Electronics & ICT Academy
Indian Institute of Technology, Guwahati

Post Graduate Certification in
Data Science

Clinical Research in Healthcare

Industry Grade Project – I: PG Certification in Data Science by E&ICT Academy IITG

Table of Contents

Aim of the Project

The aim of this project is to build Machine Learning models to predict health conditions of patients based on Electronic Health Records

Background

When a person gets ill and visits a hospital, there are two common questions that are posed in front of the person:

1. What has happened to me?
2. How does my future look like?

While the first question refers to the diagnosis of the current situation to know the illness, the second is about predicting future medical risks. Exploring the answer to the first question is easy due to the availability of a broad spectrum of diagnostic tools today. The technology is much less advanced today to provide answers to the second question. Traditionally, this question about medical risk is answered only by experienced clinicians.

The challenge that the Healthcare industry is facing today, is that the experienced specialists are highly expensive and have limited availability. This is where modern Electronic Health Records (EHRs) bring a ray of hope as they promise to offer a fast and cheap alternative.

Typically, an EHR contains the history about the patient, hospital encounters, diagnoses and interventions, lab tests, and clinical narratives. The full adoption of EHRs has led to intensified research in building predictive models from this rich data source in the past few years.

Data Science facilitates building of such predictive models on patient-level temporal healthcare data. These models must address four open challenges:

- Long-term dependencies on healthcare. For instance, the onset of diabetes at middle age remains a risk factor for the rest of the life
- Representation of admission: The admission data is complex and consists of information related to diagnoses, interventions, and other such details.
- Medical records varying significantly in length are inherently episodic and irregular
- Medical records are a blend of the course of illness, intervening, and the developmental processes

In addition to addressing these four challenges, a predictive system is expected to comprehend medical records, determine present illness states, and predict future outcomes based on the data at hand.

Additional Reading: Over the past few years, the Machine Learning community has witnessed widespread advances in the field of Deep Learning for research on EHR data. This Arxiv Paper known as DeepEHR, surveys the current research on applying the cutting edge Deep Learning technology to clinical tasks based on EHR data. Deep Learning has offered the researchers new lens to explore unparalled techniques and frameworks that can be applied to several types of clinical applications including information extraction, representation learning, outcome prediction, phenotyping, and de-identification. In this survey, they also identify several limitations of current research involving topics such as model interpretability, data

heterogeneity, and lack of universal benchmarks. Finally, they conclude by summarizing the status of research in the field and identifying avenues of future deep EHR research. Undoubtedly, this is a great read and should be of great help to those interested in comprehending the landscape of EHR research and its applications.

In our project, we will not be using Deep Learning techniques to build models, although you are free to explore these on your own.

Use Cases

The models shall mainly support the following use cases:

1. Predicting the future health outcomes of the patients: Build a model to use a patient's health timeline to predict the future conditions that the patient might be diagnosed with. This model can be used to predict any condition that might appear in the future. You can focus on lifelong non-communicable conditions like diabetes, hypertension, obesity, et al.

2. Predicting the future costs incurred by the patients: Build a model to predict the costs incurred by the patients in the next 12 months. Your model should be able to distinguish between the costs for medications and encounters. Think of it as a solution for an insurance company that wants to assess how much a patient would be spending on visiting a hospital for checkups and treatment and what they would be spending on medications.

3. Clustering of patients based on their health conditions: Build an unsupervised clustering model to group the patients into multiple clusters. Assess these clusters to get a sense of

which clusters (and with what conditions) patients fall into. This exercise will be useful in identifying various co-morbidities that co-occur in patients.

Process Flow

As mentioned in the above use cases, the final objective of this project is to build Machine Learning models that can solve the healthcare problems. To achieve that objective, it is essential for you to follow the following three phases of Data Science:

1. Data Exploration:

Explore the dataset provided to build a thorough understanding of the terms and data points available in an EHR. You are supposed to do a detailed exploratory data analysis for the given dataset.

