**SUPPLEMENTAL METHODS**

**Data Sources**

The AdventHealth big data platform is a Dedicated Cloudera Distribution Big Data cluster housed on site in Oracle Big Data Appliance (BDA). This BDA houses the IDEA platform, which has a Data Lake and an Enterprise Data Warehouse. The Data Lake is catalogued by source system and additional subject area specific conformed repositories are interconnected with Enterprise level dimensional data. This architecture contains near real time data (4 to 5 second latency) feeds from over 1,000 source data tables. A Kafka footprint is also in place to support near real time data streaming. Based on the various source systems the feed timing can be near real time, daily, weekly, monthly, quarterly or on demand. There is a wide variety of data sources feeding this repository but some of the larger footprints are near real time CERNER EHR data, Athena EHR Data Warehouse feed Version 17.1, Population Health Claims data, and Patient Experience. The Data Lake is not only capable of ingesting traditional data, but it also supports semi-structured and unstructured data, including FHIR data.

The COVID-19 data analytics effort is supported by the AdventHealth IDEA scalable distributed data platform. Briefly, raw data from the EHR systems are ingested into the Data Lake. No modeling is done at this stage, but only minimal data processing such as merging EHR data from all geographical domains and conforming date/time elements to a common local time-zone. This allows multiple use cases for research, providing an almost real-time understanding of the size and distribution of the COVID-19 patient cohort across all 53 AdventHealth hospitals. A shared curated data set specific to COVID-19 was created to centralize the collection effort of key subject areas to include patient, encounters, and clinical data.

The Clinical Classifications Software Refined (CCSR) is a publicly available taxonomy created and maintained by the Agency for Healthcare Research and Quality which aggregates International Classification of Diseases, 10th Revision, Clinical Modification/Procedure Coding System (ICD-10-CM/PCS) codes into clinically meaningful categories.

A COVID-19 positive PCR test was defined as a lab result in which (1) *PCR\_order\_flag* = 1, (2) *result\_positive\_flag* = 1, and (3) *nomenclature\_short\_string* was explanatory for SARS-CoV-2 testing.

Patients were defined as deceased if having *deceased\_flag* = 1 in the patients table, or if having had an encounter with *discharge\_disposition* = “Expired – 20.” Being placed on a ventilator was defined as having a procedure with *ccs\_catg\_dsc* = “Respiratory intubation and mechanical ventilation” that occurred at the time of PCR sample specimen collection or later. Similarly, being admitted to Inpatient care was defined as having an encounter with *enctr\_type\_class* = “Inpatient” at the time of PCR sample specimen collection or later. A patient was defined as having been admitted to the ICU if a patient had a lab test occurring at the time of PCR sample collection or later with (1) *nomenclature\_short\_string* = “Admitted to ICU for condition” and (2) *result\_value* = “YES.”

LOS was defined using the following criteria: First, a COVID-19 positive encounters was defined as an encounter in which *enc\_c19pos* = 1. Next, these COVID-19 positive encounters were considered valid for LOS calculation if the *discharge\_dttm* was specified. If a patient had one or more valid COVID-19 positive encounters, LOS was taken to be the sum of LOS values for each valid COVID-19 positive encounter. By contrast, patients who had only invalid COVID-19 positive encounters (i.e., COVID-19 positive encounters are listed, but all encounters have unspecified *discharge\_dttm*) had LOS defined as not available. Finally, any patient who had no COVID-19 positive encounters (valid or invalid) had LOS defined as zero.

**Feature derivation**

Diagnosis features within each time bin were defined as the presence or absence of diagnoses extracted from the diagnosis table. Any patient with a listed ICD10 code (*icd10\_cm\_cd*) at a time (*active\_status\_dt\_tm*) within a defined time bin was set to TRUE for this diagnosis feature in this time window, whereas this corresponding diagnosis feature was set to FALSE for all other patients in this time window. Infrequently occurring diagnosis features were aggregated using the **icd** library using R package version 4.0.9. (https://jackwasey.github.io/icd/). Patients who were discharged before the beginning of a particular time window had all diagnosis features for this time window assigned as missing.

Similarly, problem features within each time bin were defined as the presence or absence of problems extracted from the Problems table. First, only active entries in the problems table (*life\_cycle\_status* = Active) were used for feature extraction. Next, any patient with a listed SNOMED code (s*nomed\_ct\_code*) at a time (b*eg\_effective\_dt\_tm*) within a defined time bin was set to TRUE for this problem feature, while this problem feature was set to FALSE for all other patients in this time window. Finally, patients who were discharged before the beginning of a particular time window had all problem features for this time window assigned as missing.

