**Classifying whether people donate to a charity**

A PROJECT REPORT

*Submitted by*

**NAME REGISTRATION NUMBER**

K.VISHNU SAINADH BL.EN.U4AIE19028

K.SATWIK BL.EN.U4AIE19034

V.ASHRITH BL.EN.U4AIE19066

***For the course***

***19AIE214-Big Data Analytics***

***Guided and Evaluated by***

***Mr.Saravanan S***

***Asst. Prof (SG),***

***Dept. of CSE.***



**AMRITA SCHOOL OF ENGINEERING, BANGALORE**

**AMRITA VISHWA VIDHYAPEETHAM**

**BANGALORE-560 035**

Title of the project

**Classifying whether people donate to a charity**

**Acknowledgement**

The satisfaction that accompanies successful completion of any task would be incomplete without mention of people whom made it possible, and whose constant encouragement and guidance have been source of inspiration throughout the course of this project work.

It is a great pleasure to express our gratitude and indebtedness to our project guide **Mr.Saravanan S**, Asst. Professor ,Dept. of CSE, Amrita School of Engineering, Bangalore for his valuable guidance, encouragement, moral support and affection throughout the project work.

#### TABLE OF CONTENTS:

1. ABSTRACT 5
2. APACHE SPARK 6
3. [MACHINE LEARNING](https://docs.google.com/document/d/1-pS-wcQTZBrotr-R17Hb54lvfDzAH2US/edit#heading=h.gjdgxs) 11
   1. SUPERVISED ML METHODS 12
      * Decision Tree 15
      * Logistic Regression 18
4. [MODEL IMPLEMENTATION](https://docs.google.com/document/d/1-pS-wcQTZBrotr-R17Hb54lvfDzAH2US/edit#heading=h.1fob9te) 22
5. CONCLUSION 29
6. FUTURE WORK 25

TABLE OF FIGURES:

## Fig 1: Pyspark

## Fig 2: Spark Ecosystem

1. Fig 3: Pipelining
2. Fig 4: Methods in Machine Learning
3. Fig 5: Working of Supervised Machine Learning
4. Fig 6: Different problems in Supervised Learning
5. Fig 7: Sample of Decision Tree
6. Fig 8: Sigmoid Function
7. Fig 9: All code Snapshots

**ABSTRACT**

We will employ a variety of supervised algorithms in this project to precisely classify whether individuals donate to a charity. With this approach, we hope to create a model that can properly predict whether a person earns more than $50,000. This type of duty can develop in a non-profit organisation that relies on donations to stay in business. Understanding an individual's income can assist a non-profit in determining how large of a gift to request, as well as whether or not to contact out in the first place. According to our past research, those who earn more than $50,000 are the most likely to donate money to a charity. This project's dataset comes from the UCI Machine Learning Repository.

**APACHE SPARK:**

Apache Spark is a lightning-fast unified analytics engine for big data and machine learning. It was originally developed at UC Berkeley in 2009. Spark has easy-to-use APIs for operating on large datasets. This includes a collection of over 100 operators for transforming data and familiar data frame APIs for manipulating semi-structured data.It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including [Spark SQL](https://spark.apache.org/docs/latest/sql-programming-guide.html) for SQL and structured data processing, [MLlib](https://spark.apache.org/docs/latest/ml-guide.html) for machine learning, [GraphX](https://spark.apache.org/docs/latest/graphx-programming-guide.html) for graph processing, and [Structured Streaming](https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html) for incremental computation and stream processing. These standard libraries increase developer productivity and can be seamlessly combined to create complex workflows. Engineered from the bottom-up for performance, Spark can be [100x faster than Hadoop for large scale data processing](https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html) by exploiting in memory computing and other optimizations. Spark is also fast when data is stored on disk, and currently holds the world record for large-scale on-disk sorting. It scales by distributing processing work across large clusters of computers, with built-in parallelism and fault tolerance.

Since its release, [Apache Spark](http://spark.apache.org/), the unified analytics engine, has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Netflix, Yahoo, and eBay have deployed Spark at massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations.

**PYSPARK:**

PySpark is an interface for Apache Spark in Python. PySpark has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for Spark. In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programming language. This has been achieved by taking advantage of the Py4j library. Py4J is a popular library which is integrated within PySpark and allows python to dynamically interface with JVM objects. PySpark features quite a few libraries for writing efficient programs. Furthermore, there are various external libraries that are also compatible. It not only allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analyzing your data in a distributed environment. PySpark supports most of Spark’s features such as Spark SQL, DataFrame, Streaming, MLlib (Machine Learning) and Spark Core.



## Fig 1: Pyspark

Spark also handles two other data types: DataFrames and Datasets, in addition to RDDs. DataFrames provide a table of data with rows and columns and are the most prevalent structured application programming interfaces (APIs). RDD, although being a crucial part of Spark, is presently in maintenance mode. DataFrames has taken over as the primary API for Spark's Machine Learning Library (MLlib) as a result of its popularity. This is critical to remember while using the MLlib API, because DataFrames ensure consistency between languages like Scala, Java, Python, and R. Like a database table, Data Frame has a row and column structure. RDDs are created from Data Frames. RDDs are immutable; once loaded, they cannot be changed. You can, however, carry out Transformations and Actions. RDDs have a similar history to Spark Data Frames. Spark Data Frames, like RDDs, are immutable. On Data Frames, you can do transformations and operations.

