CS598 Deep Learning for Healthcare-Final Paper: HEAP-DL – Healthcare Cost Prediction using Deep Learning models

https://www.youtube.com/watch?v=CJY_If9cqzQ

Bhagwat, Rohit
UIUC
rohitb2 at illinois.edu

Mishra, Bigyan Swarup
UIUC
bigyanm2 at illinois.edu

Balasubramanian, Sembian
UIUC
Sembian2 at illinois.edu

ABSTRACT

The healthcare spending in the US continues to grow rapidly, in 2019 (pre-COVID-19) it accounted for more than 17% of the National GDP [1][2]. Accurately predicting future costs and identifying which patients will incur high costs is critical for healthcare providers, payers, and patients to help manage and control resources efficiently. The current methods for predicting healthcare costs include rule-based cost prediction, statistical methods, and supervised learning models. Here, we want to develop a deep learning model on healthcare claims data to accurately predict future healthcare costs as proposed in the paper Deep learning for prediction of population health cost [10]. We will compare the output against ridge regression model.

Keywords

Deep Learning, Healthcare Cost Prediction, Healthcare Claims Data, Unsupervised Learning

1. INTRODUCTION

Predicting healthcare costs has been a challenging problem for various years. Medical costs are influenced by many factors including patient's history, demographics, environmental influences, genetics, and random events [3]. Some of the popular methods of predicting future costs include rule-based prediction, statistical and supervised learning models [4][5]. Rule-based methods can be challenging as they require a lot of domain experience and expertise. On the other hand, statistical methods require huge amounts of data and are susceptible to data veracity. Most popular techniques include supervised learning models which typically require large datasets for training. However, there are limitations of using supervised learning, as they typically do not perform better leveraging interactions between variables in the high dimensional datasets [5].

With recent advancements in scalability of computing, cloud economics of scale, developments in deep learning algorithms, it has made possible to implement systems that learn complex patterns from high dimensional data.

Here, we want to implement a deep learning model and named it as HEAP-DL to predict future health costs [10].

2. MOTIVATION

According to CMS, the national healthcare expenditure increased 4.6% to $3.8 trillion in 2019 and is anticipated to grow even further. [1][2]. Healthcare organizations (HCOs) are actively seeking efforts to accurately predict future healthcare costs so that they can better manage the resources and efficiently plan costs. Accurately predicting future costs is important for various stakeholders including patients, healthcare providers, payers, and policymakers [3].

Healthcare claims data provides structured information on patient interactions with the medical healthcare system over a long period of time and is collected regularly. This dataset provides information including diagnosis, patient demographics, medication details, etc.[10]. We want to prove that this dataset, consisting of a range of features can be used to develop a deep learning model to accurately predict future healthcare costs.

3. LITERATURE SURVEY

With a total healthcare expenditure reaching $3.8 trillion in 2019, analyzing healthcare claims data is much more imperative for extracting valuable insights. Our project work focused on healthcare cost prediction using deep learning models to accurately predict healthcare costs in the US, using claims data which can help manage health costs among the population of patients [1][2][3].

Referring *M. A. Morid et al.*, population health prediction models can inform policymakers about future disease burden and draft the impact of public health actions. In these recent times of COVID – 19 pandemics, such studies can better help & equip policymakers. So far, most of the predictive modeling in population health has used statistical regression models.[4][5]. There has been an evolving interest in healthcare using deep neural networks for cost prediction [7][8][9].

The most frequently used machine learning algorithms (*Morgenstern, Jason Denzil, et al.,*) were neural networks, support vector machines, single tree-based methods, and random forests. The referenced article(s) also made a comparison with statistical methods, which were logistic regression or moving average models with autoregressive integrated [6].

The deep learning literature [7][8][9][10] has identified the immense potential of convolutional neural network (CNN) architecture to represent learning from grid-like data (e.g., claims data). Studies have also applied CNN architecture on multivariate time series data and have used initial network layers to learn temporal patterns from time-series data. Some studies have integrated multiple univariate feature maps at a late stage in CNN architectures as well. [11][12][13].

However, the literature "Deep learning for prediction of population health costs." [10] dives deeper into the application of deep neural networks to predict future costs from health insurance claims records. The authors have applied a deep neural network and a ridge regression model to a sample of German insurant(s) to predict total on-year health care costs. The literature further shows how the parameters can be interpreted and that the model uses relevant features for cost prediction.

This paper is an attempt to implement the aforementioned literature [10] and we gave a new unique name to it, **HEAP-DL**. We have utilized the United States IQVIA PharMetrics® Plus Claims data, provided by IQVIA, to replicate the model and predict costs re-mapped to match the IQVIA dataset.

4. DATA

This study is based on IQVIA PharMetrics® Plus real-world claims' dataset [b].

4.1 IQVIA PharMetrics Plus dataset

IQVIA PharMetrics® Plus dataset is comprehensive real-world data on commercially insured patients in the US with a dimensional view into their treatments, diagnoses, and costs. PharMetrics Plus is a trusted source of data for pharmaceutical – and healthcare industry for supporting a wide range of analytical & data science projects within all phases of development & commercialization. This dataset offers a detailed representation of payers (types), HCOs, specialties, geographic dimensionalities, and connectedness capabilities.

Specifically, it contains a HIPAA [a] compliant longitudinal view of patient services, prescription and outpatient administered medicines, associated costs, and enrollment information for most health plans. Further details & facts about the data can be found elsewhere [b].

All patient- and payer/provider-level information are anonymized to comply with the US Health Insurance Portability and Accountability Act 1996 [a].

4.1.1 Health Plan Enrollment Detail

4.1.1.1 Enroll_Synth

This is a patient-level file of key demographics and enrollment standard variables for the selected population. The file consists of once record per enrollee.

4.1.1.2 Enroll2

This is also a patient-level file with six string-based records per person. These string-based records serve as a month-by-month indicator of coverage at a more granular level.

