CS634 Programming Assignment 2

**TASK 1: Parallel training on 4 ec2 Instances**

This project is implemented with Spark data frames API and MLib libraries, with this Native spark implementation application is automatically parallelized and distributed natively.

Wine prediction application is developed using Spark Data Frames and MLlib. Running it on AWS EMR cluster automatically parallelize and distribute job execution. Hadoop Distributed Files system is used for locating dataset files and for storing trained models.

**How to create EMR Cluster?**

* After logging into AWS console,
* Go to EMR Service & Create Cluster
  + Enter cluster name
  + Launch Mode cluster
  + Vendor Amazon
  + Release emr-5.3.10
  + Select spark application – version 2.4.5
  + Hardware Configurations
    - select the instance type
    - number of instances to 4(1 master 3 slaves)
  + select the ec2 key pair or generate one to access the master node.
  + click on create cluster.

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

**­­­­Upload files to EMR Cluster Master node**

1) Download the ppk key (as I am using PuTTY) from aws learner lab.

a. Start the lab.

b. Go to AWS Details tab in Learner lab.

c. Click on ‘Download PPK’

2) I used PuTTY to connect with ec2 instances. So, in PuTTY upload the downloaded ppk key and

start ssh session for master node instance.

3) I used WinSCP to upload files to master node ec2-instance. Create a directory ‘Data’ under /home/hadoop/ and upload TrainingDataSet.csv and ValidationDataset.csv to Data folder and upload app.jar in /home/hadoop.

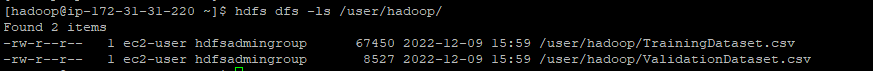
**Copy files to HDFS :**

* Now all files are on our master node we want to move them to HDFS so that all slave nodes can also access them, and we don’t have to manually copy them to all ec2 nodes.
* Use this command to copy files from Master node to HDFS.





* Use this command to verify if files are successfully copied to HDFS



**Launch ModelTrainer application :**

* Now everything is done, we want to launch Apache-spark application on EMR cluster.
* Execute following command to run application