Datasets are a type-safe, object-oriented programming interface that is an extension of DataFrames. Unlike DataFrames, Datasets are by default a collection of tightly typed JVM objects.

Data from DataFrames and SQL data stores, such as Apache Hive, may be queried with Spark SQL. When performed in another language, Spark SQL queries produce a DataFrame or Dataset.

RDDs **(Resilient Distributed Datasets)** serve as the foundation for all Spark applications.

* Resilient: It is fault tolerant and capable of rebuilding data in the event of a failure.
* Distributed: Data in a cluster is distributed across numerous nodes.
* Dataset: A collection of data that has been partitioned and given values.

It's a data abstraction layer that sits on top of the distributed collection. It is immutable and follows sluggish transformations in nature.

You can execute two types of operations with RDDs:

1. Transformations: To generate a new RDD, these procedures are used.

2. Actions: These operations are used on an RDD to tell Apache Spark to perform calculations and provide the results to the driver.

## Apache Spark Ecosystem:

### Structured Data: Spark SQL

### Many data scientists, analysts, and general business intelligence users rely on interactive SQL queries for exploring data. Spark SQL is a Spark module for structured data processing. It provides a programming abstraction called DataFrames and can also act as distributed SQL query engine. It enables unmodified Hadoop Hive queries to run up to 100x faster on existing deployments and data. It also provides powerful integration with the rest of the Spark ecosystem (e.g., integrating SQL query processing with machine learning).

### Streaming Analytics: Spark Streaming

### Many applications need the ability to process and analyze not only batch data, but also streams of new data in real-time. Running on top of Spark, Spark Streaming enables powerful interactive and analytical applications across both streaming and historical data, while inheriting Spark’s ease of use and fault tolerance characteristics. It readily integrates with a wide variety of popular data sources, including HDFS, Flume, Kafka, and Twitter.

### Machine Learning: MLlib

### Machine learning has quickly emerged as a critical piece in mining Big Data for actionable insights. Built on top of Spark, MLlib is a scalable machine learning library that delivers both high-quality algorithms (e.g., multiple iterations to increase accuracy) and blazing speed (up to 100x faster than MapReduce). The library is usable in Java, Scala, and Python as part of Spark applications, so that you can include it in complete workflows.

### Graph Computation: GraphX

### GraphX is a graph computation engine built on top of Spark that enables users to interactively build, transform and reason about graph structured data at scale. It comes complete with a library of common algorithms.

### General Execution: Spark Core

Spark Core is the underlying general execution engine for the Spark platform that all other functionality is built on top of. It provides in-memory computing capabilities to deliver speed, a generalized execution model to support a wide variety of applications, and Java, Scala, and Python APIs for ease of development. The Spark Core and cluster manager distribute data across the Spark cluster and abstract it. This distribution and abstraction make handling Big Data very fast and user-friendly.

### 

## Fig 2: Spark Ecosystem

**PIPELINING:**

A Pipeline chains multiple Transformers and Estimators together to specify an ML workflow. A Pipeline is specified as a sequence of stages, and each stage is either a Transformer or an Estimator. These stages are run in order, and the input DataFrame is transformed as it passes through each stage. For Transformer stages, the transform() method is called on the DataFrame. For Estimator stages, the fit() method is called to produce a Transformer (which becomes part of the PipelineModel, or fitted Pipeline),and that Transformer’s transform() method is called on the DataFrame.

Pipeline components:

Transformers:

A Transformer is an abstraction that includes feature transformers and learned models. Technically, a Transformer implements a method transform(), which converts one DataFrame into another, generally by appending one or more columns. For example:

* A feature transformer might take a DataFrame, read a column (e.g., text), map it into a new column (e.g., feature vectors), and output a new DataFrame with the mapped column appended.
* In Spark ML package, [OneHotEncoder](https://spark.apache.org/docs/latest/ml-features.html#onehotencoder) transforms a column with a label index into a column of vectored features

### Estimators:

An Estimator abstracts the concept of a learning algorithm or any algorithm that fits or trains on data. Technically, an Estimator implements a method fit(), which accepts a DataFrame and produces a Model, which is a Transformer. For example, a learning algorithm such as LogisticRegression is an Estimator, and calling fit() trains a LogisticRegressionModel, which is a Model and hence a Transformer. Another example is K Means as an estimator accepts a training DataFrame and produces a K Means model which is a transformer.

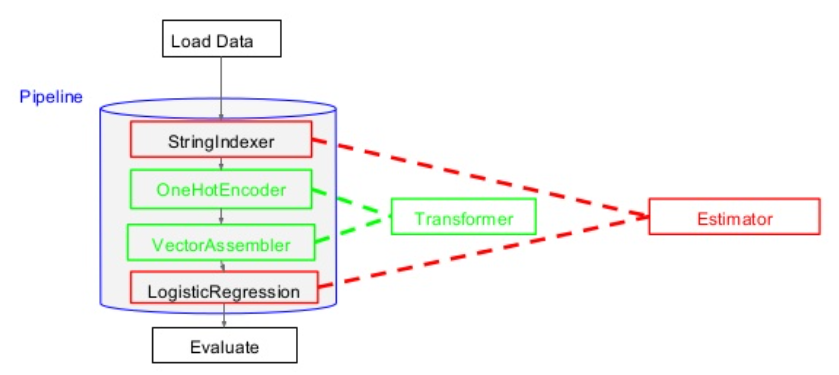


Fig 3: Pipelining

* StringIndexer encodes a string column of labels to a column of label indices.
* VectorAssembler is a transformer that combines a given list of columns into a single vector column. It is useful for combining raw features and features generated by different feature transformers into a single feature vector, in order to train ML models like logistic regression and decision trees
* One-hot encoding  maps a column of label indices to a column of binary vectors, with at most a single one-value. This encoding allows algorithms which expect continuous features, such as Logistic Regression, to use categorical features

**MACHINE LEARNING:**

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed.