4.1.2 Claims

The "Claims" files (2015 - 2019) contain transactional records sourced from the claims submitted by providers to health plans for reimbursement. We can see all available medical and pharmacy claims for the selected population or cohort. Claims are typically sorted according to the patient ID variable (PAT_ID), then by date (FROM_DT), which helps ensure that all of the claims for a given patient will be found within a single CLAIMS file. Records contained within the CLAIMS and ENROLL [4.1.1] files can be joined using the PAT_ID variable.

4.1.3 Reference lookup tables

The lookup tables contain information required to interpret and use the diagnosis, procedure, and medication codes.

4.1.3.1 Diagnosis Reference

It contains reference information for all the ICD-9 and ICD-10 diagnosis codes, including diagnosis names and descriptions.

4.1.3.2 Medication Reference

It contains reference information for all the medication NDC codes, including product names, dose form, route of administration, strength, and proprietary drug identifiers (Generic Product Identifier, GPI).

4.1.3.3 Procedure Reference

It contains reference information for all ICD-9 and ICD-10, CPT and HCPCS procedure codes, including procedure names and descriptions.

4.1.3.4 Place of Service

It contains reference information for all codes used to specify where the service was rendered.

4.1.3.5 Revenue Codes

It contains reference information for all codes representing a high-level description of services performed by a hospital/other facility.

4.2 Data Dictionary and User Guide

A detailed data dictionary and user guide is available with project source repository. Further details can be found elsewhere [c].

4.3 Data Characteristics

The available datasets consist of three types of files: health plan enrollment details, claims and reference lookup files for clinical code sets.

The data contains a randomized sample of roughly 30K unique patients (enrollees). Of which, most of the patients are with a medical and/or pharmacy claim. The average length of enrollment for individuals in the dataset is about 25 months. More than 2000 patients have 3 or more years of continuous enrollment. Commercial insurance is the most frequent plan type captured for the enrollee population, but other types are also found, including Commercial Medicare, Commercial Medicaid, Self-Insured, Indemnity and Others. PharMetrics Plus contains information on the pharmacy and medical benefit (copayment, deductible), the inpatient stay and provider details (provider specialty included in all extracts). Economic variables include the negotiated rate between the plan and providers and the actual amount paid by health plans to the provider for all services rendered. Other data elements include dates of service, demographic - age, gender, and geography, claims type, insurance type, and dates of health plan enrollment.

An important limitation of this data is that the richness (in terms of #patients & #cliams) declines from Q2 2017 [Figure 1].

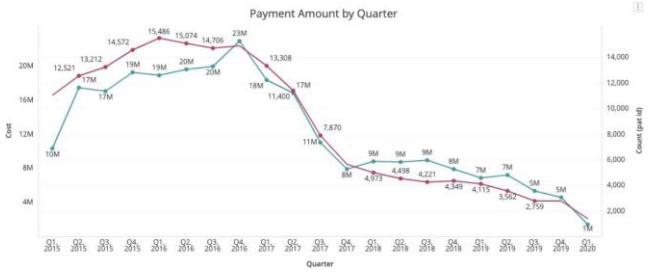

Figure 1 Paid Amount by Quarter

4.4 Data Preparation

The dataset consists of claims records with 73 distinct data elements. Referring *Bertsimas et al* [3], we have identified a set of 20 cost related features. If a record did not have any cost for a specific time it was considered as zero; therefore, there are no missing values in the dataset. The input for algorithm takes the diagnosis codes, procedure codes and place of service that are commonly observed between health plan enrollment details, claims and reference lookup files. The inclusion criteria for patients relies on PAT_IDs between claims and health plan enrollment details.

4.5 Feature Engineering

After careful consideration we finalized the columns to be used for model training as given in Table 1

Feature	Description
pat_id	Encrypted patient identifier
quarter	Derived quarter value from to_dt
from_dt	Date on which services began for inpatient services or date of service for same day services, office visits, outpatient services, etc.
to_dt	The final date of service delivery for a single record or claim period. The To Date is the same as From Date for same day services
duration	Timedelta derived values between from_dt & to_dt
paid_dt	The date the claim was paid
rectype	Record Type: Each record is classified as one of the following six record types: (M)anagement (S)urgical (F)acility (A)ncillary (P)harmaceutical (J) – J code is a non-clinical code used to net Costs associated with a confinement
conf_num	The confinement is constructed by using the 'FROM_DT' on the first room and board record and the 'TO_DT' on the last facility record in a series of facility records that have the same provider id and overlapping or contiguous dates. The CONF_NUM is then assigned to all records that fall within that time frame. Confinement number is not unique within an extract. You must use PAT_ID and CONF_NUM for uniqueness.
icdprc1	Used to report inpatient hospital procedures only. ICDPRC1 is the primary procedure on a facility bill
diag_admit	Also considered the primary diagnosis code. ICD diagnosis code describing the condition chiefly responsible for a patient's admission to a facility. It may be different from the principal diagnosis, which is the diagnosis assigned after evaluation
diag1	These codes describe the patient's condition or diagnosis
proc_cde	A unique code identifying each procedure.
ndc	The NDC is an eleven-digit number that identifies the manufacturer, product name, and package size of each approved or repackaged prescription drug. These codes are assigned by the FDA.
bill_spec	The billing provider's primary specialty. For physicians, this usually represents his/her board registered specialty. For non-physicians, specialty reflects the type of provider/facility
pos	A CMS standard variable identifying the location, or place, where medical services were rendered
der_sex	If there is a sex listed on the enrollment record, it is used. Otherwise, the sex is derived from the claims file
der_yob	If there is a YOB listed on the enrollment record, it is used. Otherwise, the YOB is derived from the claims file.
pat_state	State of residence for the enrollee/patient from the most recent enrollment record
pat_age	Derived age value from der_yob
age_group	Derived age groupings from pat_age
quan	Quantity of drug dispensed expressed in metric decimal units as submitted by the pharmacy.
formulary	At the claim record level, value indicated whether the prescription drug was paid as included in the plan's formulary at the record level. Valid values are: Y Formulary N Non-Formulary
pmt_st_cd	Indicates whether the claim was paid or denied. P - Paid D – Denied
paid	The dollar amount actually paid by the health plan to a provider for services rendered
copay	Amount an insured individual pays directly to a provider at the time the services or supplies are rendered. Usually, COPAY will be a fixed amount per service, such as $15.00 per office visit.
dispense_fee	Dispense Fee. Rx Claim only

Table 1 Feature Description

We chose to include limited fields corresponding to diagnosis code (2 out of 13) and procedure code (2 out of 13) as most of these fields did not have any values. Dates were coded as quarter (Q1, Q2, Q3 & Q4). All the null values in the numerical fields were filled with 0 and in the categorical variables they were populated as '-'. For prediction target a derived column was created as a sum of *paid*, *copay and dispense_fee* columns.

