SQL and Machine Learning Task Report

Yaman Dawod

Table of Contents

Part A: SQL

I decided to work on MySQL through MySQL Workbench 8.0. Imported the CSV files through the table data import wizard. To import the data I had to change the format of the dates to YYYY-MM-DD.

After importing the tables it was time to merge them using with SQL left join to prioritize the main employee data, storing the data in a new table 'result' :

```
create table result as select employee_master.*,
employee_performance.`PK`, employee_performance.`EmploymentRating`,
employee_performance.`DaysOfAbsence`,
employee_performance.`EducationLevel`,
employee_performance.`CertificationsEarned` from employee_master left
join employee_performance on
employee_master.`EEID`=employee_performance.`EEID`
```

We selected all columns from the master table and all tables except EEID from the performance table since there is already an EEID column in the master table.

Then we inserted 10 new rows of employees with the following line:

```
insert into result values (EEID,Full Name,Job Title, Department,
Business Unit, Gender, Ethnicity, Age, Hire Date, Annual Salary,
Bonus %, Country, City, Exit Date, PK, EmploymentRating,
DaysOfAbsence, EducationLevel, CertificationsEarned)
```

For each of the 10 employees added (full information in the table next page), for the IT employees we added:
Bob Dylan, EEID: E02323, Business Unit: Manufacturing, PK: 1001
Albert Wagner, EEID: E07986, Business Unit: Corporate, PK: 1006
Michelle Dumas, EEID: E09121, Business Unit: Research & Development, PK: 1008

For the conditional update the following line was used:

```
update result set City='Abu Dhabi' where City='Phoenix' and
Gender='Male'and Department='IT'
```

Finally each department information is exported by using the query:

```
select * from result where department='department name'
```

and then exporting each table on its own

New Employees Information:

Format: EEID | Full Name | Job Title | Department | Business Unit | Gender | Ethnicity | Age | Hire Date | Annual Salary | Bonus | Country | City | Exit Date | PK | Employment Rating | Days of Absence | Education Level | Certifications Earned

1. E02323 | Bob Dylan | Technical Architect | IT | Manufacturing | Male | Caucasian | 52 | 2000-04-05 | $123,000 | 0 | United States | Miami | 2017-10-04 | 1001 | 7 | 9 | PhD | 3

2. E01124 | Bruce Lee | Manager | Marketing | Specialty Products | Male | Asian | 30 | 2020-11-20 | $70,000 | 0.2 | United States | Chicago | 2024-09-10 | 1002 | 6 | 10 | Master's | 4

3. E02389 | John Doe | Business Partner | Human Resources | Research & Development | Male | Asian | 45 | 2015-07-20 | $65,000 | 0 | United States | Phoenix | 2022-02-16 | 1003 | 2 | 4 | Bachelor's | 2

4. E05656 | Eddy Gordo | Quality Engineer | Engineering | Research & Development | Male | Latino | 41 | 2005-12-30 | $94,120 | 0 | Brazil | Manaus | 2009-03-23 | 1004 | 3 | 11 | Bachelor's | 0

5. E00250 | Kay Tu | Analyst II | Finance | Research & Development | Female | Asian | 30 | 2020-03-27 | $59,120 | 0.4 | United States | Seattle | 2022-08-12 | 1005 | 10 | 0 | PhD | 4

6. E07986 | Albert Wagner | Systems Analyst | IT | Corporate | Male | Caucasian | 60 | 1998-10-03 | $65,120 | 0 | United States | Seattle | 2013-11-11 | 1006 | 7 | 6 | Bachelor's | 2

7. E09191 | Sarah Fortune | Director | Human Resources | Specialty Products | Female | Caucasian | 40 | 2012-01-20 | $170,600 | 0.25 | United States | Miami | 2022-06-21 | 1007 | 10 | 0 | Bachelor's | 3

8. E09121 | Michelle Dumas | Network Analyst | IT | Research & Development | Female | Black | 54 | 2007-09-20 | $201,762 | 0.1 | United States | Chicago | 2020-04-19 | 1008 | 7 | 4 | Master's | 3

9. E01112 | Michelle Yo | Sr. Analyst | Accounting | Specialty Products | Female | Asian | 62 | 2003-11-15 | $101,352 | 0.15 | United States | Seattle | 2020-01-01 | 1009 | 8 | 2 | Master's | 1