Two of the most widely adopted machine learning methods are **supervised learning** which trains algorithms based on example input and output data that is labeled by humans, and **unsupervised learning** which provides the algorithm with no labeled data in order to allow it to find structure within its input data.

Machine learning is a crucial part of the rapidly expanding discipline of data science. Algorithms are trained to generate classifications or predictions using statistical approaches, revealing crucial insights in data mining initiatives. Following that, these insights drive decision-making within applications and enterprises, with the goal of influencing important growth KPIs. As big data expands and grows, the demand for data scientists will rise, necessitating their assistance in identifying the most relevant business questions and, as a result, the data needed to answer them.

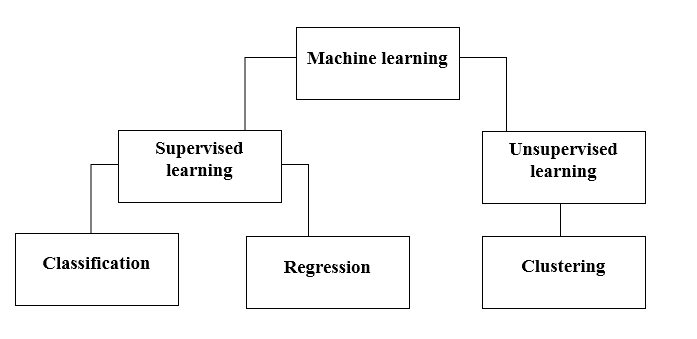


Fig 4: Methods in Machine Learning

### Supervised Learning

Supervised learning is a sort of machine learning in which machines are taught with well-labeled training data and then predict output using that data. The labelled data indicates that some of the input data has already been tagged with the appropriate output.

In supervised learning, the training data presented to the machines acts as a supervisor, instructing the machines on how to correctly predict the output. It uses the same notion as when a student learns under the guidance of a teacher.

The process of supplying input data as well as proper output data to the machine learning model is known as supervised learning.The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

## How Supervised Learning Works?

Models are trained using a labelled dataset in supervised learning, where the model learns about each category of input. The model is tested using test data (a subset of the training set) when the training phase is completed, and it then predicts the output.

The working of Supervised learning can be easily understood by the below example and diagram:

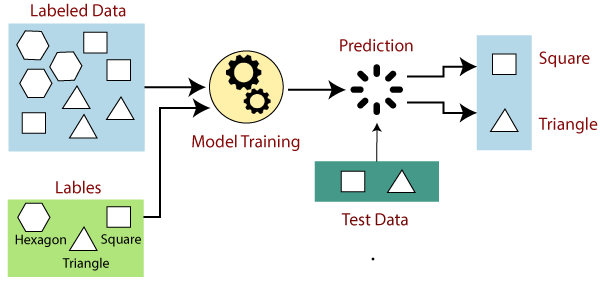


Fig 5: Working of Supervised Machine Learning

Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

* If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
* If the given shape has three sides, then it will be labelled as a triangle.
* If the given shape has six equal sides then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

## Steps Involved in Supervised Learning:

* First Determine the type of training dataset
* Collect/Gather the labelled training data.
* Split the training dataset into training dataset, test dataset, and validation dataset.
* Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
* Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.
* Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
* Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

## Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:

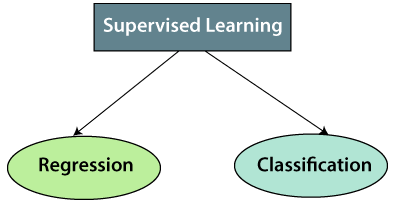


Fig 6: Different problems in Supervised Learning

1. Regression

If there is a relationship between the input and output variables, regression procedures are applied. It's used to predict continuous variables like weather forecasting, market trends, and so on. Some popular supervised learning regression algorithms are listed below:

* Linear Regression
* Regression Trees
* Non-Linear Regression
* Bayesian Linear Regression
* Polynomial Regression

2. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

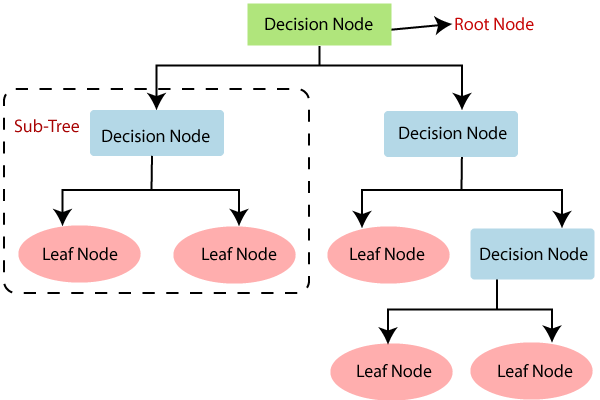
Spam Filtering,

* Random Forest
* Decision Trees
* Logistic Regression
* Support vector Machines

We have used Decision Trees and Logistic Regression in our project.