For machine learning, the categorical features (*'rectype', 'icdprc1', 'diag_admit', 'diag1', 'proc_cde', 'bill_spec', 'pos', 'ndc', 'formulary'*) were embedded as either ordinal or one hot coded vector using Scikit-learn [24]. The *duration* for in-patient claims were categorized into seven groups ('<7', '8-15', '16-30', '31-60', '60>') and derived age values into nine different *age_group* ('0-10', '11-20', '21-30', '31-40', '41-50', '51-60', '61-70', '71-80', '81).

Figure 2 shows some key characteristics of the categorical features' observation period and the evaluation period.

```
data_2015[categorical_features].describe().T
```

	count	unique	top	freq
quarter	622158	4	4	175752
rectype	622158	6	A	258213
icdprc1	622158	203	-	620645
diag_admit	622158	579	-	619414
diag1	622158	9618	-	184190
proc_cde	622158	5237	-	202382
bill_spec	622158	64	-	199396
pos	622158	39	11	212779
ndc	622158	10291	-	438606
formulary	622158	5	-	522163
pat_state	603926	51	CA	85034

Figure 2 Statistics of Categorical data

The outliers in continuous features including the target value (i.e. the derived column) were removed using the IQR-Statistic method **Error! Reference source not found.** of identifying outliers, Figure 3. Scaling was applied before feeding to the machine learning model, Figure 4.

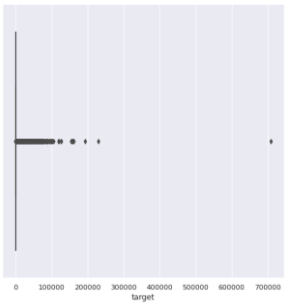

Figure 3 Outliers

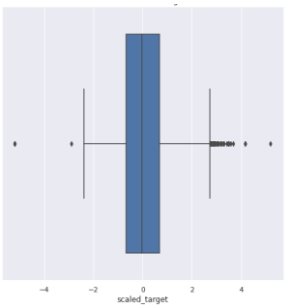

Figure 4 Scaling

Embedding layer is concatenated using *PyTorch nn.Embedding* and fed into the HEAP-DL network. All models have been implemented using *PyTorch* library [28].

4.6 Train and Test Data

As discussed in the previous section, due to the limitations of the dataset available, we have divided the data into *train* and *test* based on a test ratio of 0.2 (20%) we further optimized test ratio 0.1(10%) the model had the best RMSE score.

The data cleansing and enrichment steps were applied to the combined dataset for the period of 2015 and 2016 (claims_2015.dat & claims_2016.dat). The combined 2015 & 2016 dataset was used for training and validation, followed by a split and testing.

5. MODEL/APPROACH

The Figure 5 below describes the approach we adapted during the draft phase. As the first step we cleaned up the dataset and aggregated at a level described in the above sections. We then split the data into Training and Testing/Validation. The training data was then fed Regression Models as well as the HEAP-DL Model with some preprocessing. For the HEAP-DL model we have also applied embeddings. Finally, we applied Integrated Gradients to the Predicted values to determine the features with highest (and lowest) impact to the model.

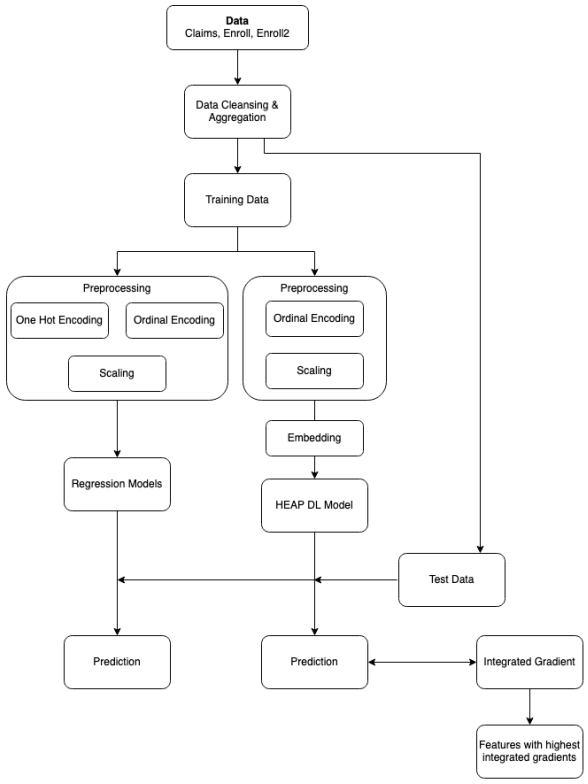

Figure 5 Process Flow for Draft

5.1 Baseline Regression Models

For the baseline regression methods, we have used feature engineered, normalized data and one hot encoding for categorical features and scaled the numerical features. We have run the data on following Scikit-learn regression models with default parameters for the following models [24]: DecisionTreeRegressor, XGBRegressor, CatBoostRegressor, LGBMRegressor, GradientBoostingRegressor, ExtraTreesRegressor and calculated Mean square error and Root mean square error [26][27]. Results of RMSE scores from the baseline model are shown in section 7. RESULTS.