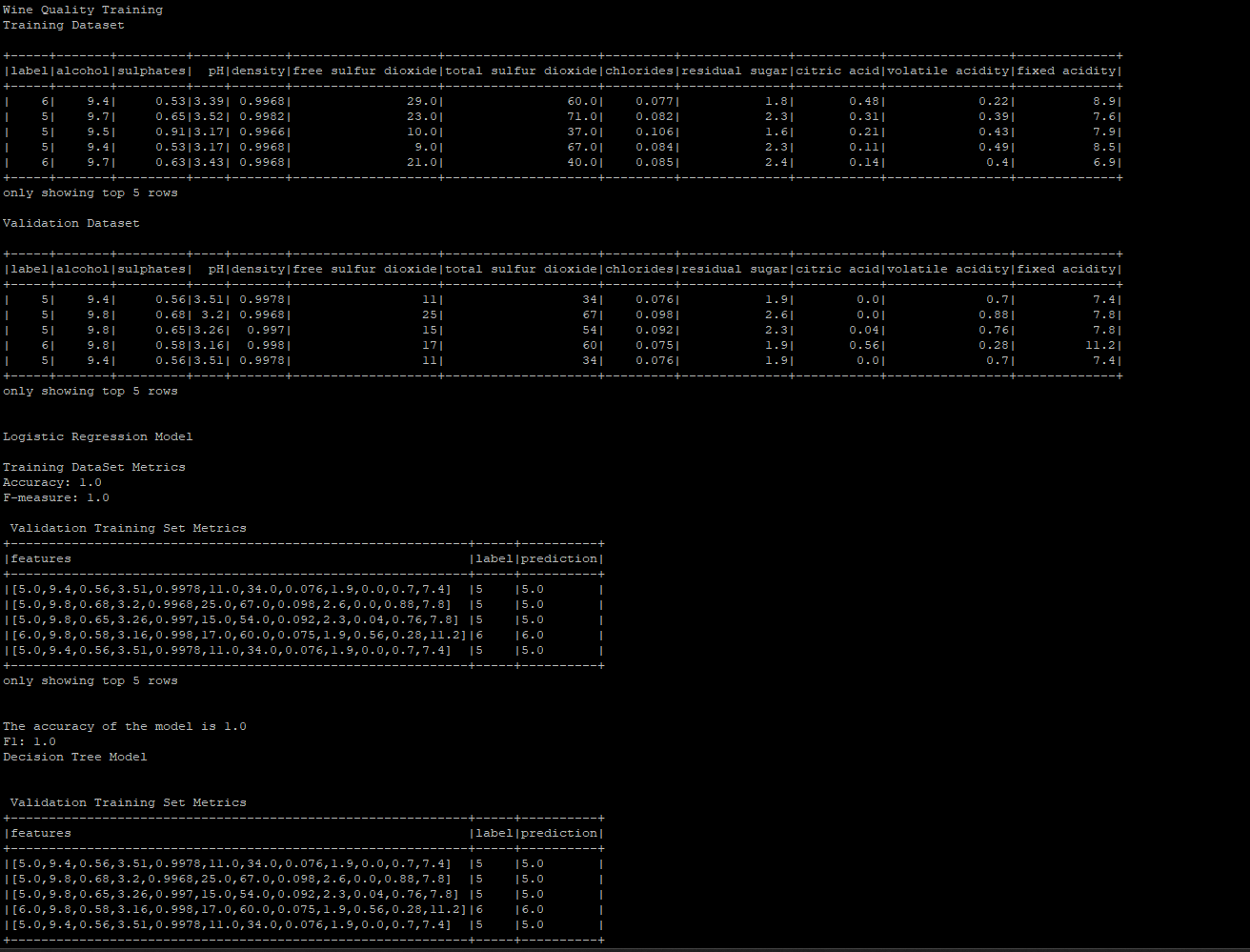


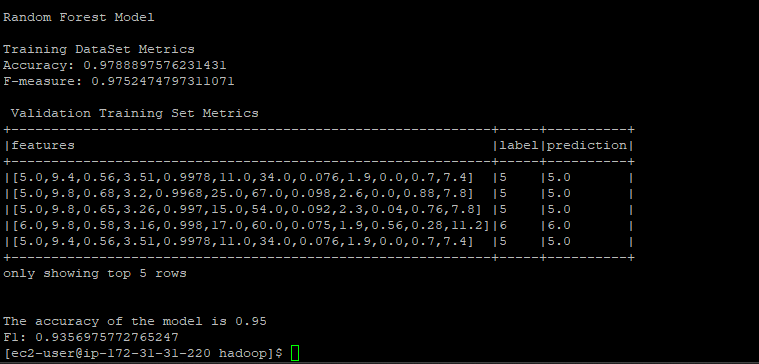
I used three different Machine Learning model for training

* LogisticRegression Model
* DecisionTreeClassifier Model
* RandomForestClassifier Model

I used Validation Dataset to check the training model F1 score and accuracy score.