Decision Trees in Machine Learning :

* Decision Tree is a supervised learning technique that may be used to solve both classification and regression problems, however it is most commonly employed to solve classification issues. Internal nodes represent dataset attributes, branches represent decision rules, and each leaf node provides the conclusion in this tree-structured classifier.
* The Decision Node and the Leaf Node are the two nodes of a Decision tree. Leaf nodes are the output of those decisions and do not contain any more branches, whereas Decision nodes are used to make any decision and have several branches.
* The decisions or the test are performed on the basis of features of the given dataset.
* It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
* It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
* In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
* A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub-trees.



## Fig 7: Sample of Decision Tree

## Decision Tree Terminologies :

* **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
* **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
* **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
* **Branch/Sub Tree:** A tree formed by splitting the tree.
* **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
* **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Logistic Regression in Machine Learning :

* The Supervised Learning methodology includes one of the most prominent Machine Learning algorithms: logistic regression. It's a method for predicting a categorical dependent variable from a set of independent variables.
* A categorical dependent variable's output is predicted using logistic regression. As a result, the result must be a discrete or categorical value. It can be Yes or No, 0 or 1, true or false, and so on, but instead of giving exact values like 0 and 1, it delivers probabilistic values that are somewhere between 0 and 1.
* The only difference between Logistic Regression and Linear Regression is how they are employed. For regression problems, Linear Regression is employed, while for classification difficulties, Logistic Regression is employed.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The logistic function's curve reflects the probability of things like whether the cells are cancerous or not, whether a mouse is obese or not based on its weight, and so on.
* Because it can generate probabilities and classify new data using both continuous and discrete datasets, Logistic Regression is a key machine learning approach.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

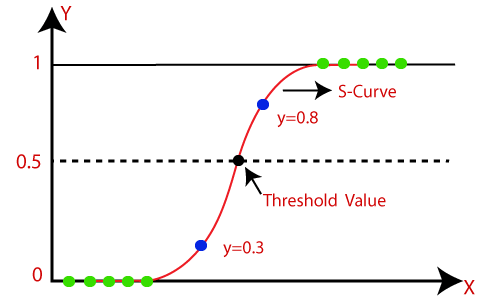


Fig 8: Sigmoid Function

## Logistic Function (Sigmoid Function):

* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1.
* The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

**Data-Set :**

Our dataset has 34587 instances and 14 features

**Features**:

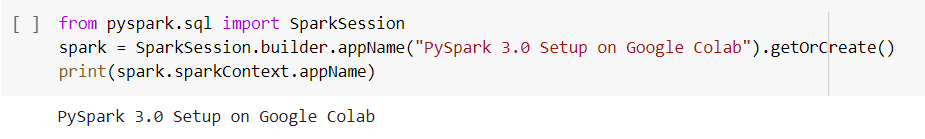
* age: Age
* workclass: Working Class (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
* education\_level: Level of Education (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)
* education-num: Number of educational years completed
* marital-status: Marital status (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
* occupation: Work Occupation (Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)
* relationship: Relationship Status (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
* race: Race (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
* sex: Sex (Female, Male)
* capital-gain: Monetary Capital Gains
* capital-loss: Monetary Capital Losses
* hours-per-week: Average Hours Per Week Worked
* native-country: Native Country (United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad & Tobago, Peru, Hong, Holand-Netherlands)

**Target Variable**

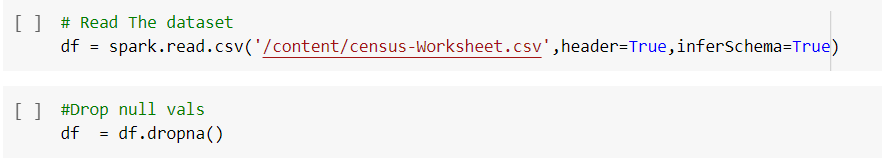
* income: Income Class (<=50K, >50K)

MODEL IMPLEMENTATION:

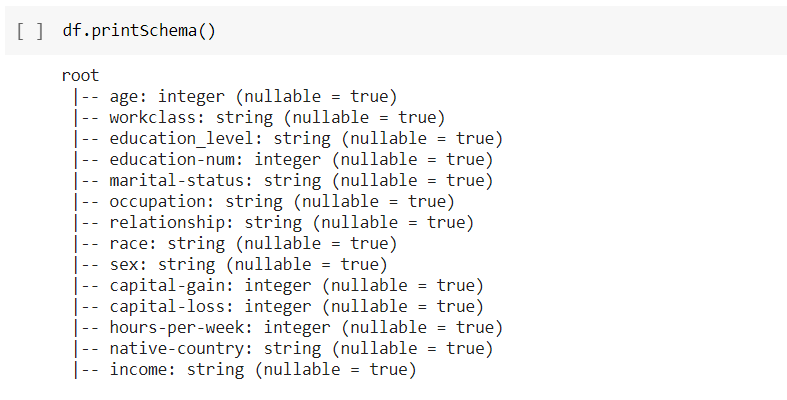
1. Firstly, we need to initialize pyspark



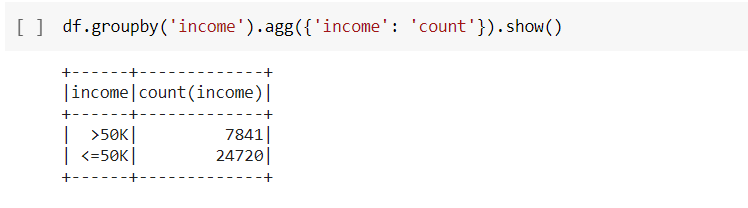
1. Now we import our dataset and drop all the null values



