входных данных данные о акции. Затем делит биржевую акцию на две категории: акции, которые следует продавать и акции, которые следует хранить.

Логистическая регрессия — один из алгоритмов классификации. API Spark для логистической регрессии подходит для задач двоичной классификации или разделения входных данных на две группы. Дополнительные сведения о логистической регрессии можно посмотреть в документации.

В целом, процесс логистической регрессии создает логистическую функцию. Используем функцию для прогнозирования вероятности того, что входной вектор принадлежит одной группе или другой.

# Ход работы

В процессе работы мы рассмотрим набор данных, состоящий из популярных детских имен. Сперва мы подключим контекст Spark, а также укажем в качестве dataframe, описанный выше набор данных. Первые 20 строк набора показаны на рисунке 1.

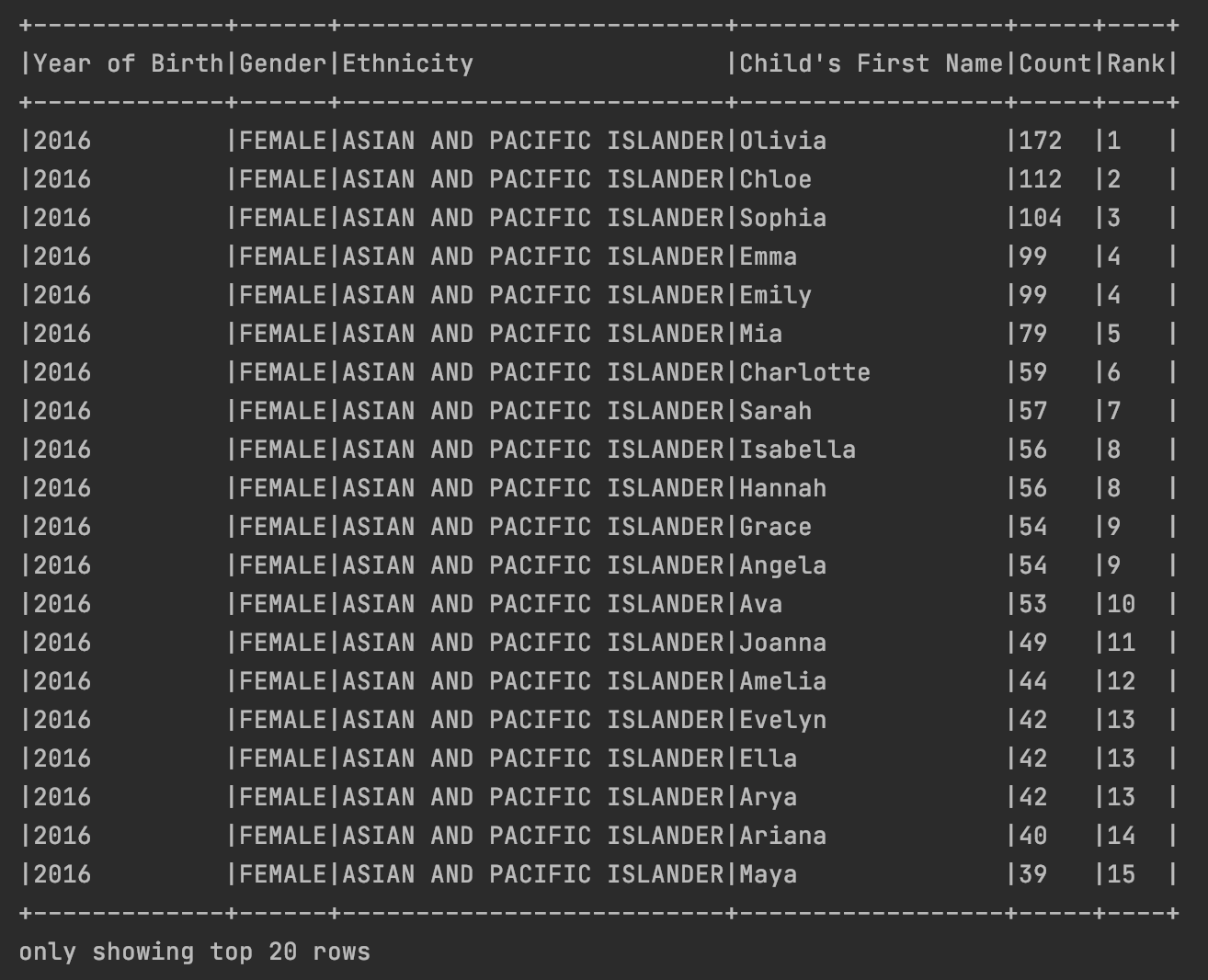


Рисунок 1. Топ 20 строк набора данных

Поставим задачу исследовать взаимосвязь числа имен, их ранга и их классификацию по этническому происхождению. В ходе исследования мы хотим попробовать предсказать, что большая или, наоборот, меньшая часть детей принадлежит конкретно взятому этносу.

Для этого мы создам дополнительные объекты языка программирования – dataframe, которые будут содержать дополнительную для нас информацию, которые касаются выбранных столбцов.

