UNIVERSITY OF SAN FRANCISCO | College of Arts and Sciences

MS in Data Science

Table of Contents

MS in Data Science

Name of the Program

MS in Data Science

Degree Type

Graduate

Contact Information

Diane Woodbridge

Program Academic Director/Associate Professor

MS in Data Science

dwoodbridge@usfca.edu

Mission Statement

The mission of our program is to produce graduates who possess a theoretical and practical understanding of many classical and modern statistical modeling and machine learning techniques; who use contemporary programming languages and technologies to scrape, clean, organize, query, summarize, visualize, and model large volumes and varieties of data; and who use their knowledge and skills to successfully solve real-world data-driven business problems and to communicate those solutions effectively.

Notes: The MSDS program's mission statements were ratified by its faculty during a January 2016 vote over email

Program Learning Outcome

Our program learning outcomes (PLOs) were changed in December of 2018 and ratified by a vote of the faculty to the following:

PLO 1. Possess a theoretical understanding of classical statistical models (e.g., generalized linear models, linear time series models, etc.), as well as the ability to apply those models effectively.

PLO 2. Possess a theoretical understanding of machine learning techniques (e.g., random forests, neural networks, naive Bayes, k-means, etc.), as well as the ability to apply those techniques effectively.

PLO 3. Effectively use modern programming languages (e.g., R, Python, SQL, etc.) and technologies (AWS, Hive, Spark, Hadoop, etc.) to scrape, clean, organize, query, summarize, visualize, and model large volumes and varieties of data.

PLO 4. Prepare for careers as data scientists by solving real-world, data-driven, business problems with other data scientists, and understand the social, ethical, legal, and policy issues that increasingly challenge and confront data scientists.

PLO 5. Develop professional communication skills (e.g., presentations, interviews, email etiquette, etc.), and begin integrating with the Bay Area data science community.

Current Curricular Map

There has been a few changes in our curriculum.

1. Swapping the sequences of MSDS 626 and MSDS 694 - MSDS 626 was offered in intersession, and MSDS 697 was offered in Spring 2. Since there are no dependencies between the two courses, the changes did not affect the rearrangement or curriculum updates of any other courses.
2. Introducing MSDS 699: Machine Learning Lab - We newly designed and introduced the course as machine learning skill sets and practices are most wanted for students' careers.
3. Introducing MSDS 689: Data Structures and Algorithms - To help students understand fundamental computer science topics and pass technical interviews and coding challenges, we introduced this course to cover core concepts for writing efficient programs.

Program Phase	Bootcamp				Fall Module One					Fall Module Two						Intersession	Spring Module One					Spring Module Two					Summer 2			10 required hours of interview training
	MSDS 501: Computation for Analytics	MSDS 502: Review of Linear Algebra	MSDS 504: Review of Probability and Statistics	MSDS 593: EDA and Visualization	MSAN 691: Relational Databases	MSDS 601: Linear Regression Analysis	MSDS 692: Data Acquisition	MSDS 610: Communications for Analytics	MSDS 640: Seminar Series I	MSDS 621: Introduction to Machine Learning	MSDS 699: Machine Learning Laboratory	MSDS 604: Time Series Analysis	MSDS 694: Distributed Computing	MSDS 605: Practicum I	MSDS 641: Seminar Series II	MSDS 629: Experiments in Data Science	MSDS 697: Distributed Data Systems	MSDS 630: Advanced Machine Learning	MSDS 689: Data Structures and Algorithms	MSDS 625: Practicum II	MSDS 642: Seminar Series III	MSDS 633: Ethics in Data Science	MSDS 626: Case Studies in Data Science	MSDS 603: Product Analytics	MSDS 627: Practicum III	MSDS 643: Seminar Series IV	MSDS 631: Special Topics in Analytics	MSDS 627: Practicum IV	MSDS 644: Seminar Series V	
Program Phase	Bootcamp				Fall Module One					Fall Module Two						Intersession	Spring Module One					Spring Module Two					Summer 2			
Units	1	1	1	1	1	2	2	1	0	2	1	2	1	1	0	2	2	2	1	2	0	1	2	2	2	0	2	1	0	0
Program Learning Outcome 1. Possess a theoretical understanding of classical statistical models (e.g., generalized linear models, linear time series models, etc.), as well as the ability to apply those models effectively.			I	D		I						D		I		M				D			D			M		M	M	M
Program Learning Outcome 2. Possess a theoretical understanding of machine learning techniques (e.g., random forests, neural networks, naive Bayes, k-means, etc.), as well as the ability to apply those techniques effectively of data.						I				D	I	I		I			D	M		D		I	D	D	D		M	M		
Program Learning Outcome 3. Effectively use modern programming languages (e.g., R, Python, SQL, etc.) and technologies (AWS, Hive, Spark, Hadoop, etc.) to scrape, clean, organize, query, summarize, visualize, and model large volumes and varieties of data.	I	I		I	D		D			D		I	M	I			D	M	D	D			D	M	D			M		
Program Learning Outcome 4. Prepare for careers as data scientists by solving real-world, data-driven, business problems with other data scientists, and understand the social, ethical, legal, and policy issues that increasingly challenge and confront data scientists.				D	I		I	I						D		D				D		M	D	D	M			M		M
Program Learning Outcome 5. Develop professional communication skills (e.g., presentations, interviews, email etiquette, etc.), and begin integrating with the Bay Area data science community.								I	I					I	D					D	M	I		D	M	M		M	M	D

Assessment Schedule between APRs

We are assessing 5 PLOs throughout the 3-year plan.

Year	Assessment Schedule
Year 1	Assessing PLO 1 based on statistics courses including MSDS 502, 504, 602, 604, and 623
Year 2	Assessing PLO 2, 3 and 4 based on computational courses including MSDS 621, 630, 689, 691, 692, 694, 697, and 699
Year 3	Assessing PLO 5 based on communication and business courses including MSDS 610 and 603

This report presents the assessment of PLO 2, PLO 3, and PLO 4. The last assessment was done in 2017-2018.

Description of the Assessment Methodology

Three faculty (Yannet Interian, Terence Parr, and Diane Woodbridge) in MSDS contributed representative questions from one or more written exams taken from eight data science courses related to the three learning outcomes mentioned above.