Analyze the columns in the files tagged in the dataset and build an Entity Relationship Diagram (ERD) for the dataset provided. Please use the schema provided in the Data Files section to come up with the relationships between these datasets. The Wiki Link will give you more details about how an ERD Diagram should look like.

2. Feature Engineering:

The focus of the data exploration exercise was to get you familiar with the columns in the data and their values from the perspective of building training data sets and features. In the second phase of feature engineering, build appropriate features using the available datasets for training of each of the three models.

- For the model where the objective is to predict health outcomes, you should be able to identify the conditions staying for life once they are diagnosed (i.e., do not have a STOP date) and then you should be able to easily find encounters before and after that particular encounter to build features from

- For the model where the objective is to predict future costs, find the costs at a patient level for encounters, medications, immunisations, etc., from the various datasets provided and aggregate them to make target variables for your regression models

- Join various datasets together and build patient level features out of the merged data frames like
    - Conditions that could be joined with encounters to find out the conditions that were diagnosed during encounters
    - Similarly, medications and observations datasets, when joined with the encounters dataset, would reveal details about which medications were prescribed and which observations were found/observed during encounters

3. Predictive Models:

Once the features have been built and training and test datasets are created as described in previous phases, build models to predict outcomes as described in the Use Cases.

Tasks to be performed

Following are the guidelines to perform various tasks related each of the three phases:

1. Data Exploration:

You would draw the diagram at your convenience, but we would like you to determine the relationships between the following tables:

- Encounters and Conditions
- Patients and Encounters
- Encounters and Medications
- Encounters and Observations

Following are some additional guiding points based on which various datasets can be explored:

- Analyse the unique values and most frequently-occurring categories in the Categorical Columns.

- You can analyze the encounters dataset
    - Why are patients visiting hospitals? What are the different reasons to visit hospitals?
    - What is the duration of various encounters?
    - How many encounters do patients have?

- You can analyze the conditions dataset
    - What the various conditions that patients have been diagnosed with?
    - What is the duration of various conditions?
    - Do some conditions occur more frequently than others?

- You can analyze the medications dataset
    - What are the different medications prescribed to the patients?
    - How long are the patients on these various medications?
    - How much do patients usually spend on medications?

- You can analyze the observations dataset
    - What are the various observations for patients? If they are in the form of numeral or text?
    - What is the distribution of values for various observations?

- These datasets can be looked at in many permutation and combinations to reveal interesting insights about patients diagnosed with various conditions.
    - Are the number of encounters and the types of encounters different for patients who have been diagnosed with certain conditions?
    - Do encounters increase after the onset of certain conditions?
    - How are observations for patients with certain conditions different from those who do not exhibit those conditions?
    - Are the number of medications and the types of medications different for patients who have been diagnosed with certain conditions?
    - Can you identify correlations between some observations and conditions?

- These steps will help you when you are building features for various predictive models.

2. Feature Engineering:

- In a healthcare dataset, medications, diagnosed conditions, observed medical parameters, immunisation and more such parameters play a critical role in determining the present and the future of a patient's health. **Joining more datasets together would help you cover a patient's health more comprehensively and offer you a better understanding in terms of what medications they were on, what observations were measured during their encounters and what conditions were they diagnosed, etc. This should be your goal while developing features for your models.**

- Remember that you should always have the problem statement that you are trying to solve in your head when you come up with features

These are guidelines for constructing training, testing and validation dataset from the EHR data provided for the different models to be created as a part of this project.