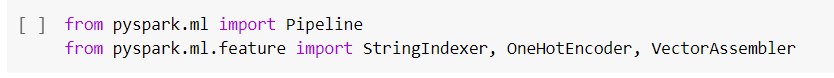
1. We now display the dataset’s schema



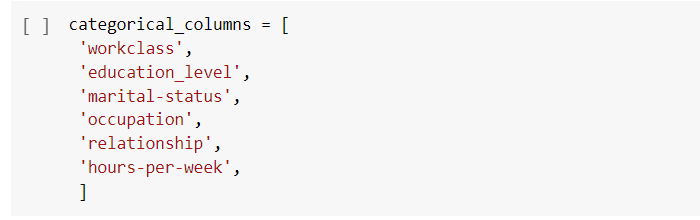
1. Observing target variable



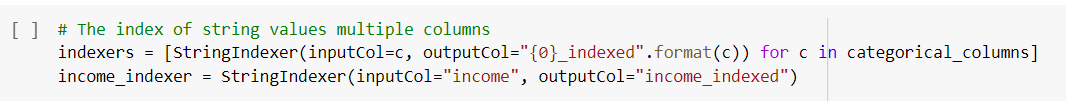
1. Now we import pipeline, string indexer, one hot encoder and Vector Assembler

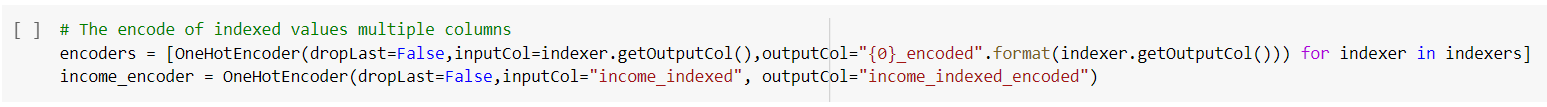