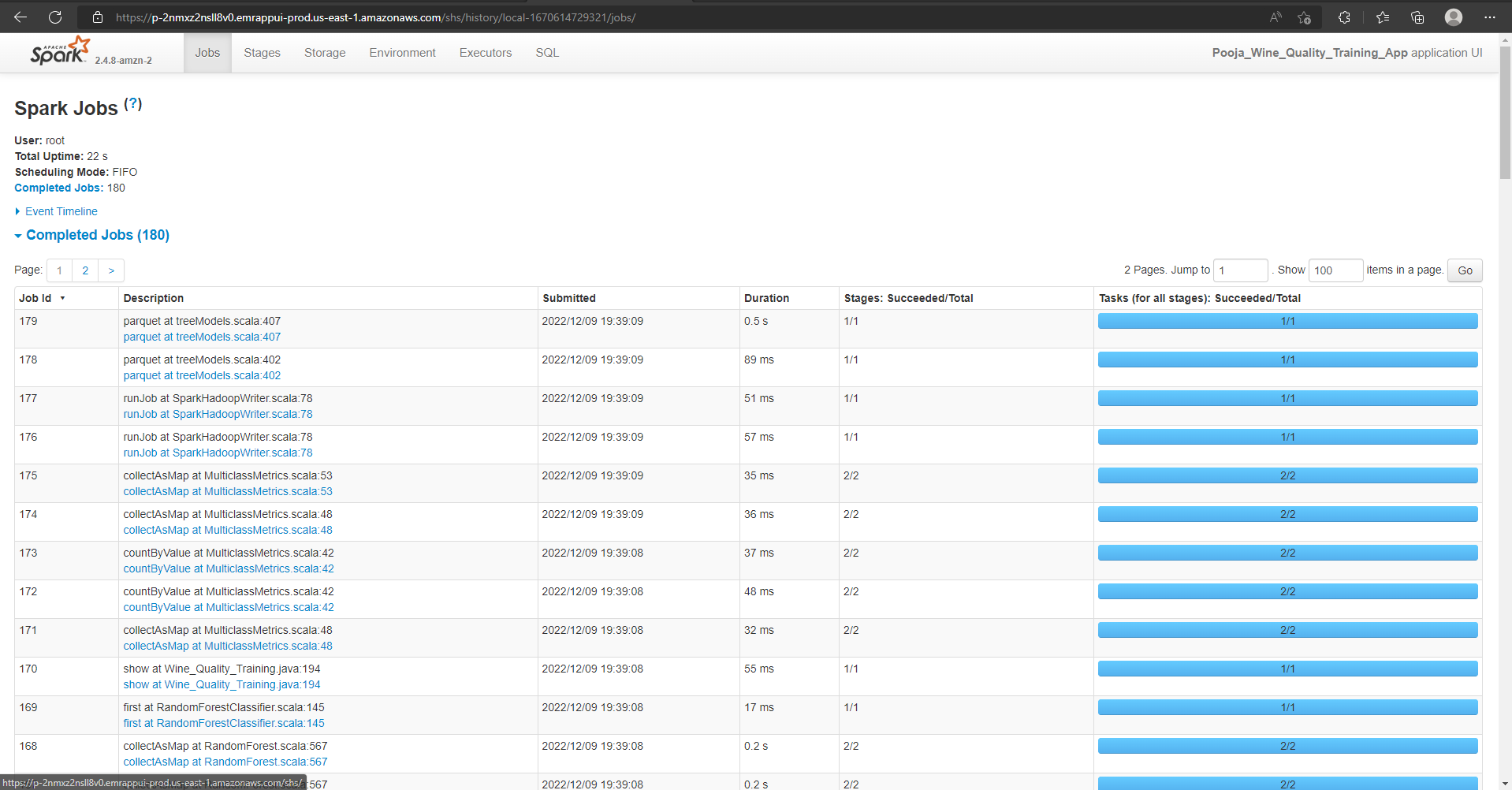
Output:



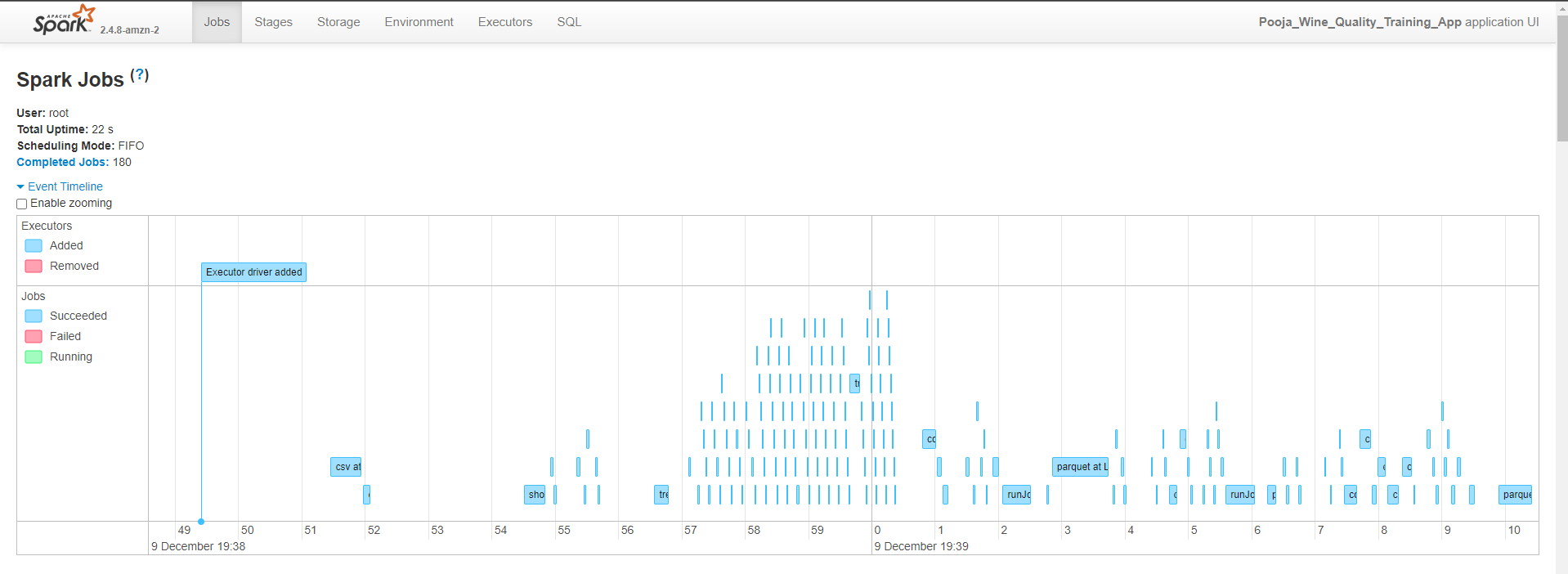


So, after seeing the F1 score, Decision Tree Model score is high than other models.

Job execution on Spark.



Event Timeline of Spark:



After running this, the ‘Model’ folder should have been created under /home/hadoop.

Copy that folder to local machine as we must use those models for prediction.

**TASK 2: Predict wine quality on single ec2 instance**

At this stage we are interested in executing prediction code on single ec2 instance. For that we need TestDataset.csv, prediction-app.jar and models folder (from task1)

**Ec2 instance Create:**

* After logging into AWS console,
* Go to EC2 -> launch instance, select AMI
* Select keypair and launch it.

Text

Description automatically generated

* **Install JAVA:**

You must update Java to latest version in each PuTTY terminal.

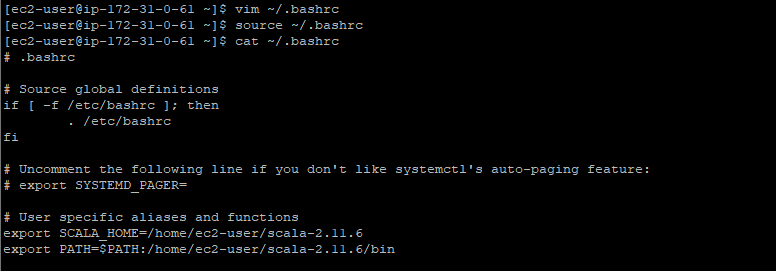
* + sudo yum install java-1.8.0-openjdk
* **Install SCALA:** 
  + wget http://downloads.typesafe.com/scala/2.11.6/scala-2.11.6.tgz
  + tar -xzvf scala-2.11.6.tgz



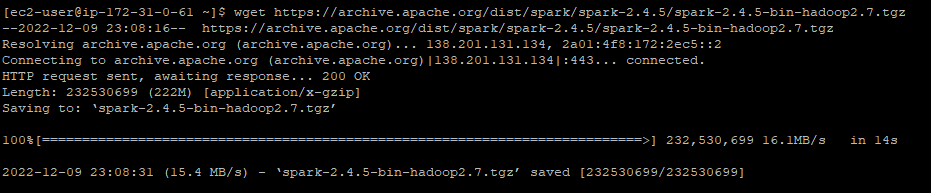
Text

Description automatically generated

* + Update PATH environment variable:
    - vim ~/.bashrc
    - copy following lines into file and then save it
      * export SCALA\_HOME=/home/ec2-user/scala-2.11.6
      * export PATH=$PATH:/home/ec2-user/scala-2.11.6/bin
    - source ~/.bashrc



* **Install SPARK:** 
  + wget https://archive.apache.org/dist/spark/spark-2.4.5/spark-2.4.5-bin-hadoop2.7.tgz
  + sudo tar xvf spark-2.4.5-bin-hadoop2.7.tgz -C /opt
  + sudo chown -R ec2-user:ec2-user /opt/spark-2.4.5-bin-hadoop2.7
  + sudo ln -fs spark-2.4.5-bin-hadoop2.7 /opt/spark
  + Update PATH Environment
    - vim ~/.bash\_profile
    - copy following lines into file and then save it
      * export SPARK\_HOME=/opt/spark
      * PATH=$PATH:$SPARK\_HOME/bin
      * export PATH
    - source ~/.bash\_profile