These learning outcomes are loosely described as computing–related.

Here are the number of questions broken down by program learning objectives and courses:

Module	Course	PLO2	PLO3	PLO4
Fall 1	MSDS 692: Data Acquisition	0	3	2
	MSAN 691: Relational Databases	0	4	1
Fall 2	MSDS 621: Introduction to Machine Learning	5	0	0
	MSDS 699: Machine Learning Laboratory	3	0	1
	MSDS 694: Distributed Computing	0	3	2
Spring 1	MSDS 630: Advanced Machine Learning	4	0	1
	MSDS 689: Data Structures and Algorithms	2	2	0
	MSDS 697: Distributed Data Systems	0	2	4
	Total	**14**	**14**	**11**

All exam questions used in the creation of this assessment report are provided in the Appendix section.

Rubrics

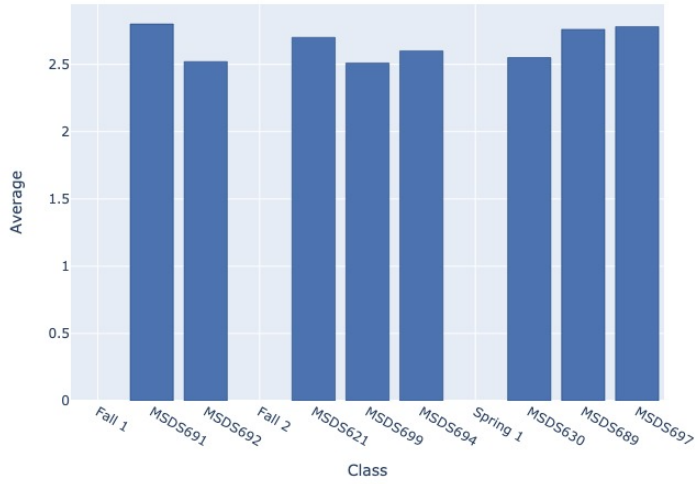
We decided on a scoring mechanism that could be used across all courses and all questions on those exams used in this report:

Score	Description
3	Student has mastered material necessary to answer a specific question
2	Student did not give a perfect answer to the question but had a solid grasp on the concepts
1	Student did not achieve a minimum level of competency for a specific question
0	Student did not answer

For the purposes of this assessment report, faculty returned to their exams, sometimes months after initially grading the students, to select and score assessment questions as 3, 2, or 1.

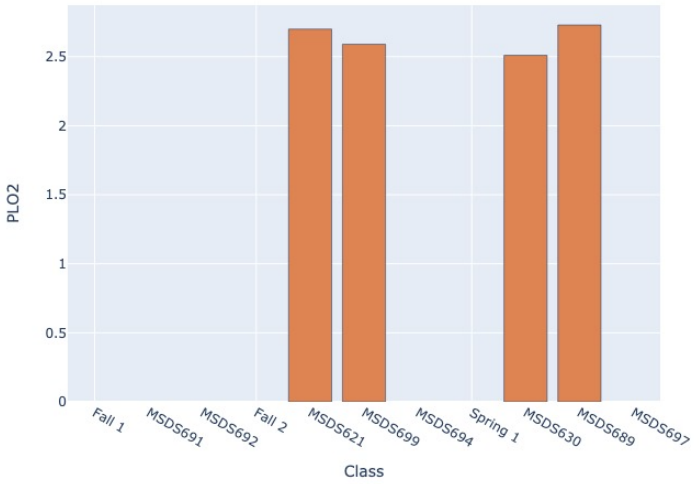
Description of Results

The averages of three PLOs for all eight courses ranged from 2.43 to 2.89, where each module had 2.66, 2.60, and 2.70, respectively, showing that students achieved close to “mastery levels”.



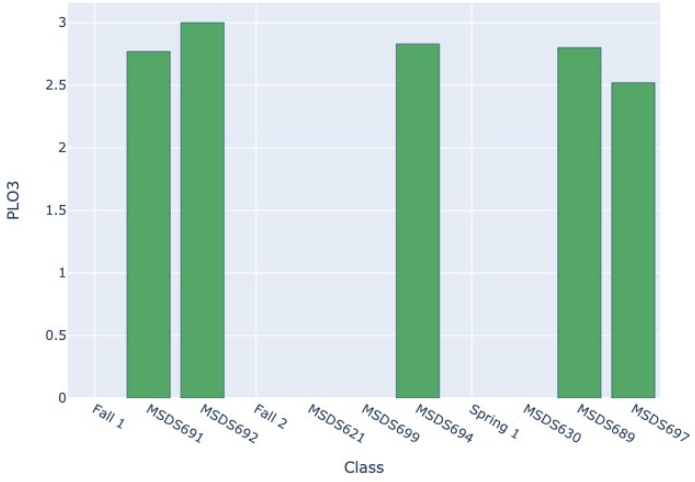
PLO2. Possess a theoretical understanding of machine learning techniques (e.g., random forests, neural networks, naive Bayes, k-means, etc.), as well as the ability to apply those techniques effectively.

We offered two courses covering PLO2 per module in Fall 2 and Spring 1 modules. PLO2 ranged from 2.51 to 2.73. MSDS 630, Advanced Machine Learning, received the lowest score, 2.51. However, this is the most advanced course in the program that requires theoretical knowledge of statistical modeling, machine learning, and technical skills. At the same time, there are two other technical courses with many assignment deadlines offered in the same module. Due to the rigor of the program, some may struggle to master the subjects. Still, the exit survey shows that the topics covered in this class have been the most helpful for preparing for job interviews and applying to real-world problems at internships and work.