5.2 HEAP-DL Definition

Figure 6 HEAP-DL Model Architecture Diagram

As referenced in the original literature [10], we used a model as shown in Figure 6 with four hidden layers each having 50 neurons. All layers used the ReLU-activation, BatchNorm1D which also acts as a regularization and a dropout rate of 0.2 during training with a learning rate of 0.03. The model uses MSELoss () and Adam optimizer. The categorical features were converted into feature embeddings and passed to the layer and there is a drop out of 0.2 applied to the embedding. The prediction is a transformed sum value of paid, dispense and copay. The HEAP-DL model was trained for 200 epochs and RMSE score [26][27]. Target, a derived numerical value, is a sum of *claims paid amount, pharmacy dispense_fee & patient copay* since these are assumed to be the direct cost for the payer as per the IQVIA PharMetrics® Plus data dictionary and user guide.

In the Data pre-processing step, the claims data is passed thru series of data cleaning and optimization steps and categorical values are label encoded and embeddings were generated and removed outliers and scaled the continuous values using *sklearn Robust Scaler*. The combined 2015 and 2016 dataset with embeddings and continuous variables is then split into train and test, the train data is then concatenated and passed thru the HEAP-DL feedforward network. The Initialization included a drop out of embeddings and batchnorm1D on the continuous data and passed to layer 1 which consists of 50 neurons, the output of layer 1 is then passed to ReLU activation, dropout and Batch Norm1D, the batch Norm1D, dropout was applied to ovoid overfitting, the output of Input layer is then passed to the second layer, the output is then applied with RELU, dropout and batch norm. the same process is repeated for the next 2 layers. The final linear layer has output size of 1 to predict the targets. The training was done for 100 epochs and the end of the training the test data is passed to the model for final predictions.

6. METRICS

For validating the accuracy of baseline regression models and the HEAP-DL, we have used Root Mean Square Error (RMSE) [26][27] as the evaluation metric. RMSE is Square root of Mean Square Error, commonly used in forecasting and regression analysis to validate the results of the prediction, RMSE is the standard deviation of the residuals and a measurement of distance of the regression line to predicted data points are, and show how spread out the residuals or in other words how concentrated the predicted data is to the line of best fit. The model with the lower RMSE is the best model.

$$RMSError = \sqrt{1 - r^2} SD_y$$

From the above formula SDy is the standard deviation of Y, the RMSE close to 0 indicates that the predictions are close to the ground truth and lower the error better the prediction.

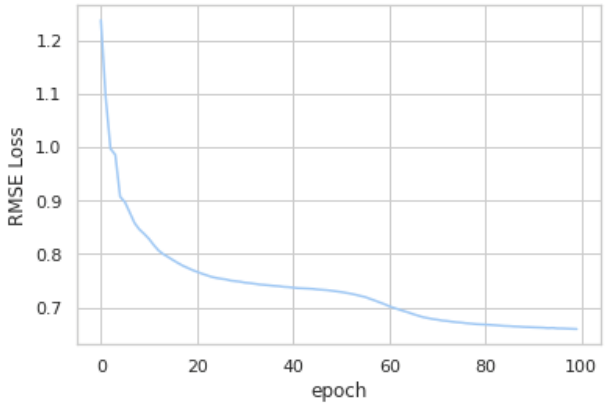

Figure 7 HEAP-DL Training - RMSE Loss/epoch

The final performance of the HEAP-DL model is shown int Figure 7 HEAP-DL Training - RMSE Loss/epoch and we achieved an RMSE score of 0.766 better than the baseline regression model.

7. RESULTS

To highlight the metrics of HEAP-DL model the Model Assessment shows the RMSE scores compared to the baseline regression models. The Table 2 clearly shows HEAP-DL achieved (**0.766**) on the RMSE scores compared to the best regression model **CatBoostRegressor** (**0.783**)

	model	Root Mean Squared Error	Accuracy on Training set	Accuracy on Testing set	Mean square error
2	CatBoostRegressor	0.783696	0.269177	0.259276	0.614180
1	XGBRegressor	0.788533	0.259795	0.250104	0.621785
3	LGBMRegressor	0.797428	0.238243	0.233091	0.635892
5	ExtraTreesRegressor	0.801956	0.548025	0.224358	0.643133
4	GradientBoostingRegressor	0.814544	0.202390	0.199817	0.663481
6	RidgeRegressor	0.817769	0.195310	0.193468	0.668746
0	DecisionTreeRegressor	0.828036	0.548025	0.173087	0.685644

Model	Root Mean Square Error (RMSE) – Test Set
Baseline Regression Models	
CatBoostRegressor	**0.783**
XGBRegressor	0.788
LGBMRegressor	0.797
ExtraTreesRegressor	0.801
GradientBoostingRegressor	0.814
Ridge Regressor	0.817
DecisionTreeRegressor	0.828
HEAP-DL	**0.766**

Table 2 Model Assessment

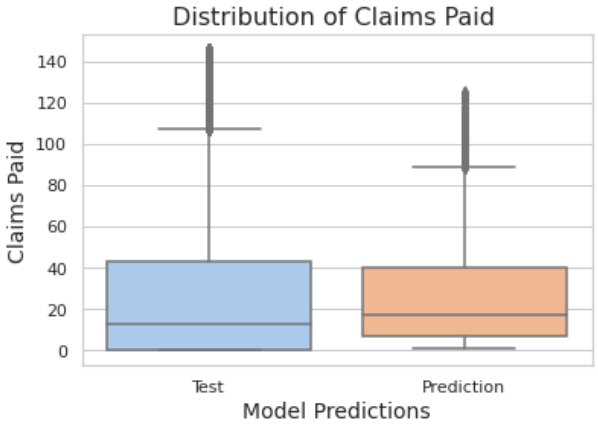

Figure 8 HEAP DL Results

Figure 8 HEAP DL Results shows a box plot of test targets and predicted claims amount.