В процессе работы у нас появляются трудности с объединением данных, поэтому мы используем VectorAssembler – это преобразователь, который объединяет заданный список столбцов в один векторный столбец. Это полезно для объединения необработанных функций и функций, созданных различными преобразователями функций, в один вектор функций.

Работать напрямую с данными, хоть и в случае моего небольшого набора данных, не составляет труда, но для удобства обращения воспользуемся StringIndexer, который кодирует строковый столбец меток в столбец индексов меток. Также введем столбец features, который будет агрегировать значения числа детей, их ранга и этноса. Результаты представлены на рисунке 2.

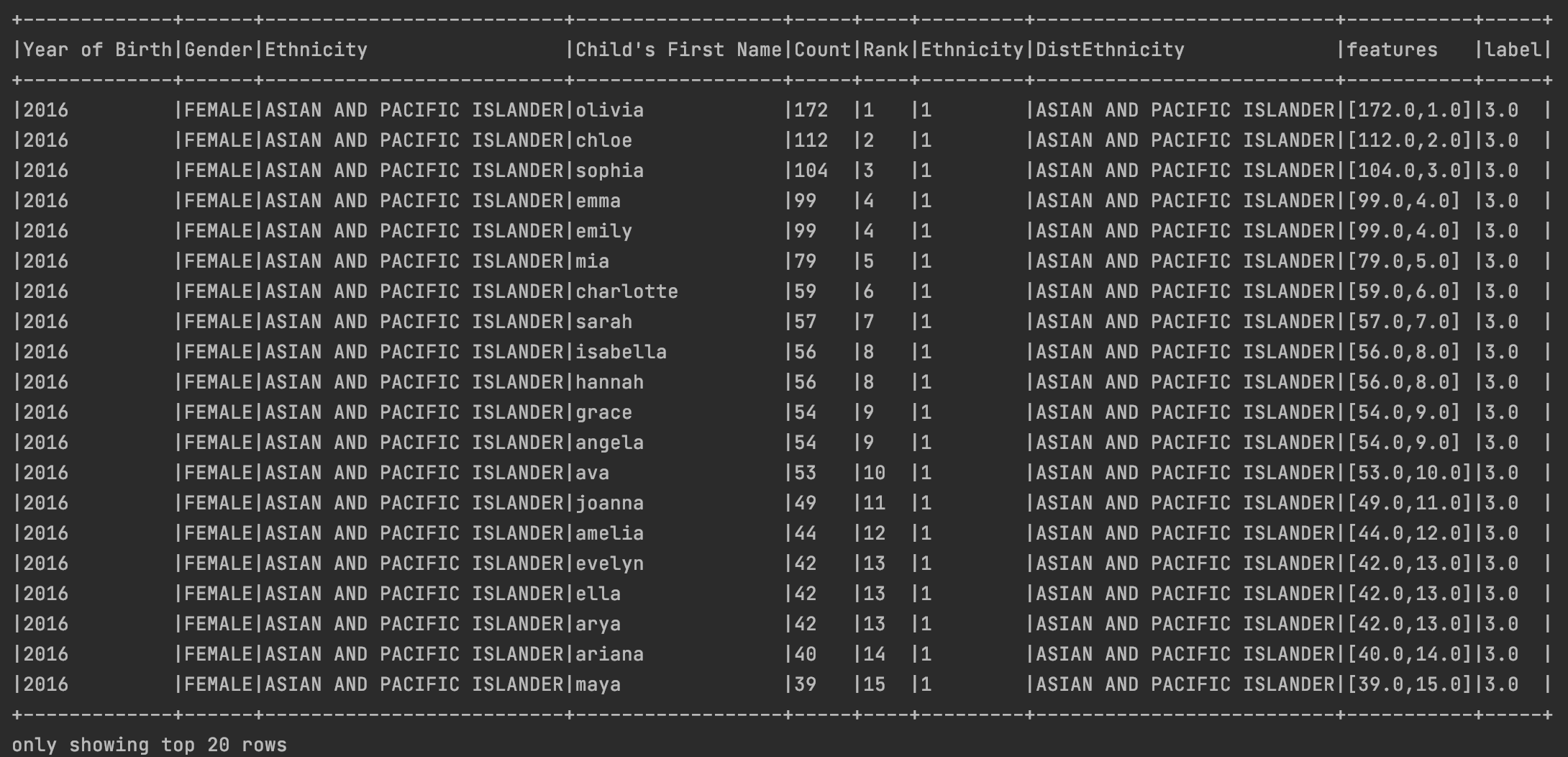


Рисунок . Результаты агрегирования данных

Ввиду наличия большого объема данных, нам представляется возможным разбить их на более мелкие части. Таким образом, мы подготовим данные для логистической регрессии.

Зададим необходимые параметры для обучения и поддержания точности на уровне, указанном в цели работы. Полученные результаты отобразим на рисунке 3.