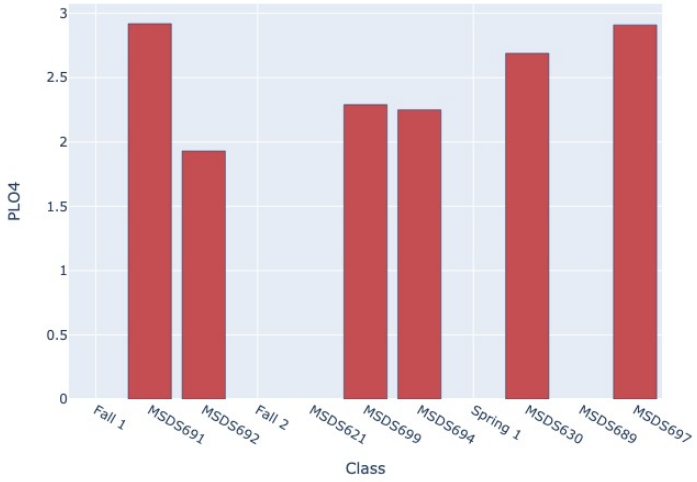
PLO3. Effectively use modern programming languages (e.g., R, Python, SQL, etc.) and technologies (AWS, Hive, Spark, Hadoop, etc.) to scrape, clean, organize, query, summarize, visualize, and model large volumes and varieties of data.

We offered two courses in Fall 1, one course in Fall 2, and two courses in Spring 1 covering PLO3. As many of our students are not coming from computer science undergraduate programs, we embed PLO3 through every module and assess student progress. This year, the PLO3 score ranged between 2.52 to 3. MSDS697, Distributed Data Systems received the lowest score of 2.52. - this course covers data engineering topics about distributed databases, distributed computing, and various cloud computing, including AWS, MongoDB Atlas, and Databricks. Most students are not familiar with the concepts or topics, regardless of their previous degrees, as many of the tools and topics are emerging in the tech industry. This means that no textbooks are available, and online documentation is the best way to review/learn them. In addition, MSDS697 was offered with two other technical courses with many assignment deadlines offered in the same module. However, this helps students to widen their career opportunities to be data engineers, and the students requested more data engineering and ML Ops courses at their exit survey this year.



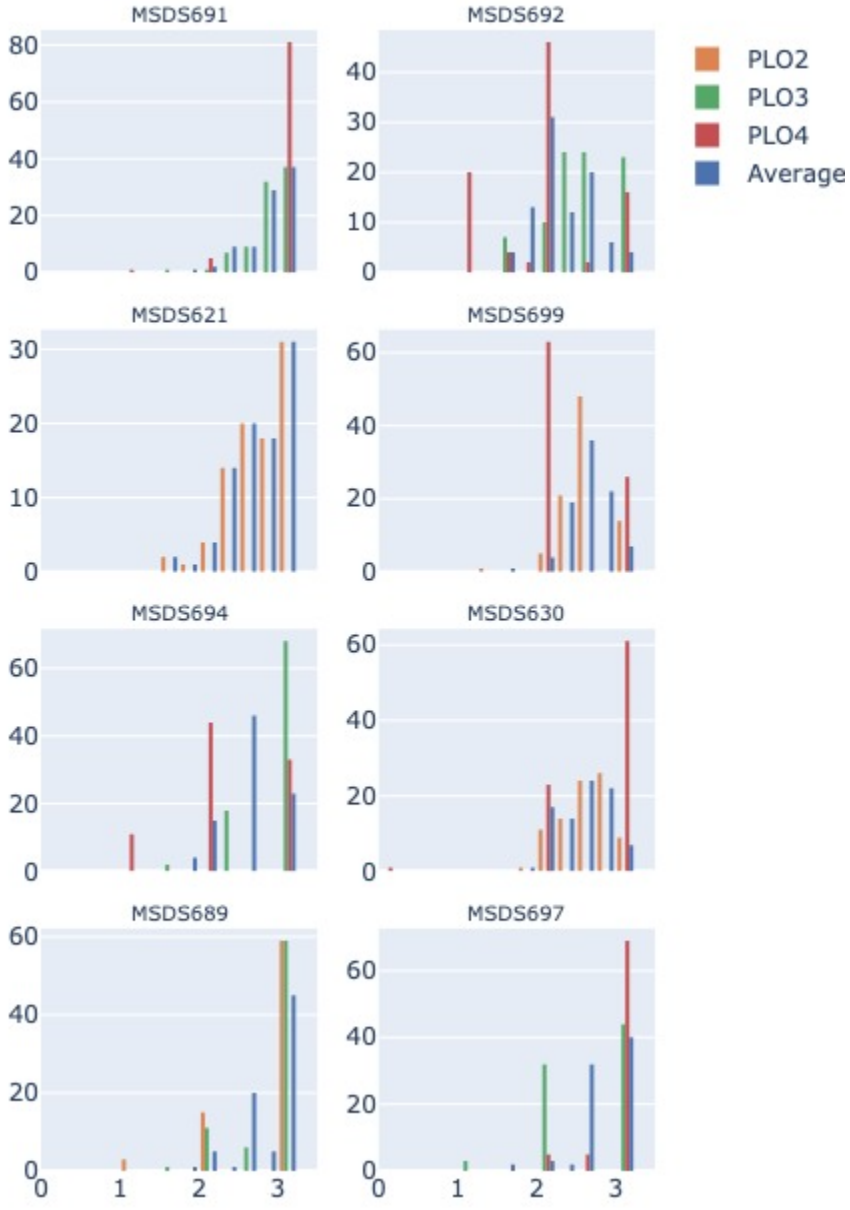
PLO4. Prepare for careers as data scientists by solving real-world, data-driven, business problems with other data scientists, and understanding the social, ethical, legal, and policy issues that increasingly challenge and confront data scientists.

For evaluating PLO4, we chose questions that used real-world data/scenarios. Two courses per module included PLO4 for assessment, where the score ranged between 1.93 and 2.91. The trends from Fall 1 to Spring 1 show the improvement in students' ability to apply theoretical and technical knowledge to real-world problems. Although MSDS691 Relational Databases have a high score, this is not representing the student progress as many of the students already took the course in their undergraduate programs or used them at their work.



Description of Sharing and Responses

In most classes, students achieved between a solid to a mastery level and showed improvement over the year. We also investigated the distribution of average grades per student in PLO2, PLO3, and PLO4. In most classes, the grades were normally or close to normally distributed. Although PLO4 for MSDS 699 Machine Learning Laboratories seemed left-skewed, there was only one question in the sample, and most students received 2 or 3. We also see that some classes have right-skewed grade distributions due to the high quality of the student population in the cohort, where many have prior work experience as data scientists or data analysts.