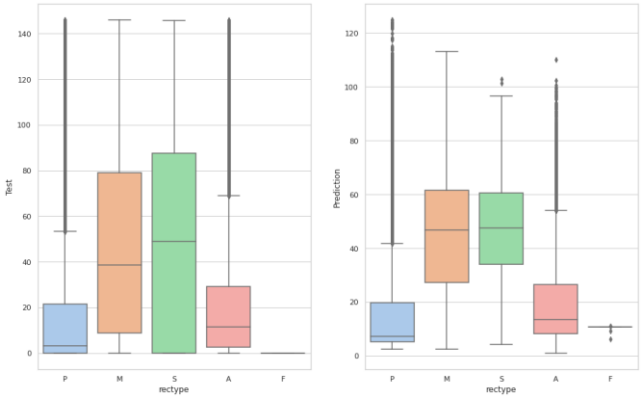

Figure 9 HEAL-DL Results by Record Type

Figure 9 HEAL-DL Results by Record Type shows a box plot of test targets and predicted claims amount by record type, given the data samples are mostly Pharmacy & Ancillary claims the model performed well on those categories adding more training samples of other types of claims would improve predictions on the other categories.

7.1 HYPER PARAMETER TUNING

7.1.1 Model Parameter Tuning

We implemented wandb.ai [19] library to run model experiments and ran 200 different experiments with different variations of model parameters and layer configurations. Link to Hyperparameter tuning experiments [wandb.ai Report] shows our best score was achieved with the following parameter settings with: Figure *10* and Table 3Figure 11.

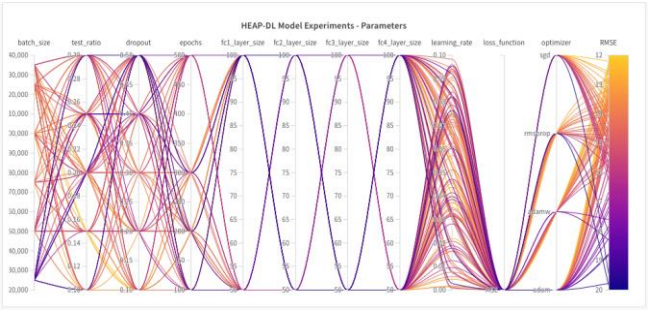

Figure 10 wandb.ai Model Experiments

Parameters	Value	Sensitivity
Wandb.ai Experiments		
Batch_size	>135000	High
Test_ratio	0.15	High
Dropout	0.1	High
Epochs	200	Low
Layer Sizes (fc1/fc2/fc3/fc4)	[100,100,100,50]	Low
Learning rate	0.03	Low
Optimizer	Sgd, Adam	Low

Table 3 Hyperparameter Tuning

We also looked at the correlation of different parameters to loss scores and identified the top parameters to optimize to the final model, Figure 11.

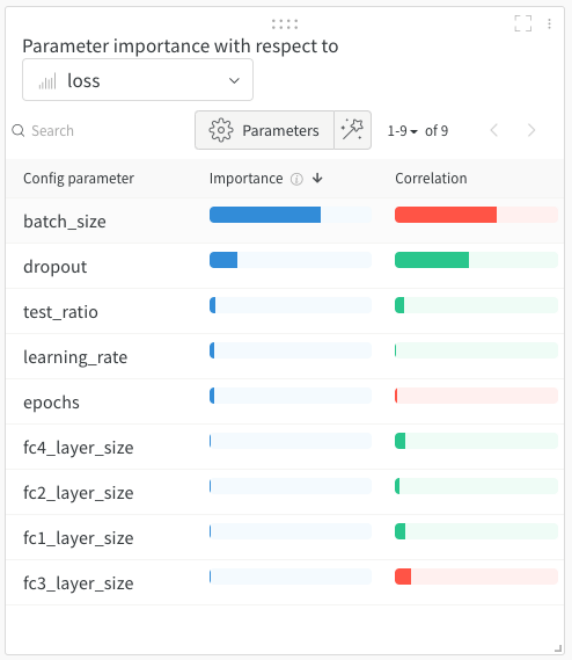

Figure 11 Parameter Importance with respect to RMSE Loss

7.1.2 Addressing Overfitting

To address the overfitting of the model we implemented several methods that include, increasing the training data size, adding batch

normalization increasing the dropout rate, tune the hyperparameters and reduced the epochs and added additional feature embeddings.

7.2 Interpreting model predictions

Understanding the reason behind model predictions is a critical part in any applications using ensemble or deep neural network models. Interpreting complex models to understand the reasoning behind the predictions on why a specific prediction was made and the importance of a given feature towards the prediction. We explored and implemented two frameworks SHAP (*SHapley Additive exPlanations*) [30] on ensemble baseline regression models and *Captum* [25] for the HEAP DL Model.

7.2.1 SHAP

SHAP (*SHapley Additive exPlanations*) [30] is a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations representing accurate additive feature attribution method based on expectations. We implemented the SHAP value explainer on the Ridge regression model to understand the feature importance in predicting the target amount. As shown in Figure 13 SHapley values for Ridge Regression & Figure 12 SHAP Summary plot showing feature attributions.

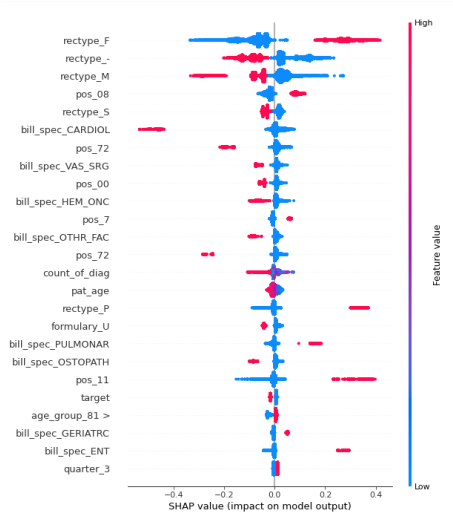

Figure 12 SHAP Summary plot showing feature attributions

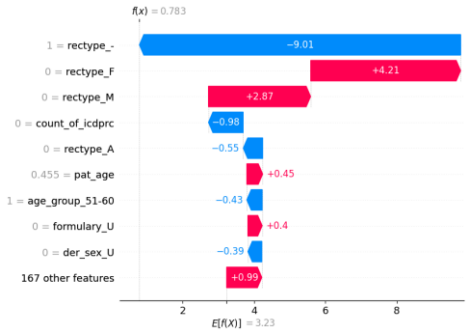

Figure 13 SHapley values for Ridge Regression (Sample[i])

7.2.2 Captum Layer Integrated Gradients

The HEAP-DL model is saved after the training using *PyTorch.save* including the *state_dict* option. The model is then loaded and embedding dimensions were passed to the Feedforward network. The categorical and continuous values were passed to the Integrated Gradients.