10. E03535 | Jimmy Kudo | Sr. Analyst | Accounting | Specialty Products | Male | Asian | 45 | 2010-07-10 | $171,352 | 0.05 | United States | Seattle | 2020-11-24 | 1010 | 9 | 2 | PhD | 2

Part B: ML

- Part 1 Loading and Cleaning Employee Data:

Data is loaded in using pd.read_csv and then cleaned with following steps:

  - Filling in missing ages through the mean of ages in the missing entries' job title
  - Transforming annual salary from string to float by removing all spaces, commas, and dollar signs using emp_IT['Annual Salary']= emp_IT['Annual Salary'].str.replace('$','') for each character
  - Filling in missing annual salaries through the mean of salaries in the missing entries' job title
  - Formatting exit and hire dates to dates through pd.to_datetime, formatted to Year-Month-Day
  - Set any employees with missing ethnicity in Brazil to latino and any employee in China to asian
  - Set any black or caucasian employees' missing countries to United States
  - Filling any missing bonus by the mean of the employee's job title, gender, and ethnicity
  - Filling in any missing cities with the most common city for the employee's country and ethnicity
  - Filling in missing job title for employee through what average annual salary matches his in his department

Original exit date column is dropped with the line:

```
emp_IT.drop(['Exit Date'],axis=1,inplace=True)
```

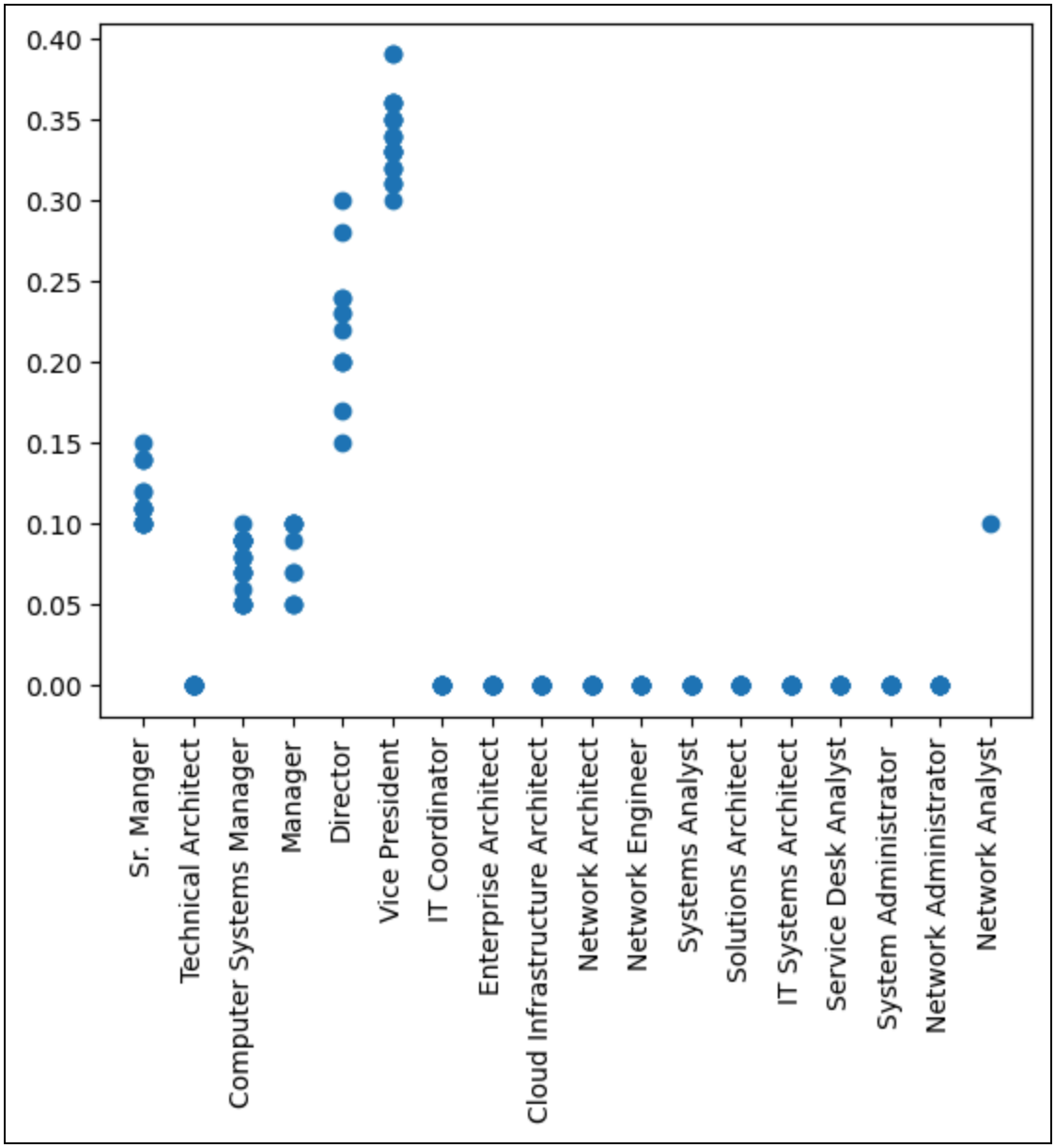
- Part 2 Exit-Date Prediction
  1. Loaded data set then checked for missing values , none were found.
  2. Changed exit date to form that could be used in training by adding a new column representing the days employees were employed at the company as well as the days after the earliest hired employee the employee was hired to represent the hire date as a number the model can be trained on.
  3. Removed outliers from data by only including data within 1.5 times the interquartile range of the 25th and 75th percentile
  4. Took the values to take into consideration (x) as age, annual salary, bonus, and hire time (now in days format), only took the training data from employees in the IT department to keep the information curated to the employees
  5. Took the result value (y) as the days between exit and hire date
  6. Separated the training data into 75% training data and 25% testing data using train_test_split
  7. Trained the model using linear regression
  8. Stored predicted data of testing data in pred_y
  9. Calculated the mean absolute error and root mean squared error
  10. Added new column to the original IT employees dataframe with the exit date calculated by adding the days to the hire date