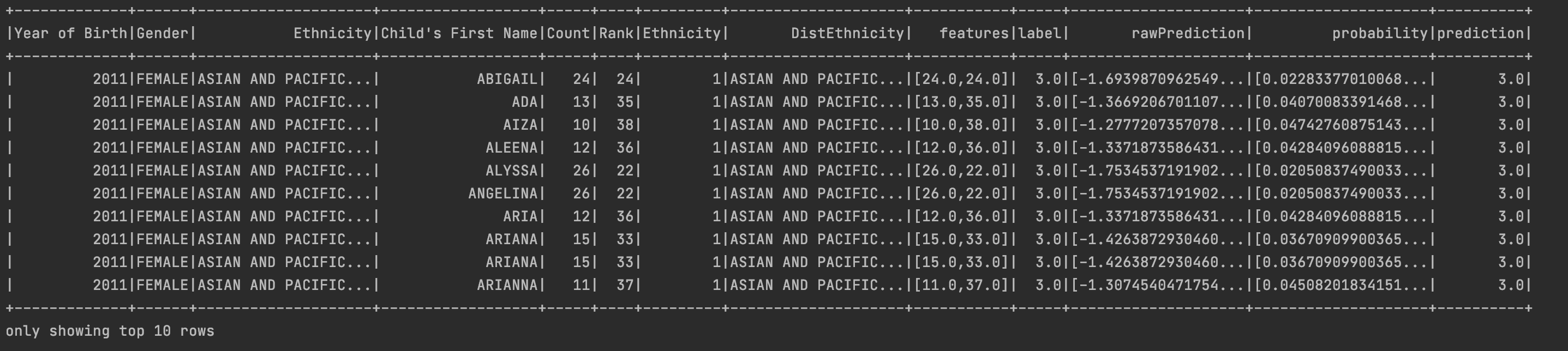


Рисунок 3. Результаты предсказаний

Несмотря на повторяющиеся имена в таблице результатов, мы можем увидеть, что теперь наша модель может предсказать по рангу и числу вероятный этнос ребенка. Крайний правый столбец, если исходить из наших предложений, совпадает с действительным значением, что является верным результатом исследования.

# Листинг кода

package LabFour  
  
import org.apache.log4j.Level.WARN  
import org.apache.log4j.LogManager  
import org.apache.spark.ml.classification.{LogisticRegression, LogisticRegressionModel}  
import org.apache.spark.ml.feature.{StringIndexer, VectorAssembler}  
import org.apache.spark.sql.functions.{col, lower}  
import org.apache.spark.sql.{DataFrame, SparkSession}  
  
object LabFour {  
 val PATH: String = "src/main/data"  
 val NODES: Int = 3  
  
 def main(args: Array[String]): Unit = {  
 val spark: SparkSession = SparkSession  
 .builder()  
 .appName("Lab4")  
 .master(s"local[$NODES]")  
 .getOrCreate  
 LogManager.getRootLogger.setLevel(WARN)  
  
 val dataframe: DataFrame = spark  
 .read  
 .format("csv")  
 .option("header", "true")  
 .option("delimiter", ",")  
 .option("inferSchema", value = true)  
 .load(s"$PATH/var.csv")  
  
 dataframe.show(false)  
   
 val seq: Seq[(Int, String)] = Seq(  
 (0, "WHITE NON HISPANIC"),  
 (1, "ASIAN AND PACIFIC ISLANDER"),  
 (2, "HISPANIC"),  
 (3, "BLACK NON HISPANIC")  
 )  
  
 import spark.implicits.\_  
 val mapper: DataFrame = seq.toDF("Ethnicity", "DistEthnicity")  
  
 val mapped: DataFrame = dataframe  
 .join(  
 mapper, dataframe("Ethnicity") === mapper("DistEthnicity"),  
 "inner"  
 )  
 val columns: Array[String] = Array("Count", "Rank")  
  
 val assembler: VectorAssembler = new VectorAssembler()  
 .setInputCols(columns)  
 .setOutputCol("features")  
 val feature: DataFrame = assembler.transform(mapped)  
  
 val indexer: StringIndexer = new StringIndexer()  
 .setInputCol("DistEthnicity")  
 .setOutputCol("label")  
  
 val label: DataFrame = indexer  
 .fit(feature)  
 .transform(feature)  
 label.show(false)  
  
 val seed: Int = 5043  
 val Array(training, test) = label  
 .randomSplit(Array(0.7, 0.3), seed)  
  
 val regression: LogisticRegression = new LogisticRegression()  
 .setMaxIter(100)  
 .setRegParam(0.02)  
 .setElasticNetParam(0.8)  
 val model: LogisticRegressionModel = regression  
 .fit(training)  
  
 val prediction: DataFrame = model  
 .transform(test)  
 prediction.show(10)  
 }  
}

# Вывод

В ходе данной лабораторной работы я изучил особенности и возможности способов работы со Spark ML и предсказания данных на основе логистической регрессии. Для этого я использовал набор данных популярных детских имен. Полученная модель умеет по числу детей, которые родились в определенный год и по рангу имени, определять этнос ребенка. Данное исследование позволяет выявить наибольшее или наименьшее число детей, которые относятся к выбранному этносу. Сами значения могут быть использованы, например, при изготовке вакцин. Так, при росте числа новорожденных в Нью-Йорке следует изготовлять вакцину с большей вероятностью для определённой этнической группы.