We have used the *Captum PyTorch* library [25], which is a model interpretability and understanding library for PyTorch. To facilitate the identification of features that contribute to our model's prediction of paid value, we have used the Layer Attribution - *LayerConductance* algorithms [22][23]. The conductance combines neuron activation with partial derivatives of neurons with the respect to input and output to build a neuron importance. This method builds on Integrated Gradients (IG) by looking at the flow of integrated gradients attribution which occurs through the hidden neuron.

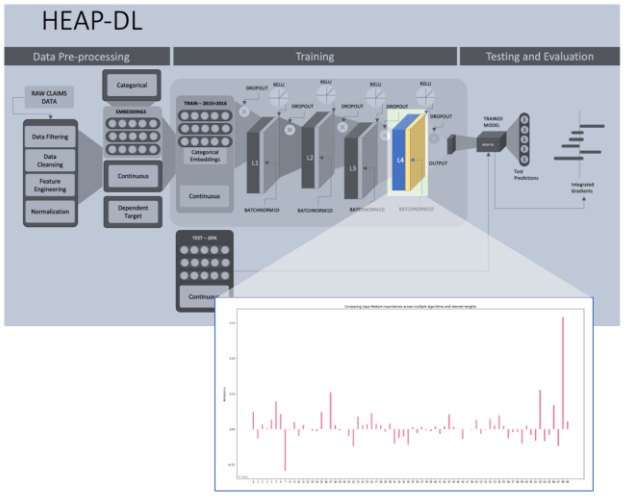

Figure 14 Integrated Gradients and Feature Attributions

For the algorithms specified in original literature [10], we want our HEAP-DL to derive Layer Integrated Gradients using gradient weights. For the final presentation, we look forward to achieving feature attributions and identifying feature names that are positively correlated to the prediction and can be used for better RMSE scores and prediction cost.

8. ENVIRONMENT AND PROJECT CODE

Our team has utilized an online notebook collaboration platform from JetBrains' *Datalore* [e]. The shared, real-time notebook runs on a 'large machine' instance (2 vCPU cores, 16GB RAM). Final results were extracted using 'GPU machine' instance (1 NVIDIA T4 GPU, 4 vCPU cores, 16 GB RAM).

The final code of the HEAP-DL model is available via : https://datalore.jetbrains.com/view/notebook/OPttEfFZr98HrUIfA3yNPS

9. DISCUSSION

Accurate prediction of future health care cost can help to better manage healthcare cost and resources. Knowing in advance that few patient cohorts are at risk of an increase in expenditure for the next year, patients can choose better expanded insurance coverage and plan. Insurance companies can better plan and effectively control cost as well. Place of service (providers) can take measure to reduce cost too.

Our study followed the path from aforementioned literature [10] and showed benefits of deep learning methods over frequently used machine learning regression models. We have also attempted to measure the effects of different types of features on overall cost prediction. The results in Figure 8 also suggests that, even with a reduced variety of predictors and lower amount of data for training the cost predictive model, the proposed HEAP-DL still has higher accurate predictions.

The focus of this HEAP–DL project work was on achieving the accuracy of the cost prediction methods with state-of-art deep learning methods [10], which has several important practical applications, as explained before. Although this study, so far, showed the feature-level importance of the input in proposed model, a limitation of HEAP-DL is the lack of individual-level explainability of the input features' contributions to the proposed model. Whereas deep learning models with categorical target variables can be explained to some extent by using visualization techniques, this model for numeric prediction is limited.

10. CHALLENGES AND LEARNING

Feature Encoding Baseline: While we were able to test the baseline regression models due to the large labels of datasets, including the diagnosis and procedure codes. We had limitations on no. categorical feature encoding on the baseline model to 177 features. We overcame the Embedding layers approach and added additional feature embeddings resulting in embedding dimensions larger than baseline encoding.

Model Training: Our initial model training iterations were set to 500 epochs and observed overfitting, so we optimized the model hyperparameters, increased the dropout, added more training data, included BatchNorm1D to the layer structure, finally reduced the no. of epochs to 100.

Integrated Layer Gradients: While working with Captum Integrated Gradients we faced challenges on getting the names of embedding features, as proposed under Future work the embedding names could be extended and derived using Interpretable Embeddings using Captum Library, further extensive data analysis can give us more features' correlation with cost prediction. We also argue that integrated gradient attribution has some limitation, and the use of more strategies to interpret IGs, might give better insights into the prediction accuracy.

11. FUTURE WORK

Currently the integrated gradients of the embedding are shown in Figure 14 Integrated Gradients and Feature Attributions and in the final project we plan to identify the feature names from integrated gradients and also try other model interpretability libraries.

Additional optimizations could be applied to the feature embeddings by utilizing the Integrated Gradients and SHAP values.

Extend wandb model experiments with additional model parameters to further optimize the RMSE scores.

Additional Temporal features using date could be implemented to improve the predictions.

Forecasting of claims cost using PyTorch Forecasting would also be beneficial given the structured nature of the data using the temporal *paid_dt* feature.