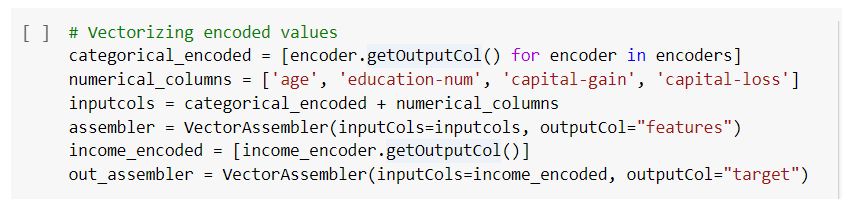
1. We select categorical features



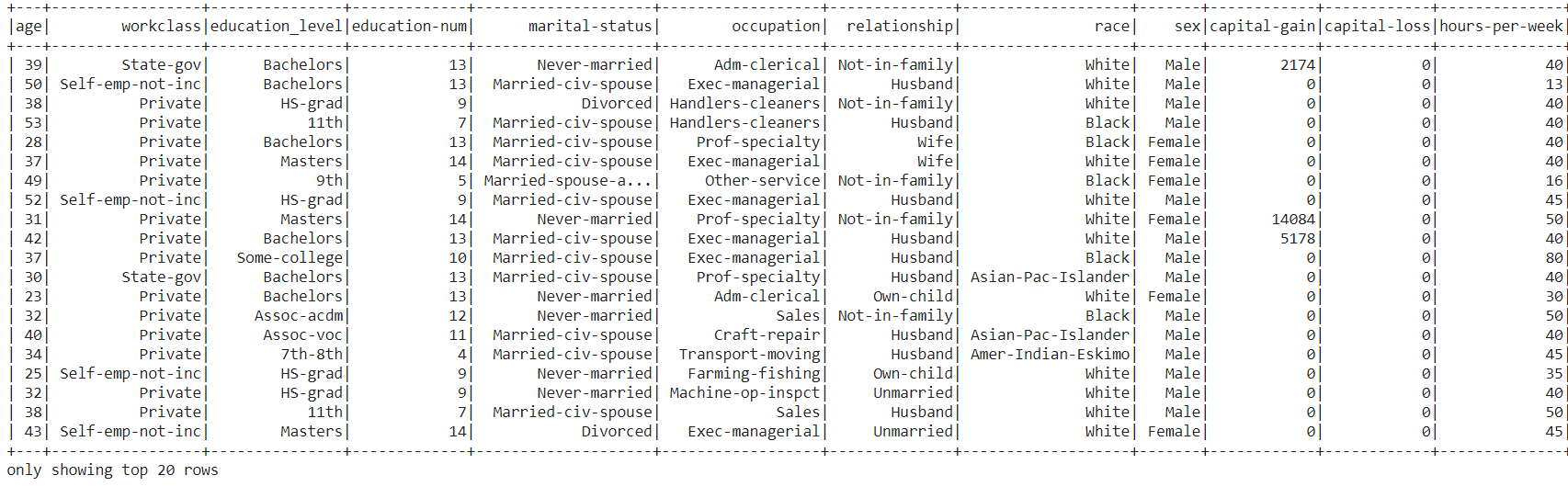
1. Implementing the string Indexer

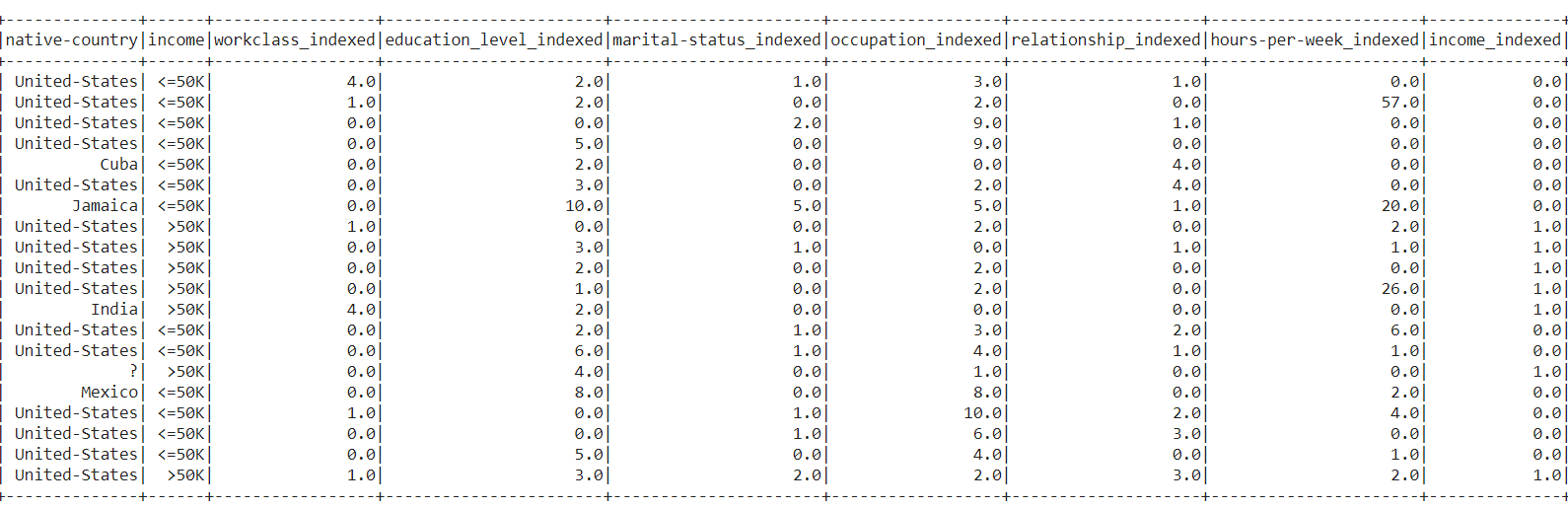


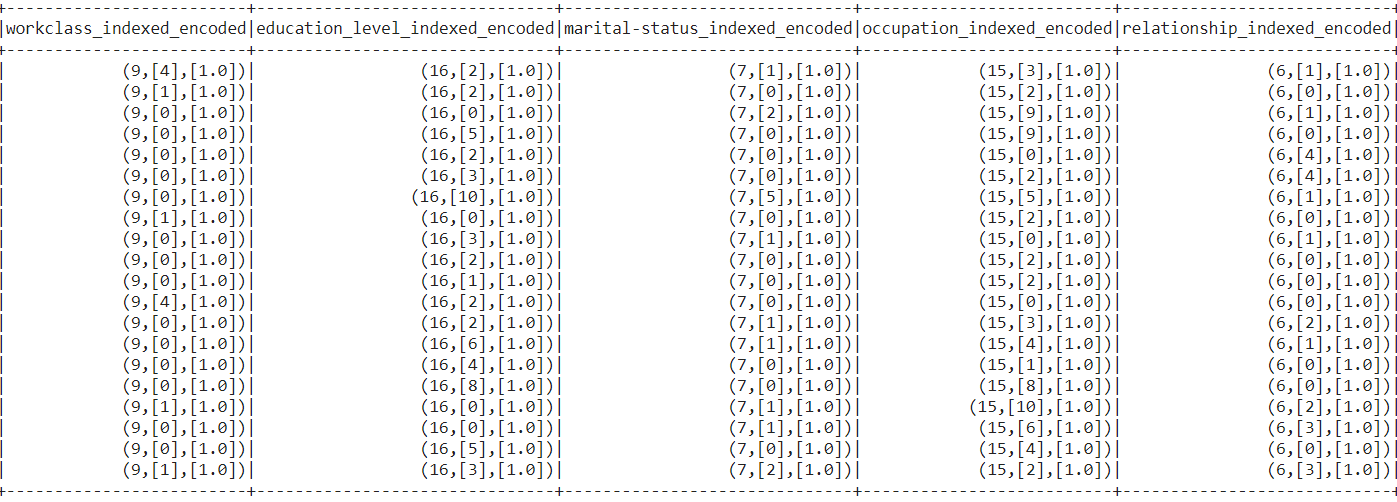
1. Implementing the one hot encoder
2. Vectorizing the encoded values