The MAE was 808 while the RMSE was 964, which are pretty high numbers, however the training data decreased significantly from ~2000 entries to 380 by focusing on only training data from IT employees. Decreasing the number of entries could have significantly affected the accuracy of the model overall but since the task focuses on IT employees we decided it should be fine.

- Part 3 Bonus Percentage Prediction
    1. Loaded data set then checked for missing values , none were found.
    2. Fixed spelling error from bouns to bonus
    3. Created new column ‘BF’ to be a bonus float by removing the % from the bonus value using str.replace('%','')
    4. Encoded education level by its stage of education, Bachelor's:0, Master's:1, Doctorate: 2
    5. Removed outliers from data by only including data below 1.5 times the interquartile range after the 75th percentile, the lower ranges were kept since the bonus values can include a lot of 0 values and if there is not a lot of 0 values any that present are welcome
    6. Took the values to take into consideration (x) as employment rating, days of absence, certifications earned, and encoded education level
    7. Took the result value (y) as the ‘BF’, the float version of the bonus column
    8. Separated the training data into 75% training data and 25% testing data using train_test_split
    9. Trained the model using linear regression
    10. Stored predicted data of testing data in pred_y
    11. Calculated the mean absolute error and root mean squared error
    12. Added new column to the original IT employees dataframe with predicted bonus value by adding an encoded education level column to use in the model

The MAE was 0.103 and the RMSE was 0.117. 254 employees received less than predicted, on the surface level this seems very extreme and could possibly show that many employees are receiving less than they deserve, however this fails to put into consideration the very flawed training data, which does not take into account the department and job title, which is a major flaw considering that different departments and job titles will have different structures to the annual salary and bonus amount. This flaw becomes very apparent in the task given, as most employees in IT, 193 out of 288 to be exact, do not have bonuses at all, and this is not an oversight or undervaluing of the employee situation, most jobs in IT have 0 bonus by design, however since the training data includes no information on department or job title the model is left to compare the bonuses of departments with very different payment structures to IT employees. Add to that that the training data has only 23 instances of 0 bonus out of 1000 and that proves that it not curated to the task at all and as such has trained the model very inaccurately to predict anything that is affected by department or job title.

Graph showing bonuses per job title in IT, most jobs in IT get no bonuses:



Graph of bonuses at each job title for IT employees