12. CONCLUSION

With the work done so far, we have attempted to achieve a higher level of accuracy for cost prediction. A mapping of numerical cost prediction to various population groups and key insights Final Cost predictions are given below:

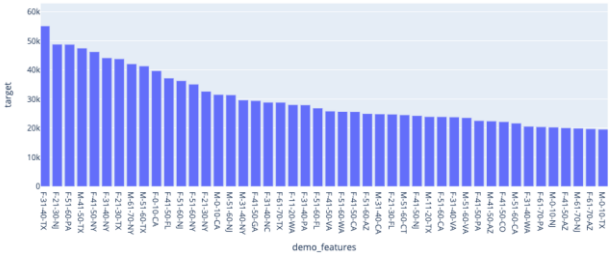

Figure 15 Final Cost Predictions by Age Group / State

Figure15: Based on the cost prediction TX-Texas age 31-40 & 41-50, NJ-New Jersey age 21-30 & 41-50, PA- Pennsylvania age 51-60, are the Top 3 states with highest cost

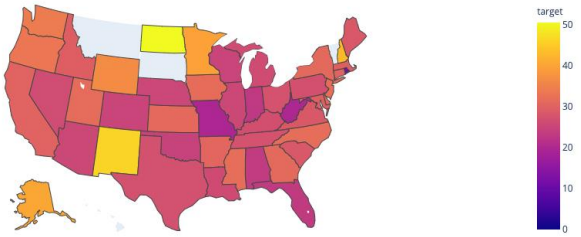

Figure 16 Final Mean Cost Predictions by State

Figure16: Based on the mean cost prediction New Mexico ($46) & North Dakota($50) have the highest mean spend in the United States

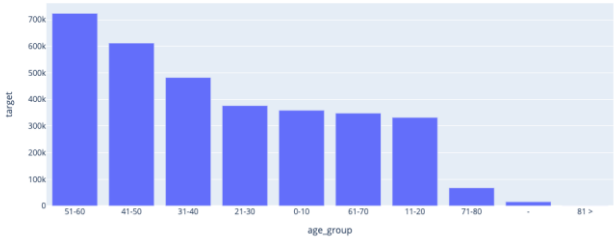

Figure 17 Final Cost Predictions by Age Group

Figure17: Based on the cost prediction by age group 51-40, 41-50, 31-40 have the highest mean spend

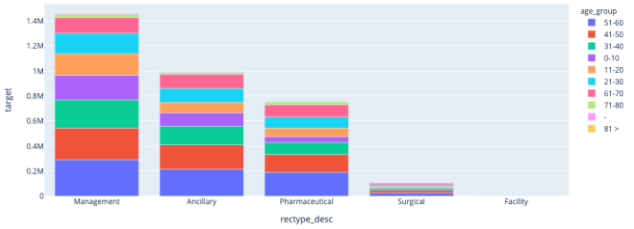

Figure 18 Final Cost Predictions by Claims Record Type and Age Group

Figure 18: Based on the cumulative cost prediction by Claims record Type and age group, Management fees for age group 51-60 have the highest spend.

13. TASK DISTRIBUTION

The overall task distribution will be a collaborative effort in determining the interests and strengths of our team members. And balance our project workload in an effort to learn from each other and achieve the overall project goals. Below is the individual task distribution summary for our project work.

13.1 Rohit

Environment Setup, Gitlab coding environment, Raw Data Exploratory Analysis, Model Design & python coding implementation, Data Preparation for models, Model Documentation, Literature review and analysis, Deliverable owner for Final Code & Paper submission to Box, Video Presentation recording and demo

13.2 Bigyan

Model Design & python coding Implementation, Data modelling, Exploratory Data Analysis, Data Preparation pipeline, Reporting and documentation, Code review and testing , Develop Comparative baseline model, Literature review and analysis, Deliverable owner for Project draft Submission to Box, video script and presentation

13.3 Sembian

Model Design & python coding Implementation, Data Pre-processing pipeline, Develop Comparative baseline model, Documentation, Reporting and Visualizations, Project Planning – Asana/Microsoft Teams Tool setup, Literature review and analysis, Deliverable owner for Project Proposal submission to Box, Final Video Presentation recording, editing and posting the video to YouTube as unlisted and share link in final paper.

14. ACKNOWLEDGMENTS

We are thankful to our mentor Rick Barber and our course professor Jimeng Sun for all their guidance and support.

15. REFERENCES

[1] I. Duncan, M. Loginov, M. Ludkovski, Testing Alternative Regression Frameworks for Predictive Modeling of Health Care Costs, North Am. Actuar. J. 20 (2016)65–87.

[2] A.B. Martin, M. Hartman, B. Washington, A. Catlin, T.N.H.E.A. Team, National. Health Care Spending In 2017: Growth Slows To Post–Great Recession Rates; Share Of GDP Stabilizes, Health Aff. 38 (2019) 10.1377/hlthaff. doi:10.1377/hlthaff.2018.05085.

[3] Bertsimas D, Bjarnadóttir MV, Kane MA, Kryder JC, Pandey R, Vempala S, et al. Algorithmic prediction of health-care costs. Operations Research. 2008;56(6):1382-92.

[4] M.A. Morid, K. Kawamoto, T. Ault, J. Dorius, S. Abdelrahman, Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation., in: Proceeding Am. Med. Informatics Assoc., 2017: pp. 1312–1321.

[5] M.A. Morid, O.R.L. Sheng, K. Kawamoto, T. Ault, J. Dorius, S. Abdelrahman, Healthcare cost prediction: Leveraging fine-grain temporal patterns, J. Biomed. Inform. 91 (2019), https://doi.org/10.1016/J.JBI.2019.103113.

[6] Morgenstern, Jason Denzil, Emmalin Buajitti, Meghan O'Neill, Thomas Piggott, Vivek Goel, Daniel Fridman, Kathy Kornas, and Laura C. Rosella. "Predicting population health with machine learning: a scoping review." BMJ open 10, no. 10 (2020): e037860.

[7] S. Amari, The handbook of brain theory and neural networks, 2003.

[8] M. L¨angkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24, https://doi.org/10.1016/J.PATREC.2014.01.008.

[9] G.K. Dziugaite, D.M. Roy, Z. Ghahramani, Deep Learning, MIT Press, 2016.