Text

Description automatically generated

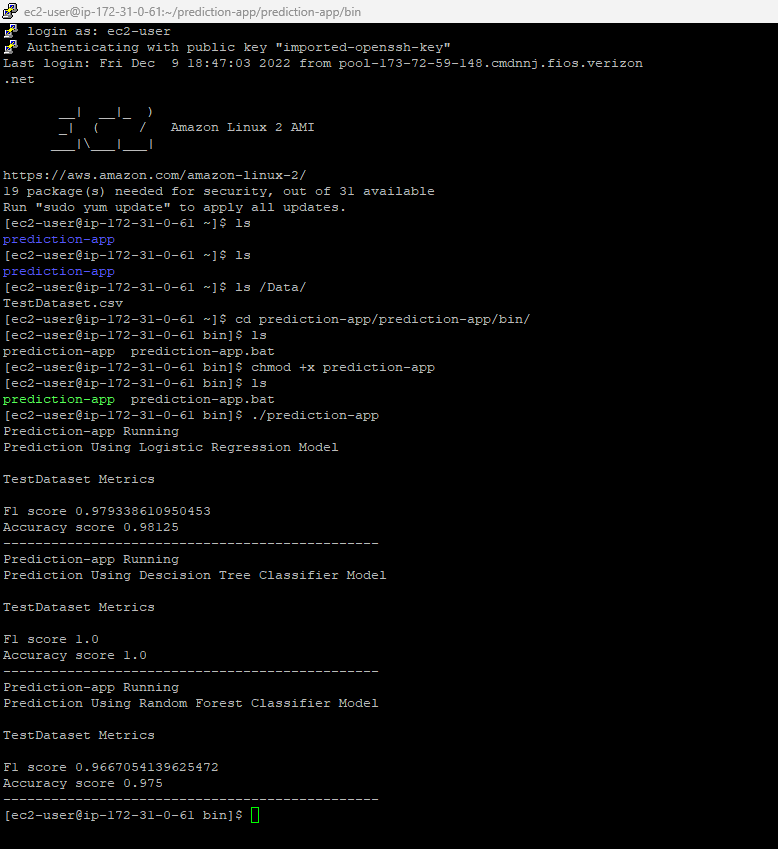
After all this, copy the files through WinSCP to ec2-user as shown in figure.

Graphical user interface, text

Description automatically generated

* **Disable unnecessary log4j :** 
  + cp $SPARK\_HOME/conf/log4j.properties.template $SPARK\_HOME/conf/log4j.properties
  + vi $SPARK\_HOME/conf/log4j.properties
  + (on line 19 of the file, change the log level from INFO to ERROR)
  + log4j.rootCategory=ERROR, console
  + Save the file and exit the text editor
* **Run wine-predict application:**





**TASK 3: Predict wine quality using docker**

For Predicting a wine quality on TestDataset.csv using docker. We need to provide local path of TestDataset.csv and pass it as an input argument to docker run command using -v flag. So that TestDataset.csv can be copied to docker container environment.

Test filename must be **TestDataset.csv,**and file must be placed under Data/ directory of container.

Two Ways for running my docker image, ruja2531/ps245\_wine\_quality:1.2, (found on docker-hub) :

1. Docker run -v <Host\_path>/TestDataset.csv:/Data/TestDataset.csv ruja2531/ps245\_wine\_quality:1.2
2. One can find in the project submission, docker-compose.yml file. Change line 11 <Host\_Path>/TestDataset.csv:/Data/TestDataset.csv

Text

Description automatically generated

**TASK 4 : Link to Github and DockerHub**

GitHub Link: [pookri/CS643\_Programming\_Assignment2 (github.com)](https://github.com/pookri/CS643_Programming_Assignment2)

DockerHub Link : [Image Layer Details - ruja2531/ps245\_wine\_quality:1.2 | Docker Hub](https://hub.docker.com/layers/ruja2531/ps245_wine_quality/1.2/images/sha256-bf0f19b305685897d89d3d05f53c8e44395d89d280c9740f0f891f8cedd211c4?context=repo)