- Use the encounters, conditions, medications, observations, allergies, careplans and immunisations data to construct a timeline for each patient
- The timeline should be broken chronologically into before and after periods, where the before period is used as input to your models and the after period is used as targets
- Analyze the dataset and remove the patients that have very few data-points for encounters and who do not have longitudinal data that suits your models
- Since there may be a lot of unique values for encounters, medications and observations, you can remove the ones that have a very high and very low frequency as they might not be very important predictors in your models
- There are a lot of feature engineering techniques that you can use to turn categorical columns into features either by using dummy variables or bucketing the categories or by embedding them using TF-IDF or other complex techniques
- Do not use date columns as is; you can instead use them to create other features. For example, you can group the encounters into 6 months or a single year or during the entire before-period and create features for these
- It does not make sense to use patient id as a predictor but you can use the patient's demographics

- Before removing NAs from data, do check if there are columns which have too many NaN. See whether you need to impute those values or need to drop that column altogether before you start removing NA observations from the entire data
- These are just ideas. Feel free to make any other features from these
- Once you have prepared your dataset and features, the training, validation and test sets should be created by dividing the dataset into three parts namely 2/3, 1/6, 1/6 data points, respectively. Breaking your data into these parts will help when you are testing your model's performance
- You could also use n-fold cross-validation to build robust models which would require you to break your dataset into training and validation data. You can cross-validate on training data and later assess the trained and cross-validated model on the validation data to get a sense of the model's performance
- If you are modifying the existing features or even creating new features on your training data, you will have to do that for test data, too. This is needed so that the model which was built using the training data can be used for the test data to make predictions
- Once you have extracted the relevant features for your experiment, you can try grid search or random search available in sklearn to tune hyperparameters in the model that you choose

It is a large dataset, and it might take a lot of time to run.

3. Predictive Models:

Model 1 – Health Outcomes:

Build a model to use a patient's health timeline to predict the future conditions that the patient might be diagnosed with. This model can be used to predict any condition that might appear in the future. You can focus on lifelong non-communicable conditions like diabetes, hypertension, obesity, et al.

- Split the temporal patient timeline into before and after periods. You could use multiple permutations and combinations for the same. Your before period could range from 2 years to anywhere up to 10 years and your after period could be the subsequent 6 months to the rest of the patient's timeline in the EHR dataset
- To ensure that you are able to detect the condition, you should try to create the before-period for patients before they were diagnosed with the conditions that you are trying to predict
- Combine at least two or more data points available in encounters, conditions, medications, observations, immunizations, allergies and care plans datasets to draw a complete picture of the patient's health
- Targets for this exercise would be the conditions diagnosed in the after-period
    - Try predicting the diagnosed condition for the next 6 months right after the before period ends
- You can also build a model to predict the conditions in the rest of the patient's lifetime

Model 2 – Future Costs:

Build a model to predict the costs incurred by the patients in the next 12 months. Your model should be able to distinguish between the costs for medications and encounters. Think of it as a solution for an insurance company that wants to assess how much a patient would be spending

on visiting a hospital for checkups and treatment and what they would be spending on medications.

- Feature building and dataset would be similar to the one you used for Model 1, i.e., construct a patient's timeline and divide it appropriately into a before and after period
- In this case, your after period would be 12 months subsequent to the ending of the before period
- Use the costs for medications and encounters as two target variables for your regression models
- You could also bucket the costs into groups and train a multi-class classification model on the cost-groups or cost-buckets

Model 3 – Clustering on health conditions:

Build an unsupervised clustering model to group the patients into multiple clusters. Assess these clusters to get a sense of which clusters (and with what conditions) patients fall into. This exercise will be useful in identifying various co-morbidities that co-occur in patients.

- Identify a cluster with the maximum number of diabetes patients
- What are the other common conditions that the patients in this cluster are diagnosed with?
- What are the common medications that the patients in this cluster are prescribed? Are these different from the ones that are prescribed to diabetes patients in general?

- How are the observations for the patients in this cluster different from those for diabetes patients in general?