We were able to see that instructors included more questions reflecting PLO4 this year, showing increases from 19.6% to 28.2%, since the 2018 assessment. Using real-world data and creating questions associated with social, ethical, legal, and policy issues are time-consuming for instructors but can greatly motivate student learning and improve their knowledge and skill sets. Including more questions with PLO4 would well reflect Ignatian pedagogy of context, experience, reflection, action, and evaluation. Students also practiced PLO4 through their practicum, which started during the Fall 2 module and helped students cultivate their abilities to apply theoretical knowledge and technical skills to real-world problems. Students' average grades in PLO4 improved over the year from 2.43 to 2.80 by the end of the Spring 1 module.

Discussion

The cohort that we assessed this year mostly achieved a high standard in the MSDS program, reaching solid to mastery grades. Thanks to the effort from our faculty to provide high-quality education and adjust the curriculum to help our students have the skill sets that the industry is seeking, the class of 2022 achieved a 95% employment rate with a $135K median salary within 5 weeks of graduation.

The feedback from AY 2017-2018 reviews includes “PLOs lack the use of active verbs and are made difficult to assess”. As MSDS went through a loss of many faculty who used to teach computational subjects, we, unfortunately, haven’t had a chance to update our PLOs. We are planning to have a curriculum meeting in November 2022, and restructuring the class subjects and updating PLOs will be reviewed.

The evaluation was done by three faculty members who are teaching computational courses collectively. We asked each instructor to place the questions and anonymize students' answers in a shared folder where everyone could review them. Also, the final report was shared with all the other faculty members in the program for reviews to accurately measure the learning objectives and assess outcomes.

We would consider having better metrics and methods for assessing students. Most questions used for this assessment are either multiple choices or True/False, causing scores to be either 1 or 3. For a better assessment, we will come up with new rubrics that can reflect student learning at a more granular level.

In addition, in our curriculum meeting, we will strategize to balance the number of questions in each PLOs to promote balanced learning and evaluate student progress. Consulting with FDCD, the Center for Teaching Excellence (CTE), and Educational Technology Services (ETS) to better align PLOs with each course might be an option as well.

In Spring 2022, our colleague, Terence Parr left USF, which is a huge loss to the program and school. Terence is a knowledgeable and skilled instructor whom many students and faculty respect. Currently, we are planning to use many of his materials to provide quality instruction and launched a search for a replacement. However, due to the competitive nature of hiring computer scientists in industry and academia, it might not be possible to hire an experienced instructor with a computer science background. This may cause an increased workload for other faculty and possibly affect student learning.

Appendix

MSDS 692: Data Acquisition Questions

Q1. Including the network fetch of the following HTML page from a remote server, how many fetches does a browser make in order to render this page?

Q2. If a web server sends JavaScript code that generates HTML+data dynamically (instead of sending HTML+data directly), how can we extract data from that server's response? I'm looking for the name of a Python tool or library.

Q3. You would like to pull data from a REST API at machine apibot.com and at file resource called /f.xml with parameters id and x. You need to set id to 5 and x to string mary. Give the exact and complete URL needed to perform the API fetch, starting with "http:".

Q4. How can a news website server track how many times your browser reads articles on that site using a single cookie? Assume no server can store anything about you or your visits. (My answer has 5 words.)

Q5. If a browser sends the following HTTP GET request to a server, what exact and complete URL is the browser requesting? (Your answer must start with "http:")

GET /x/y HTTP/1.1

Host: hello.org

Cookie: ID=99

MSAN 691: Relational Databases Questions

Q1.

For the following table called ***weather***, write a SQL query to perform the given search.

```
        column_name         | data_type
----------------------------+-----------
zip                         | integer
date                        | date
temperature                 | real
history_avg_temperature     | real
precipitation               | real
history_avg_precipitation | real
```

An example data table is given below. However, your code should be generic and work with any valid data.

```
 zip  |    date    | temperature | history_avg_temperature | precipitation | history_avg_precipitation
-------+------------+-------------+-------------------------+---------------+--------------------------
94105 | 2021-09-01 |          65 |                   62.53 |             0 |                        0
94105 | 2021-08-31 |       63.08 |                    65.4 |             0 |                        0
94105 | 2021-08-30 |       61.69 |                    65.4 |             0 |                        0
94105 | 2021-08-29 |       63.83 |                    65.3 |             0 |                        0
70726 | 2021-09-01 |       81.17 |                    81.5 |             0 |                      3.6
70726 | 2021-08-31 |       80.29 |                    83.2 |          0.71 |                      4.7
70726 | 2021-08-29 |       74.35 |                    81.9 |          0.09 |                      3.7
```

*You can create the shipping table using the following queries.

```
CREATE TABLE weather
(
 zip INTEGER,
 date DATE,
 temperature REAL,
 history_avg_temperature REAL,
 precipitation REAL,
 history_avg_precipitation REAL,
 PRIMARY KEY (zip, date)
);


INSERT INTO weather VALUES
(94105, '2021-09-01',    65,       62.53,   0.00,    0.00),
(94105, '2021-08-31',    63.08,    65.4,    0.00,    0.00),
(94105, '2021-08-30',    61.69,    65.4,    0.00,    0.00),
(94105, '2021-08-29',    63.83,    65.3,    0.00,    0.00),
(70726, '2021-09-01',    81.17,    81.5,    0.00,    3.60),
(70726, '2021-08-31',    80.29,    83.2,    0.71,    4.70),
(70726, '2021-08-29',    74.35,    81.9,    0.09,    3.70);
```

Return all the rows where the temperature (***temp*** after the previous question) is higher than the historical average temperature (***history_avg_temperature***) ordered by ***zip*** and ***date*** in ascending order.

- Write only one SQL query (0.2 pt).
- Return all qualifying rows (0.5 pt).

- Make sure that the constraints meet the criteria (0.8 pt).
- Order of the table (0.5 pt).

Ex.

```
 zip  |    date    | temp | history_avg_temperature | precipitation | history_avg_precipitation
-------+------------+------+-------------------------+---------------+--------------------------
94105 | 2021-09-01 |   65 |                   62.53 |             0 |                         0
```

Q2.