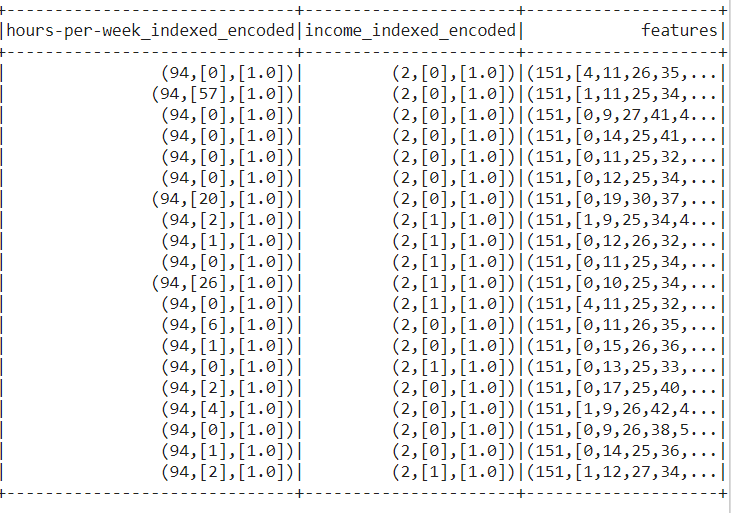


1. The pipeline and the transformed data

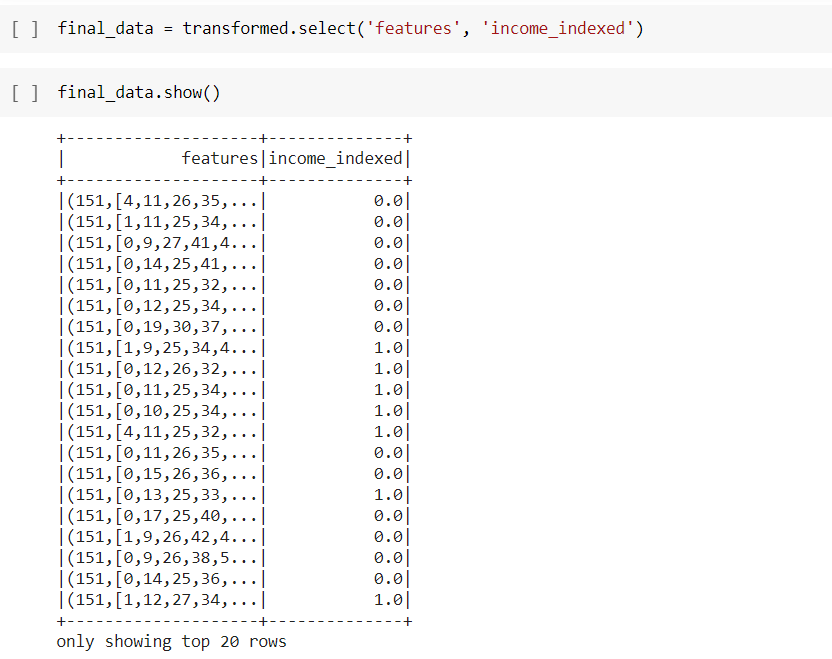






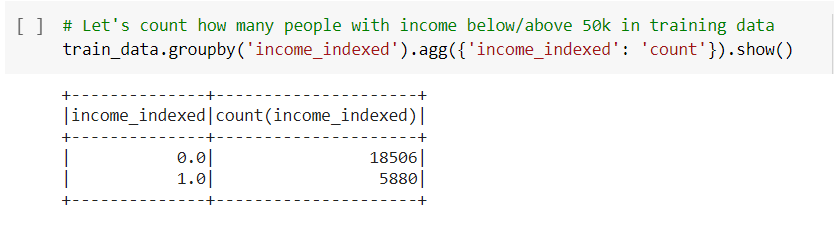


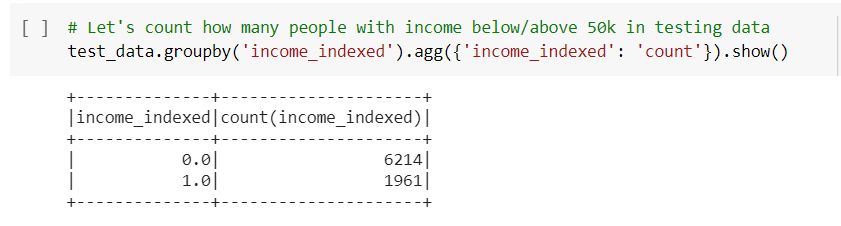
1. The indexed version of our target variable