Dataset Description

The dataset comprises multiple CSV files that correspond to the data available in EHR and follow the FHIR standards

Complete List of Files

File	Description
allergies.csv	Patient allergy data.
careplans.csv	Patient care plan data, including goals.
conditions.csv	Patient conditions or diagnoses.
encounters.csv	Patient encounter data.
imaging_studies.csv	Patient imaging metadata.
immunizations.csv	Patient immunization data.
medications.csv	Patient medication data.
observations.csv	Patient observations including vital signs and lab reports.

organizations.csv	Provider organizations, including hospitals.
patients.csv	Patient demographic data.
payer_transitions.csv	Payer Transition data (i.e., changes in health insurance).
payers.csv	Payer organization data.
procedures.csv	Patient procedure data, including surgeries.
providers.csv	Clinicians that provide patient care.
supplies.csv	Supplies used in the provision of care.

Data Dictionary information for each CSV table follows below.

Allergies

Column Name	Data Type	Description
Start	Date (YYYY-MM-DD)	The date the allergy was diagnosed.

Stop	Date (YYYY-MM-DD)	The date the allergy ended, if applicable.
Patient	UUID	Foreign key to the patient.
Encounter	UUID	Foreign key to the Encounter when the allergy was diagnosed.
Code	String	Allergy code from SNOMED-CT
Description	String	Description of the allergy

CarePlans

Column Name	Data Type	Description
Id	UUID	Primary key. Unique identifier of the care plan.
Start	Date (YYYY-MM-DD)	The date the care plan was initiated.
Stop	Date (YYYY-MM-DD)	The date the care plan ended, if applicable.
Patient	UUID	Foreign key to the patient.

Encounter	UUID	Foreign key to the encounter when the care plan was initiated.
Code	String	Code from SNOMED-CT
Description	String	Description of the care plan.
ReasonCode	String	Diagnosis code from SNOMED-CT that this care plan addresses.
ReasonDescription	String	Description of the reason code.

Conditions

Column Name	Data Type	Description
Start	Date (YYYY-MM-DD)	The date the condition was diagnosed.
Stop	Date (YYYY-MM-DD)	The date the condition resolved, if applicable.
Patient	UUID	Foreign key to the patient.

Encounter	UUID	Foreign key to the Encounter when the condition was diagnosed.
Code	String	Diagnosis code from SNOMED-CT
Description	String	Description of the condition.

Encounters

Column Name	Data Type	Description
Id	UUID	Primary key. unique identifier of the encounter.
Start	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the encounter started.
Stop	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the encounter concluded.
Patient	UUID	Foreign key to the patient.
Organization	UUID	Foreign key to the organization.

Provider	UUID	Foreign key to the provider.
Payer	UUID	Foreign key to the payer.
EncounterClass	String	The class of the encounter, such as the ambulatory, emergency, inpatient, wellness, or urgent care.
Code	String	Encounter code from SNOMED-CT
Description	String	Description of the type of encounter.
Base_Encounter_Cost	Numeric	The base cost of the encounter, not including any line item costs related to medications, immunizations, procedures, or other services.
Total_Claim_Cost	Numeric	The total cost of the encounter, including all line items.
Payer_Coverage	Numeric	The amount of cost covered by the payer.
ReasonCode	String	Diagnosis code from SNOMED-CT, only if this encounter targeted a specific condition.

ReasonDescription	String	Description of the reason code.

Imaging Studies

Column Name	Data Type	Description
Id	UUID	Primary key. Unique identifier of the imaging study.
Date	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the imaging study was conducted.
Patient	UUID	Foreign key to the patient.
Encounter	UUID	Foreign key to the encounter where the imaging study was conducted.
Body Site Code	String	A SNOMED Body Structures code describing what part of the body the images in the series were taken of.
Body Site Description	String	Description of the body site.
Modality Code	String	A DICOM-DCM code describing the method

		used to take the images.
Modality Description	String	Description of the modality.
SOP Code	String	A DICOM-SOP code describing the Subject-Object Pair (SOP) that constitutes the image.
SOP Description	String	Description of the SOP code.