For the following tables called "customer" and "transaction", write a SQL query to perform the given search.

customer

```
 column_name |     data_type
-------------+-------------------
 id          | integer
 type        | character varying
```

transaction

```
 column_name | data_type
-------------+-----------
 customer_id | integer
 date        | date
 amt         | numeric
```

Example data tables are given below. However, your code should be generic and work with any valid data.

customer

```
 id |        type
----+--------------------
  1 | business
  2 | business_executive
  3 | personal
  4 | franchise_business
  5 | personal
```

transaction

```
 customer_id |    date    | amt
-------------+------------+-----
           1 | 2021-09-29 |  10
           1 | 2021-09-30 |  20
           2 | 2021-09-30 |  30
           2 | 2021-10-01 |  30
           3 | 2021-10-01 |  30
           4 | 2021-10-01 |  24
```

*You can create the shipping table using the following queries.

```
CREATE TABLE customer
(id INTEGER,
type VARCHAR,
PRIMARY KEY (id));

INSERT INTO customer
```

```
VALUES
(1,'business'),
(2, 'business_executive'),
(3, 'personal'),
(4,'franchise_business'),
(5, 'personal');


CREATE TABLE transaction
(customer_id INTEGER,
 date DATE,
 amt NUMERIC,
 PRIMARY KEY (customer_id, date),
 FOREIGN KEY (customer_id) REFERENCES customer(id)
);


INSERT INTO transaction
VALUES
(1, '2021-09-29', 10),
(1, '2021-09-30', 20),
(2, '2021-09-30', 30),
(2, '2021-10-1', 30),
(3, '2021-10-1', 30),
(4, '2021-10-1', 24);
```

Return ***date***, ***customer_id*** and ***amt*** from the ***transaction*** table for ***customers*** whose ***type*** includes the substring 'business' ordered by ***date*** and ***customer_id*** (ascending).

- Content of the table. (2.5 pt)

- Order of the table. (0.5 pt)

Ex.

```
    date    | customer_id | amt
------------+-------------+-----
 2021-09-29 |           1 |  10
 2021-09-30 |           1 |  20
 2021-09-30 |           2 |  30
 2021-10-01 |           2 |  30
 2021-10-01 |           4 |  24
```

Q3.

For the following tables called "customer" and "transaction", write a SQL query to perform the given search.

customer

```
 column_name |     data_type
-------------+-------------------
 id          | integer
 type        | character varying
```

transaction

```
 column_name | data_type
```

```
-------------+-----------
 customer_id | integer
 date        | date
 amt         | numeric
```

Example data tables are given below. However, your code should be generic and work with any valid data.

customer

```
 id |        type
----+--------------------
  1 | business
  2 | business_executive
  3 | personal
  4 | franchise_business
  5 | personal
```

transaction

```
 customer_id |    date    | amt
-------------+------------+-----
           1 | 2021-09-29 |  10
           1 | 2021-09-30 |  20
           2 | 2021-09-30 |  30
           2 | 2021-10-01 |  30
           3 | 2021-10-01 |  30
           4 | 2021-10-01 |  24
```

*You can create the shipping table using the following queries.

```
CREATE TABLE customer
(id INTEGER,
type VARCHAR,
PRIMARY KEY (id));

INSERT INTO customer
VALUES
(1,'business'),
(2, 'business_executive'),
(3, 'personal'),
(4,'franchise_business'),
(5, 'personal');

CREATE TABLE transaction
(customer_id INTEGER,
 date DATE,
 amt NUMERIC,
 PRIMARY KEY (customer_id, date),
 FOREIGN KEY (customer_id) REFERENCES customer(id)
);

INSERT INTO transaction
VALUES
(1, '2021-09-29', 10),
(1, '2021-09-30', 20),
(2, '2021-09-30', 30),
(2, '2021-10-1', 30),
(3, '2021-10-1', 30),
```

```
(4, '2021-10-1', 24);
```

Return the **id** and average transaction ***amt*** (as ***avg_amt*** and as an integer) for all customers who have less than 2 transaction records. Output should be ordered by ***avg_amt*** in descending order.

If it does not have any transaction record its average should appear as 0.

- Return all the customer ids that satisfy the given conditions. (3 pt)
- avg_amt should be calculated per customer id. (1.5 pt)
- avg_amt should be converted to an integer. (1 pt)
- Order of the table. (0.5 pt)

Ex.

```
 id | avg_amt
----+---------
  3 |      30
  4 |      24
  5 |       0
```

Q4.

For the following tables called "instructor" and "class", write a SQL query to perform the given search.

instructor

```
 column_name |     data_type
-------------+-------------------
 id          | numeric
 name        | character varying
```

class

```
 column_name | data_type
-------------+-----------
 info        | json
```

Example data tables are given below. However, your code should be generic and work with any valid data.

instructor

```
 id |       name
----+------------------
  1 | Terence Parr
  2 | Yannet Interian
  3 | Diane Woodbridge
  4 | Shan Wang
  5 | Michael Ruddy
(5 rows)
```

class

```
                       info
-------------------------------------------------
 {
         "name" : "Relational Databases",
         "class_id" : "MSDS691",
         "instructor_id" : 3,
         "start_date" : "2021-08-23"
}
 {
         "name" : "Time Series Analysis",
```

```
        "class_id" : "MSDS604",
        "instructor_id" : 4,
        "start_date" : "2021-10-18"
}
 {
        "name" : "EDA and Viosualization",
        "class_id" : "MSDS593",
        "instructor_id" : 4,
        "start_date" : "2021-07-05"
}
 {
        "name" : "Communications for Analytics",
        "class_id" : "MSDS610",
        "instructor_id" : 5,
        "start_date" : "2021-08-23"
}
 {
        "name" : "Intro to Machine Learning",
        "class_id" : "MSDS621",
        "instructor_id" : 1,
        "start_date" : "2021-10-18"
}
 {
        "name" : "Machine Learning Laboratory",
        "class_id" : "MSDS699",
        "instructor_id" : 3,
        "start_date" : "2021-10-18"
}
(6 rows)
```