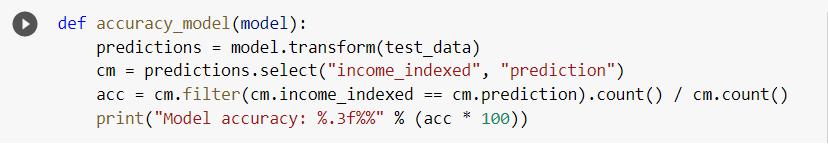
1. Splitting the data



1. People with income below/above 50k in training data
2. People with income below/above 50k in testing data

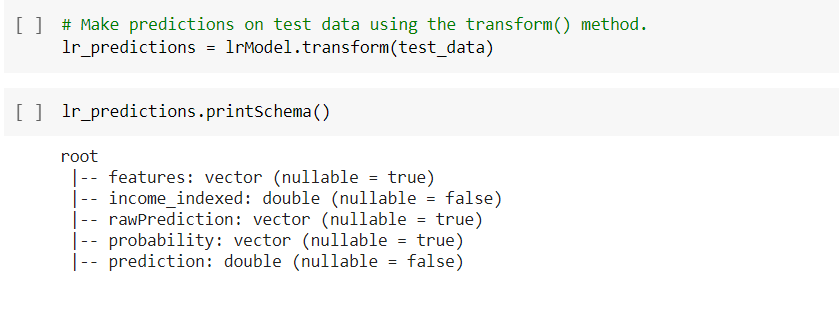


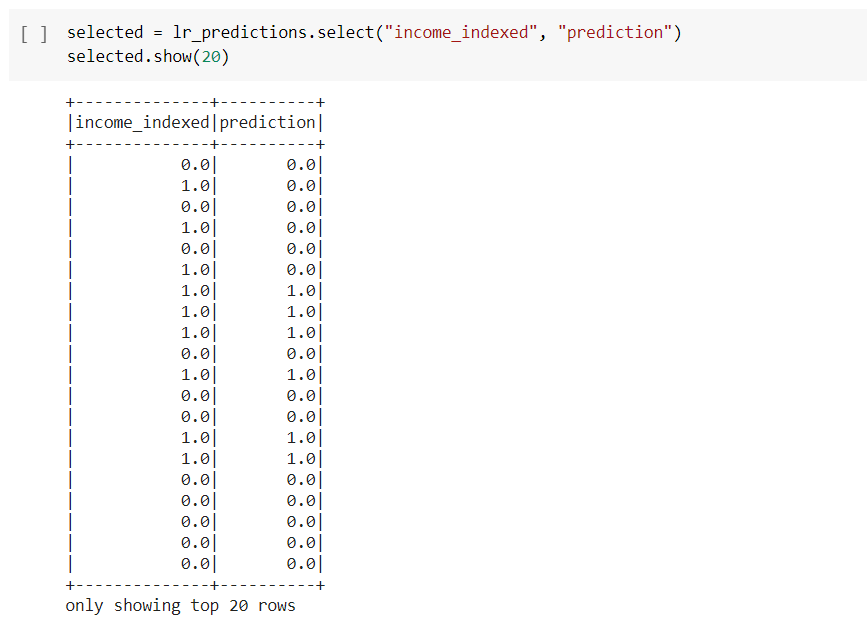
1. We need to look at the accuracy metric to see how well the model performs



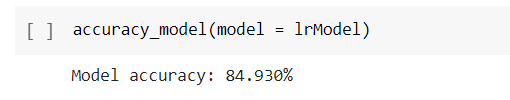
1. Logistic Regression



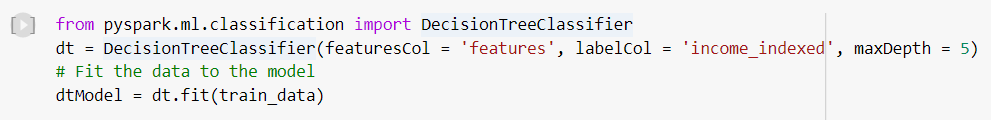


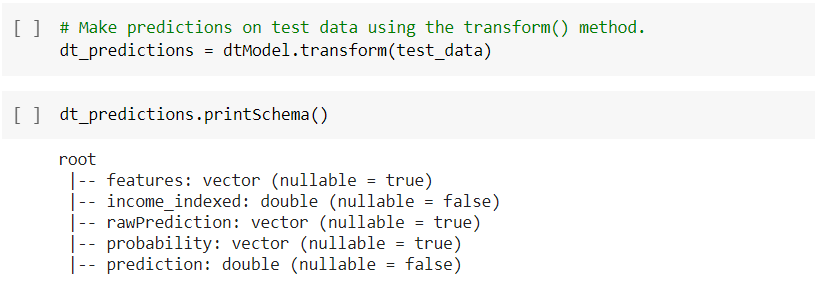


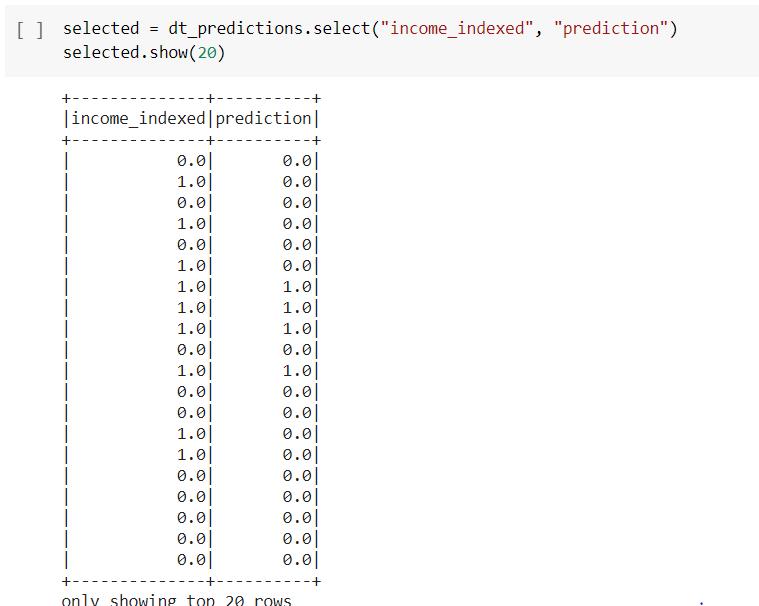
1. Accuracy of the model



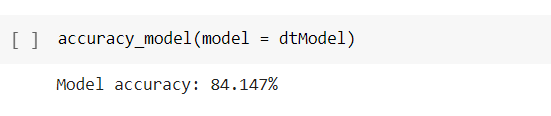
1. Decision Tree







1. Accuracy of the model



CONCLUSION:

In conclusion we have used both decision tree and logistic regression to classify our data. The classification accuracy in the test dataset of the of the two classifiers are shown and are very close to each other. By this models we can say that the employees whose income is greater than 50K are likely to donate for the charity.