Immunizations

Column Name	Data Type	Description
Date	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date the immunization was administered.
Patient	UUID	Foreign key to the patient.
Encounter	UUID	Foreign key to the encounter where the immunization was administered.
Code	String	Immunization code from CVX.

Description	String	Description of the immunization.
Cost	Numeric	The line item cost of the immunization.

Medications

Column Name	Data Type	Description
Start	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the medication was prescribed.
Stop	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the prescription ended, if applicable.
Patient	UUID	Foreign key to the patient.
Payer	UUID	Foreign key to the payer.
Encounter	UUID	Foreign key to the encounter where the medication was prescribed.
Code	String	Medication code from RxNorm.
Description	String	Description of the medication.

Base_Cost	Numeric	The line item cost of the medication.
Payer_Coverage	Numeric	The amount covered or reimbursed by the payer.
Dispenses	Numeric	The number of times the prescription was filled.
TotalCost	Numeric	The total cost of the prescription, including all dispenses.
ReasonCode	String	Diagnosis code from SNOMED-CT specifying why this medication was prescribed.
ReasonDescription	String	Description of the reason code.

Observations

Column Name	Data Type	Description
Date	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the observation was performed.

Patient	UUID	Foreign key to the patient.
Encounter	UUID	Foreign key to the encounter where the observation was performed.
Code	String	Observation or Lab code from LOINC
Description	String	Description of the observation or lab.
Value	String	The recorded value of the observation.
Units	String	The units of measure for the value.
Type	String	The datatype of value: text or numeric

Organizations

Column Name	Data Type	Description
Id	UUID	Primary key of the organization.
Name	String	Name of the organization.
Address	String	Organization's street address without commas or newlines.

City	String	Street address city.
State	String	Street address state abbreviation.
Zip	String	Street address zip, or postal code.
Lat	Numeric	Latitude of organization's address.
Lon	Numeric	Longitude of organization's address.
Phone	String	Organization's phone number.
Revenue	Numeric	The monetary revenue of the organization during the entire simulation.
Utilization	Numeric	The number of encounters performed by the organization.

Patients

Column Name	Data Type	Description
Id	UUID	Primary Key. Unique identifier of the patient.
BirthDate	Date (YYYY-	The date the patient was born.

	MM-DD)	
DeathDate	Date (YYYY-MM-DD)	The date the patient died.
SSN	String	Patient Social Security identifier.
Drivers	String	Patient Drivers License identifier.
Passport	String	Patient Passport identifier.
Prefix	String	Name prefix, such as Mr., Mrs., Dr., etc.
First	String	First name of the patient.
Last	String	Last, or surname of the patient.
Suffix	String	Name suffix, such as Ph.D., MD, JD, etc.
Maiden	String	Maiden name of the patient.
Marital	String	Marital Status. M is married, S is single. Currently no support for divorce (D) or widowing (W)

Race	String	Description of the patient's primary race.
Ethnicity	String	Description of the patient's primary ethnicity.
Gender	String	Gender. M is male, F is female.
BirthPlace	String	Name of the town where the patient was born.
Address	String	Patient's street address without commas or newlines.
City	String	Patient's address city.
County	String	Patient's address county.
State	String	Patient's address state.
Zip	String	Patient's zip code.
Lat	Numeric	Latitude of patient's address.
Lon	Numeric	Longitude of patient's address.

Healthcare_Expenses	true	
Healthcare_Coverage	true	

Payer Transitions

Column Name	Data Type	Description
Patient	UUID	Foreign key to the patient.
Start_Year	Date (YYYY)	The year the coverage started (inclusive).
End_Year	Date (YYYY)	The year the coverage ended (inclusive).
Payer	UUID	Foreign key to the payer.
Ownership	String	The owner of the insurance policy. Legal values: Guardian, Self, Spouse.