[10] Drewe-Boss, Philipp, Dirk Enders, Jochen Walker, and Uwe Ohler. "Deep learning for prediction of population health costs." arXiv preprint arXiv:2003.03466 (2020).

[11] F. Wang, N. Lee, J. Hu, J. Sun, S. Ebadollahi, Towards heterogeneous temporal clinical event pattern discovery, in: Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '12, ACM Press, New York, New York, USA, 2012: p. 453. 10.1145/2339530.2339605.

[12] P.W. Mirowski, Y. LeCun, D. Madhavan, R. Kuzniecky, Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG, in: 2008 IEEE Work. Mach. Learn. Signal Process., IEEE, 2008: pp. 244–249. 10.1109/ MLSP.2008.4685487.

[13] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks, in: Springer, Cham, 2014: pp.298–310. 10.1007/978-3-319-08010-9_33.

[14] Morid, Mohammad Amin, Olivia R. Liu Sheng, Kensaku Kawamoto, and Samir Abdelrahman. "Learning hidden patterns from patient multivariate time series data using convolutional neural networks: A case study of healthcare cost prediction." Journal of Biomedical Informatics 111 (2020): 103565.

[15] Kim, Byung-Hak, Seshadri Sridharan, Andy Atwal, and Varun Ganapathi. "Deep Claim: Payer Response Prediction from Claims Data with Deep Learning." arXiv preprint arXiv:2007.06229 (2020).

[16] Maisog, José M., Wenhong Li, Yanchun Xu, Brian Hurley, Hetal Shah, Ryan Lemberg, Tina Borden et al. "Using massive health insurance claims data to predict very high-cost

claimants: a machine learning approach." arXiv preprint arXiv:1912.13032 (2019).

[17] Yang, Chengliang, Chris Delcher, Elizabeth Shenkman, and Sanjay Ranka. "Machine learning approaches for predicting high cost high need patient expenditures in health care." biomedical engineering online 17, no. 1 (2018): 1-20.

[18] Lin ED, Hefner JL, Zeng X, Moosavinasab S, Huber T, Klima J, Liu C, Lin SM. A deep learning model for pediatric patient risk stratification. Am J Manag Care. 2019 Oct 1;25(10):e310-e315. PMID: 31622071.

[19] Biewald, Lukas. "Experiment Tracking with Weights and Biases." (2020).

[20] Lakhani, Chirag M., Braden T. Tierney, Arjun K. Manrai, Jian Yang, Peter M. Visscher, and Chirag J. Patel. "Repurposing large health insurance claims data to estimate genetic and environmental contributions in 560 phenotypes." Nature genetics 51, no. 2 (2019): 327-334.

[21] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." In International Conference on Machine Learning, pp. 3319-3328. PMLR, 2017.

[22] Shrikumar, Avanti, Jocelin Su, and Anshul Kundaje. "Computationally efficient measures of internal neuron importance." arXiv preprint arXiv:1807.09946 (2018).

[23] Dhamdhere, Kedar, Mukund Sundararajan, and Qiqi Yan. "How important is a neuron?." arXiv preprint arXiv:1805.12233 (2018).

[24] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[25] Captum PyTorch Library https://github.com/pytorch/captum

[26] Root-mean-square deviation(RMSD) or Root-mean-square-error(RMSE) https://en.wikipedia.org/wiki/Root-mean-square_deviation

[27] Stephanie Glen. "RMSE: Root Mean Square Error" From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/

[28] PyTorch is an open source machine learning framework that accelerates the path from research prototyping to production deployment. Learn more at pytorch.org.

[29] Lundberg, Scott M., Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. "From local explanations to global understanding with explainable AI for trees." Nature machine intelligence 2, no. 1 (2020): 56-67.

[30] Lundberg, Scott, and Su-In Lee. "A unified approach to interpreting model predictions." arXiv preprint arXiv:1705.07874 (2017).

[31] Lundberg, Scott M., Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. "From local explanations to global understanding with explainable AI for trees." Nature machine intelligence 2, no. 1 (2020): 56-67.

[32] Lundberg, Scott M., Bala Nair, Monica S. Vavilala, Mayumi Horibe, Michael J. Eisses, Trevor Adams, David E. Liston et al. "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery." Nature biomedical engineering 2, no. 10 (2018): 749-760.

[33] Troubleshooting Deep Neural Networks. A Field Guide to Fixing Your Model: http://josh-tobin.com/troubleshooting-deep-neural-networks

[34] Engineering Statistics Book: What are outliers in the Data https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

16. SUPPLIMENTARY DATA

a. Health Insurance Portability and Accountability Act – [Link]

b. IQVIA PharMetrics® Plus – [Link]

c. IQVIA Data Dictionary and User Guide – [Link]

d. National Healthcare Expenditure Fact Sheet – [Link]

e. JetBrains Datalore – [Link]

About the authors:

Author 1 Rohit Bhagwat has a Bachelor's Degree in Computer Science from Rashtrasant Tukdoji Maharaj University, Nagpur. Currently working as a Senior Architect (Data Engineering) for InfoCepts LLC. As a part of this role, Rohit is responsible for helping clients of InfoCepts LLC architect and build Data and Analytics Platforms to support their business.

Author 2 Bigyan Swarup Mishra has a Bachelor's Degree in Computer Science from RTM Nagpur University. Certified in data story-telling & design thinking, currently working as Senior Project Lead (Data Analytics' UI) for InfoCepts LLC. As a part of this role, Bigyan is responsible for building Business Intelligence & Analytics' solutions for pharmaceutical clients. An avid Sherlock Holmes fan!

Author 3 Sembian Balasubramanian has a Master's Degree (MFA) from University of Madras specialized in Visual Communication Design and Masters in Business Administration (MBA) from University of Illinois Urbana Champaign with certifications including PMP, Six-Sigma & Certified Usability Analyst (HFI). Sembian's invention include an approved Patent for Application Security Framework (USPTO Link). Currently working as Director of Medical Program Management supporting Women's Health at AbbVie US Medical Affairs.