Assignment

Sirui Wang

1. SQL:

```
SELECT
Users.UserId, UE.Employer, PT.PayDate, PT.EvaluationTime, PT.TransactionID
FROM
Users
LEFT JOIN
(SELECT
UE.UserId, UE.Employer, max(UE.CreatedOn)
FROM
UserEmployement UE
GROUP BY      UE.UserId, UE.Employer)
ON
Users.UserId = UE.UserId
LEFT JOIN
PayrollTracking PT
ON
UE.UserId = PT.UserId    AND    max(UE.CreatedOn) >= PT.PayDate
WHERE Users.CreatedOn BETWEEN '2017-01-01' AND '2017-05-31'
;
```

2.a and 2.b

Please see the attached Project.html/Project.pdf/Project.ipynb.

2.c

After we built and evaluated the classifier, we now would like to introduce the classifier to the back-end team. Here is a brief tutorial on how to apply the classifier to future data.

Step 1. Prepare input transaction data:

The format of the input data of the bank transactions is the same as that in BankTransactions_credit.csv file as below:

UserId	Transactio	Amount	Descriptior	PostedDat	Pending	Categories
458919	keNBRdad	3.33	TRANSFER	2016-12-1	FALSE	['Transfer', 'Credit']
458919	YV6B45w5	50.68	TRANSFER	2016-12-1	FALSE	['Transfer', 'Credit']
458919	EgkZXDPD	35	TRANSFER	2016-11-2	FALSE	['Transfer', 'Credit']
458919	KXxEMyky	10	TRANSFER	2016-10-2	FALSE	['Transfer', 'Credit']

In specific,

(1) The file must be separated by comma.

(2) The columns in the csv file must contain: UserID, TransactionID, Amount, Description, PostedDate. The names can be any names but must follow the above variable order.

(3) The PostedDate must be in format YYYY-MM-DDTXXXXX.

(4) Pending and Categories are not necessarily needed.

(5) Each record(row) can have missing value(s).

Step 2. Start the classifier (Namely Test.py):

The interface can be Windows or Linux, but Python needs to be installed.

(1) Open Windows Command/ Anaconda or Bash. Type "cd directory" (for example, cd C:\Users\wang\Desktop\Assignment) to access the directory where your Test.py has been placed.

(2) Type “python Test.py”. Then the following will prompt:

```
Please enter the path for bank transactions CSV file
```

Enter the path of your bank transaction CSV file. If the path is wrong, the program will quit. For example:

```
ccccc.csv
Cannot open ccccc.csv !
```

Apply the correct path, the following will prompt:

```
There are 3 records missing. Check Missing.csv for missing values!
```

If your dataset contains missing values, the program will detect them and gather those records with any missing value. The program will output Missing.csv file to the current directory for further check. However, the program will continue after it automatically drops the records with missing value(s).

```
Processing continues...

Dropping NULL records...
```

The program will finally output two CSV files to the current directory: Missing.csv and Output.csv. Output.csv contains the classified data.

```
Writing output to Output.csv file...

Writing missing values to Missing.csv file...

Success! Quitting...
```

Step 3. Look at the result:

The result is in Output.csv file. “NO” in column “result” indicates the transaction is classified as non-payroll and “YES” indicates the transaction is classified as payroll.

id	transid	amount	desc	date	result
458919	keNBRdad	3.33	TRANSFER	2016-12-1	NO
458919	YV6B45w5	50.68	TRANSFER	2016-12-1	NO
458919	EgkZXDPD	35	TRANSFER	2016-11-2	NO
458919	AZJ7nd6dI	36	TRANSFER	2016-10-2	NO
458919	5K67qVYV:	79	VISA NHUI	2016-10-1	NO

Step 4. Look at the missing value:

The Missing.csv file contains the records that have missing value(s) in them. This is for the user information check.

id	transid	amount	desc	date
458919	KXxEMykyaoTv77J3ZV		TRANSFER	2016-10-24T00:00:00-04:00
458919	geyDmX0XVMH3RRO		TRANSFER	2016-09-30T00:00:00-04:00
458919	peaNo151	7	TRANSFER	#982693 FROM CHECKING XXX-XXX-2820

Step 5: Feel free to try out the attached input.csv file (with 3 missing values).