In-house medication features and home medication features for each time bin were derived from the InHouse Medications and Home Medications tables, respectively. For in-house medication features, a value of TRUE was assigned if a patient received an in-house medication (*med\_name*) at a time (*start\_date\_time*) within a specified time bin; otherwise, a value of FALSE was assigned. Notably, in-house medications with name “premix diluent” were excluded from in-house medication feature generation. Similarly, for home medication features, a value of TRUE was assigned if a patient was prescribed a medication (*ordered\_drug*) at a time (*req\_start\_dttm*) within a specified time bin, else a value of FALSE was assigned for patients who didn’t receive this medication within this time bin. Patients who were discharged before the beginning of a particular time window had all in-house medication and home medication features for this time window assigned as missing.

Procedure features for each time bin were derived from the Procedures table in a similar approach to the diagnosis features. For procedure features, a value of TRUE was assigned if a patient underwent a procedure (icd10\_pcs\_cd) within a time window of interest (*active\_status\_dt\_tm*), while patients who did not were assigned a value of FALSE. Infrequent procedure features were aggregated using the **icd** library. Patients who were discharged before the beginning of a particular time window had all procedure features for this time window assigned as missing.

Lab value features for each time bin were derived from the Lab Results table. First, any entries in the Lab Results table with undefined numeric values (*result\_value\_numeric*) or normalcy listed as “No Flag” were excluded. Next, exploratory data analysis found that lab results with unspecified normalcy were in the normal range, thus lab results with unspecified normalcy were considered normal. Lab value features were then defined corresponding to each lab test LOINC code (*loinc*) with description (*nomenclature\_short\_string*) measured for a given patient within a time window of interest (*lab\_order\_date*), with features values taken to be the lab value normalcy (Normal, Abnormal, High, Low, or Critical). If a particular lab value was measured more than once for a given patient in a time window of interest, the lab values measured on the nearest date to the date of COVID-19+ PCR specimen collection were used. If multiple tests were reported on this date, the lab result feature was defined as the most severe lab result measured, according to a ranking of Critical > High > Low. Finally, patients who did not have recorded lab values in this time window had their lab value features assigned as missing.

In a similar manner, vital status features for each time bins were derived from the Clinical Events table. First, any entries in the Clinical Events table with normalcy set to “No Flag” were excluded. Additionally, entries with value zero (*result\_val\_numeric*) and unspecified normalcy were excluded. As had been shown with Lab values, exploratory data analysis found that entries with unspecified normalcy were within the normal range, therefore entries with unspecified normalcy were considered normal. Vital status features were then defined corresponding to each event code (*event\_cd*) and event description (*event\_desc*) measured for a given patient within a time window of interest (*valid\_from\_dt\_tm*), with feature values taken to be the vital sign normalcy (Normal, Abnormal, High, or Low). To reduce the number of feature levels, entries with normalcy outside the test bounds (<LLOW and >HHI) were considered LOW and HI, respectively. Vital sign measurements that were listed more than once for a given patient in a time window of interest, were summarized by using the vital sign measured on the nearest date to the date of COVID-19 positive PCR specimen collection. If multiple vital signs were reported on this date, the vital sign feature was defined as the most severe vital sign result measured, according to a ranking of HI > LOW > ABN. Finally, as with the derivation of lab test features, patients who did not have recorded vital sign values in this time window had their vital sign features assigned as missing.

**Elastic Net regression method**

Optimization of the *alpha* and *lambda* parameters demonstrated that setting *alpha* = 1 minimizes the error of the fit for all *lambda* values. The *alpha* parameter determines the relative contribution of LASSO and ridge regression penalties to Elastic Net, so setting *alpha* = 1 simplified Elastic Net regression to LASSO regression. The optimal *lambda* value was selected by a cross-validation approach as provided by the R package **glmnet**.

**Bootstrapping**

To delineate further the basis of the observed variability of the LASSO regression model, and to generate confidence intervals for the generated coefficients, 10 versions of the dataset with imputed values was generated, and each of them in turn was used to generate a population of 1000 datasets with similar underlying distributions, by the bootstrap method. We ensured that each bootstrap sample was balanced in terms of the number of patient-specific mortality. The 10,000 samples were each analyzed by LASSO regression.