*You can create the tables using the following queries.

```
CREATE TABLE instructor
(
    id NUMERIC,
    name VARCHAR,
    PRIMARY KEY (id)
);

INSERT INTO instructor VALUES
(1, 'Terence Parr'),
(2, 'Yannet Interian'),
(3, 'Diane Woodbridge'),
(4, 'Shan Wang'),
(5, 'Michael Ruddy');


CREATE TABLE class
(
    info json
);

INSERT INTO class VALUES
(
    '{
    "name" : "Relational Databases",
    "class_id" : "MSDS691",
```

```
    "instructor_id" : 3,
    "start_date" : "2021-08-23"
    }'
),
(
    '{
    "name" : "Time Series Analysis",
    "class_id" : "MSDS604",
    "instructor_id" : 4,
    "start_date" : "2021-10-18"
    }'
),
(
    '{
    "name" : "EDA and Viosualization",
    "class_id" : "MSDS593",
    "instructor_id" : 4,
    "start_date" : "2021-07-05"
    }'
),
(
    '{
    "name" : "Communications for Analytics",
    "class_id" : "MSDS610",
    "instructor_id" : 5,
    "start_date" : "2021-08-23"
    }'
),
(
```

```
    '{
    "name" : "Intro to Machine Learning",
    "class_id" : "MSDS621",
    "instructor_id" : 1,
    "start_date" : "2021-10-18"
    }'
),
(
    '{
    "name" : "Machine Learning Laboratory",
    "class_id" : "MSDS699",
    "instructor_id" : 3,
    "start_date" : "2021-10-18"
    }'
);
```

Return ***instructor_id*** **(integer)**, ***date*** **(date)**, ***name*** **(text),** and the corresponding instructor's next class starting ***date*** **(name : *next_class_date,* type : date)** and its ***name*** **(name: *next_class_name*, type: text)** from the class table ordered by ***instructor_id*** and ***date*** (ascending).

Feel free to use the ***class_view*** in Question 1. (Optional)

- Make sure that data type and column name follow the given criteria (1 pt).
- Content of the table, including the order (2 pt).

Ex.

```
 instructor_id |    date    |             name             | next_class_date |
next_class_name
```

```
---------------+------------+--------------------------------+-----------------+-
---------------------------

             1 | 2021-10-18 | Intro to Machine Learning      |                 |

             3 | 2021-08-23 | Relational Databases           | 2021-10-18      |
Machine Learning Laboratory

             3 | 2021-10-18 | Machine Learning Laboratory    |                 |

             4 | 2021-07-05 | EDA and Viosualization         | 2021-10-18      |
Time Series Analysis

             4 | 2021-10-18 | Time Series Analysis           |                 |

             5 | 2021-08-23 | Communications for Analytics   |                 |

(6 rows)
```

Q5.

For the following tables called "instructor" and "class", write a SQL query to perform the given search.

instructor

```
 column_name |     data_type
-------------+-------------------
 id          | numeric
 name        | character varying
```

class

```
 column_name | data_type
-------------+-----------
 info        | json
```

Example data tables are given below. However, your code should be generic and work with any valid data.

instructor

```
 id |       name
----+------------------
  1 | Terence Parr
  2 | Yannet Interian
  3 | Diane Woodbridge
  4 | Shan Wang
  5 | Michael Ruddy
(5 rows)
```

```
class
                        info
--------------------------------------------------
 {
          "name" : "Relational Databases",
          "class_id" : "MSDS691",
          "instructor_id" : 3,
          "start_date" : "2021-08-23"
}
 {
          "name" : "Time Series Analysis",
          "class_id" : "MSDS604",
          "instructor_id" : 4,
          "start_date" : "2021-10-18"
}
 {
          "name" : "EDA and Viosualization",
          "class_id" : "MSDS593",
          "instructor_id" : 4,
          "start_date" : "2021-07-05"
}
 {
          "name" : "Communications for Analytics",
          "class_id" : "MSDS610",
          "instructor_id" : 5,
          "start_date" : "2021-08-23"
}
 {
          "name" : "Intro to Machine Learning",
```

```
        "class_id" : "MSDS621",
        "instructor_id" : 1,
        "start_date" : "2021-10-18"
}
 {
        "name" : "Machine Learning Laboratory",
        "class_id" : "MSDS699",
        "instructor_id" : 3,
        "start_date" : "2021-10-18"
}
(6 rows)
```


*You can create the tables using the following queries.

```
CREATE TABLE instructor
(
    id NUMERIC,
    name VARCHAR,
    PRIMARY KEY (id)
);

INSERT INTO instructor VALUES
(1, 'Terence Parr'),
(2, 'Yannet Interian'),
(3, 'Diane Woodbridge'),
(4, 'Shan Wang'),
(5, 'Michael Ruddy');
```

```
CREATE TABLE class
(
    info json
);

INSERT INTO class VALUES
(
    '{
    "name" : "Relational Databases",
    "class_id" : "MSDS691",
    "instructor_id" : 3,
    "start_date" : "2021-08-23"
    }'
),
(
    '{
    "name" : "Time Series Analysis",
    "class_id" : "MSDS604",
    "instructor_id" : 4,
    "start_date" : "2021-10-18"
    }'
),
(
    '{
    "name" : "EDA and Viosualization",
    "class_id" : "MSDS593",
    "instructor_id" : 4,
    "start_date" : "2021-07-05"
```

```
    }'
),
(
    '{
    "name" : "Communications for Analytics",
    "class_id" : "MSDS610",
    "instructor_id" : 5,
    "start_date" : "2021-08-23"
    }'
),
(
    '{
    "name" : "Intro to Machine Learning",
    "class_id" : "MSDS621",
    "instructor_id" : 1,
    "start_date" : "2021-10-18"
    }'
),
(
    '{
    "name" : "Machine Learning Laboratory",
    "class_id" : "MSDS699",
    "instructor_id" : 3,
    "start_date" : "2021-10-18"
    }'
);
```

Create a function called ***class_instructor*** which takes ***class_name*** and return the corresponding instructor's ***id*** **(integer)** and ***name*** **(varchar)**

Feel free to use the ***class_view*** in Question 1. (Optional)

- The function's name (0.25 pt).
- The syntax for creating function (2 pt).
- Function's input/output and logic(2.75 pt).

Ex. The following shows the example query and output for the created view.

```
SELECT * FROM class_instructor('Relational Databases');

 id |       name

----+------------------

  3 | Diane Woodbridge
```

MSDS 621: Introduction to Machine Learning Questions

Q1. You are given a random forest trained on a data set in df with validation log lossscore s. What is the most likely effect on s if we add two noise columnsinto df, retrain the forest, and recompute the validation log lossscore?