Payers

Column Name	Data Type	Description
Id	UUID	Primary key of the payer (e.g., Insurance).
Name	String	Name of the payer.
Address	String	Payer's street address without commas or newlines.
City	String	Street address city.
State_Headquartered	String	Street address state abbreviation.
Zip	String	Street address zip or postal code.
Phone	String	Payer's phone number.
Amount_Covered	Numeric	The monetary amount paid to organizations during the entire simulation.
Amount_Uncovered	Numeric	The monetary amount not paid to organizations during the entire simulation and covered out of pocket by patients.

Revenue	Numeric	The monetary revenue of the payer during the entire simulation.
Covered_Encounters	Numeric	The number of encounters paid for by this payer.
Uncovered_Encounters	Numeric	The number of encounters not paid for by this payer, and paid out of pocket by patients.
Covered_Medications	Numeric	The number of medications paid for by this payer.
Uncovered_Medications	Numeric	The number of medications not paid for by this payer, and paid out of pocket by patients.
Covered_Procedures	Numeric	The number of procedures paid for by this payer.
Uncovered_Procedures	Numeric	The number of procedures not paid for by this payer, and paid out of pocket by patients.
Covered_Immunizations	Numeric	The number of immunizations paid for by this payer.
Uncovered_Immunizations	Numeric	The number of Immunizations not paid for by this payer, and paid out of pocket by patients.
Unique_Customers	Numeric	The number of unique patients enrolled with this payer during the entire simulation.

QOLS_Avg	Numeric	
Member_Months	Numeric	The total number of months that patients were enrolled with the payer during the simulation and paid monthly premiums (if any).

Procedures

Column Name	Data Type	Description
Date	iso8601 UTC Date (yyyy-MM-dd'T'HH:mm'Z')	The date and time the procedure was performed.
Patient	UUID	Foreign key to the patient.
Encounter	UUID	Foreign key to the encounter where the procedure was performed.
Code	String	Procedure code from SNOMED-CT
Description	String	Description of the procedure.
Base_Cost	Numeric	The line item cost of the procedure.

ReasonCode	String	Diagnosis code from SNOMED-CT specifying why this procedure was performed.
ReasonDescription	String	Description of the reason code.

Providers

Column Name	Data Type	Description
Id	UUID	Primary key of the provider/clinician.
Organization	UUID	Foreign key to the organization that employs the provider.
Name	String	First and last name of the provider.
Gender	String	Gender. M is male, F is female.
Speciality	String	Provider specialty.
Address	String	Provider's street address without commas or newlines.
City	String	Street address city.

State	String	Street address state abbreviation.
Zip	String	Street address zip, or postal code.
Lat	Numeric	Latitude of provider's address.
Lon	Numeric	Longitude of provider's address.
Utilization	Numeric	The number of encounters performed by the provider.

How to Start with the Project?

1. Login to the Edureka's Big Data My Lab (Available under the course Big Data Storage and Analytics). Download the necessary datasets from the LMS and load them to the Big Data Cluster. Login to Jupyter and start building your notebook.

2. Import all the necessary Python packages. Numpy and Pandas for numerical processing, data importing, preprocessing etc. Matplotlib for plotting pose joints and showing images, sklearn for splitting datasets, PySpark for performing machine learning on Big Data, etc.

3. From here you can take over to the project and start building the machine learning model for analyzing EHR datasets.

How to submit your project?

- Share your project solution in an .ipynb file to support@edureka.co
- The .ipynb file should contain the details of each step in the markdown
- Add description of approach taken for each of the three phases
- Add answers to any subjective decisions made throughout the process in the markdown
- You can even upload your code into your github repository and share the link of your repository with us

Marks Allocation

1. Data Exploration [30 Marks]
2. Feature Engineering [40 Marks] (Marks based on number of datasets combined to build the features)
3. Predictive Models [10 x 3 Marks]
4. Detailed explanation of steps in the markdown is important to allocate marks