Q2. Random forests use what two primary techniques to de-correlate trees (increase tree independence)? (My answer has 11 words, though with the right keywords it could probably be done in three or four.)

Q3. Why do training metrics improve so fast initially as we add more trees to a random forest? (My answer has 12 words.)

Q4. With a very large number of trees in a random forest, the model's accuracy eventually starts to drift and the validation error will start to climb as we add more trees.

Q5. Imagine training a random forest regressor with max features (number of candidate split variables) equal to p, the number of variables, and getting a very low training error. If you reduce max features to int(p*0.65), would you expect the training error to increase or decrease?

MSDS 699: Machine Learning Laboratory Questions

Q1. Compute 2-fold regularized mean encoding of the feature “state”. Assume the first 4 rows are the first fold.

state	y	city_mean_enc
New York	0	
New York	1	
California	0	
New York	1	
California	0	
New York	1	
California	0	
New York	0	

Q2. Suppose we use Lasso to fit a model. What is the effect of increasing λ on bias and variance?

(a) Increases bias, increases variance

(b) Increases bias, decreases variance

(c) Decreases bias, increases variance

(d) Decreases bias, decreases variance

(e) Not enough information to tell

Q3. You are doing two-fold cross validation for logistic regression. Model 1 is a logistic regression model trained with the first 4 observations. Model 2 is a logistic regression model trained with the last 4 observations. The following table gives you the actual label and the predictions of both models on all the data. (a) What is the cross-validated prediction? (b) Compute cross-validation AUC. (Show me your steps)

y	$\hat{y}$ (Model1)	$\hat{y}$ (Model2)
0	0.1	0.25
0	0.5	0.45
1	0.9	0.7
1	0.8	0.95
1	0.4	0.3
0	0.3	0.1
0	0.2	0.35
1	0.9	0.8

Q4. You will be using Logistic regression with L1 regularization to predict income > 50k as a binary classification task.

(a) Download the data:

https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data

and

https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test

(b) Encode variables for logistic regression. Consider age, fnlwgt, capital-gain, capital-loss, hours-per-week as continuous variables and anything else as categorical. Ignore the column 'education-num'.

(c) Use cross-validation k=3 to select the appropriate level of regularization for a logistic regression model with L1 penalty. Consider the following values of the penalization constant [0.01, 0.1, 1, 10, 100]. Use roc auc score as your metric.

(d) What is the roc auc score on the test set? Retrain on the whole training data using the best penalty and compute roc auc score on the test set.

(e) How many variables is logistic regression setting to 0?

MSDS 694: Distributed Computing Questions

Q1.

Question 4 0.5 pts

What is the output of the following line, when rdd is a resilient distributed dataset?

```
type(rdd.count().count())
```

- ○ Error
- ○ int
- ○ str
- ○ list

Q2.

Question 10 0.5 pts

What is the output of the following code?

```
sc = pyspark.SparkContext()
rdd = sc.parallelize([101, 103, 102, 104])
rdd.sortBy(lambda x: x).first()
```

- ○ 101
- ○ 103
- ○ 102
- ○ 104

Q3.

Question 12 0.5 pts

What is the output of the following code?

```
sc = pyspark.SparkContext()
rdd = sc.parallelize([ ])
rdd.reduce(lambda x, y: x + y)
```

- ○ Error
- ○ 0
- ○ [ ]
- ○ [0]

Q4.

Question 2 0.5 pts

Which Spark component orchestrates resources and monitors an application?

- ○ client
- ○ driver
- ○ executor
- ○ livy

Q5.

Question 11 0.75 pts

How many shuffles happen for the RDD ('final') in the last line of the following code?

```
sc = pyspark.SparkContext()
lines = sc.parallelize([('1', 'a'), ('1', 'b'), ('2', 'a'), ('2', 'b')], 4)
final = lines.map(lambda x : (int(x[0]), x[1]))\
             .sortByKey()\
             .coalesce(5)\
             .reduceByKey(lambda x, y : x+y)
```

- ○ 1
- ○ 2
- ○ 3
- ○ 4

MSDS 630: Advanced Machine Learning Questions

Q1. You are give some data from a binary classification problem where your torch tensors have the following shapes: X.shape is [264, 23] and y.shape is [264]

(a) Write a PyTorch model for this problem. You can use nn.sequential() and nn.Linear()

(b) From the following loss functions with you would you use together with your model:

F.binary_cross_entropy, F.binary_cross_entropy_with_logits, F.cross_entropy, F.mse_loss

(c) Let your predictions be y_hat = model(x). What is the shape of y_hat?

Q2. A local bookstore has a very loyal user base and a database of rating (1-5). They send monthly emails with personalized book recommendations. Their recommendations are based on a version of matrix factorization in which the user embedding matrix U has been already computed (with some previous data). They do monthly updates of the item matrix V . In this setting consider the following questions:

(a) How many parameters get updated in gradient descent each month?

(b) Consider matrices U, V1, V2 and the following test set. Test set: y0,0 = 1, y4,3 = 4 Which matrix V1 or V2 fits better the test data? (Show me your work)

$$U = \begin{bmatrix} 0.2 & 2 & 0 \\ 1.6 & 1 & 2 \\ 2.6 & -0.6 & 1.5 \\ 0.9 & 0.7 & 0.2 \\ 2. & 0 & 1 \\ -1.9 & 0.5 & 2.3 \\ 0.8 & -0.9 & 0.9 \end{bmatrix}, V_1 = \begin{bmatrix} 0 & 0.5 & 1 \\ 3.6 & 0 & 0.1 \\ 0 & 4 & 0.9 \\ -2 & 0.1 & 3 \end{bmatrix}, V_2 = \begin{bmatrix} 1 & 1.5 & 1.0 \\ 3 & 0 & 0.1 \\ 0 & 1 & 0.9 \\ 2.5 & 1 & -1 \end{bmatrix}$$

Q3. Consider the following dataset. You will be applying gradient boosting for MSE. Let each of the trees be a stump.

x	y	f_0	$y - f_0$	T_1	f_1
7	-3				
2	3				
-1	6				

(a) Compute f0, T1, f1 on the table above.

(b) What is the value of f1(−2)?

(c) What is the training error of f1?

Q4.

Given a training set $x^{(1)}, \ldots, x^{(5)}$, the table below shows the weights w_i that Adaboost assigns to $x^{(i)}$ for 3 iterations. Note that weights are normalized at each iteration.

(a) Which point(s) were misclassified by the first base classifier $T_1(x)$? What are the values for err_1 and α_1?

(b) Which point(s) were misclassified by the second base classifier $T_2(x)$? What are the values for err_2 and α_2?

(c) Given that $y^{(1)} = 1$, what is the expression for $F(x^{(1)})$ after two iterations of Adaboost. What is the value for $\hat{y}^{(1)}$ after two iterations?

m	w_1	w_2	w_3	w_4	w_5	err	α
1	1/5	1/5	1/5	1/5	1/5		
2	1/4	1/6	1/6	1/6	1/4		
3	3/16	1/4	1/4	1/8	3/16		

MSDS 689: Data Structures and Algorithms Questions

Q1. Imagine that we have a list of numbers in the range 1000 to 1004 (inclusive) and we'd would like to use pigeonhole sort to sort the numbers. We can use 5 pigeonholes labeled 0,1,2,3,4 to do this. Choose from the following what the pigeonholes would look like after adding [1004,1003,1001,1001,1000].

Q2. Consider bubble sort on array A. What is the worst case condition of the elements in A that causes time complexity?

Q3. Consider a highly-unbalanced classification data set and using accuracy (number correct / n) as a metric and a random forest for modeling. True/false: A near-zero permutation importance value indicates is not important for any records.

Q4. If spearman‚Äôs R coefficient between feature and target is high magnitude, dropping always strongly decreases model training accuracy.

MSDS 697: Distributed Data Systems Questions

Q1.

Question 2 0.5 pts

What is the database type of MongoDB?

- ○ Graph Database
- ○ Document Database
- ○ Relational Database
- ○ Column-family database (Columnar database)

Q2.

Question 5 0.5 pts

For MongoDB, which of the following is false?

- ○ A strictly defined schema is required.
- ○ It routes a user request to the correct machines.
- ○ It takes care of balancing data.
- ○ It stores data in a JSON-like document format (BSON).

Q3.

Question 12 0.5 pts

```
db.numbers.find() // 1)
db.numbers.find({"number.type":"even"},{"number":{$elemMatch:{"value":1.0}}}) // 2)
```

The first line in the above code, "db.numbers.find()" returned the following output.

```
{"number" : [{"value" : 1.0,"type" : "odd"}, {"value" : 2.0, "type" : "even"}, {"value" :
3.0, "type" : "odd"}, {"value" : 4.0, "type" : "even"}] }
```

Choose the output of line 2).

- ○ { "number" : [{"value" : 1.0, "type" : "odd"}]}
- ○ { "number" : [{"value" : 1.0}]}
- ○ { "number" : [{"value" : 1.0, "type" : "odd"}, {"value" : 3.0, "type" : "odd"}]}
- ○ None of the above

Q4.

Question 3 0.5 pts

"**db.usf_locations.find()**" returned the following documents.

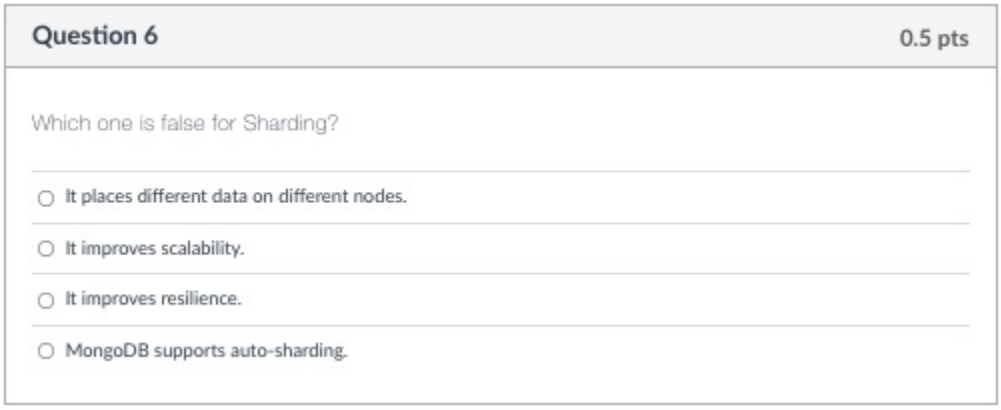
```
{
    "campus_name" : "USF Downtown Campus",
    "department" : [
        {
            "name" : "Master of Science In Data Science",
            "building" : "101 Howard"
        },
        {
            "name" : "Bachelor of Science in Management",
            "building" : "101 Howard"
        },
        {
            "name" : "Bachelor of Science in Management",
            "building" : "101 Howard"
        }
    ],
    "address" : {
        "street" : "101 Howard St",
        "city" : "San Francisco",
        "state" : "CA",
        "zip" : NumberInt(94105)
    }
}
{
    "campus_name" : "Main Campus USF",
    "department" : [
        {
            "name" : "School of Nursing and Health Professions",
            "building" : "Cowell Hall"
        },
        {
            "name" : "Fromm Residence Hall",
            "building" : "Fromm Hall"
        },
        {
            "name" : "Computer Sience Department",
            "building" : "Harney SCIenCe Center"
        },
        {
            "name" : "Environmental Science Department",
            "building" : "Harney Science Center"
        },
        {
            "name" : "Arts and Social Sciences Departments",
            "building" : "Kalmanovitz Hall"
        }
    ],
    "address" : {
        "street" : "2130 Fulton Street",
        "city" : "San Francisco",
        "state" : "CA",
        "zip" : NumberInt(94117)
    }
}
```

How many documents will be returned by the following query?

```
db.usf_locations.aggregate({$match:{"campus_name" : "USF Downtown Campus"}},
                           {$unwind:"$department"})
```

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5

Q5.


Question 6
0.5 pts
Which one is false for Sharding?
It places different data on different nodes.
It improves scalability.
It improves resilience.
MongoDB supports auto-sharding.

Q6.


Question 7
0.5 pts
Which one is false for Primary-Secondary Replication?
It provides good read resilience.
It works well with intensive writes.
It provides redundancy and increases data availability.
It provides fault tolerance against the loss of a single database server.