Data Challenge

This is an exercise to assess analytical, communication, and statistical skills. This questionnaire covers a broad range of topics related to the analysis of radiology reports.

Study Design (20 points)

Suppose that you are interested in assessing the inter-reviewer variability among a set of reviewers across exams, where inter-reviewer variability is the degree of agreement among independent reviewers who assess the same exam. For the sake of simplicity, let us assume that the reviewers study ACL exams (original reader's diagnosis and images) done by original readers, and give their assessments in terms of an agreement value: 1) agree, 2) false negative, 3) false positive, 4) undercall, or 5) overcall. False negative occurs when the reviewer thinks that the ACL is abnormal but the original reader thought that the ACL is normal. Moreover, let us assume that the review assessments are used to assess the diagnostic accuracies of the original readers.

1. Why is it important to assess the inter-reviewer variability? Do you expect that the inter-reviewer variability varies across agreement values?
2. How would you assess the inter-reviewer variability? What kind of data would you need to have?
3. Can you think of any use cases for the inter-reviewer variability?
4. What are the limitations associated with the approach you proposed in question (2)?

[A1.1] Why important?

Inter-reviewer variability tells us the consistency in the "labeling" (i.e. diagnoses in this case) of the target instances (ACL exams).

Consistent diagnoses across different reviewers ensure, with a higher degree confidence, that the diagnosis is valid and reliable; by contrast, inconsistent diagnoses could indicate the lack of reliability of the diagnosis in at least two different aspects:

i) Either party of the reviewers may not be sufficiently skilled or experienced in diagnosing this target disease ii) The target disease itself is so complex with many unknown clinical properties that there isn't yet an efficient way for accurate diagnosis. Incidentally, we may quantify the situation in ii) with the notion of Bayes error, meaning that there is an irreducible error that even experts or groups of experts cannot (yet) overcome. For complex diseases with a relatively high Bayes error rate in diagnosis, we would expect to observe a relatively low inter-reviewer agreement. Very intricate, fine ACL tears could have relatively larger Bayes error (than common cases).

In developing AI and ML systems, inter-reviewer variability is a good measure for **data quality** in that it tells us how reliable the "gold standard" is. Low predictive performance of an ML model could be due to the labeling inconsistency in the training data.

[A1.2] Do I expect it to vary across different agreement values?

The short answer is yes, and performing multiple experiments on different sets of patients will reveal their differences if they exist. I am going to create an running example to facilitate the answers to other questions as well.

First, let's consider the format of the data for investigating inter-reviewer variability. It is reasonable to assume that the data comes in the form of frequencies or counts in an inter-reviewer study with a fixed set of agreement values (5 categories), where different agreement values typically have different associated counts (of cases).

To observe inter-reviewer variability across agreement values, consider statistics such as the mean and variance of counts within each agreement categories across multiple experiments.

Different agreement values usually have different clinical implications. For instance, FN is usually a more severe misdiagnosis relative to FP; intuitively, I'd expect to observe different statistics between FN and FP cases. Depending on the nature of the target disease, the statistics for positive and negative examples can be quite different. Rare diseases, for instance, have a highly skewed data distribution and therefore, FN counts are typically smaller than FP counts (because the sample size of positive cases is small to begin with; FNs cannot be too big). Since the range of FN counts is small and limited (say we only have 2 positive cases out of 100 in an experiment, in which we have one type-FN disagreement, one agreement), chances are that the scale of their variance would be smaller than that of the FP cases (with a wider count variability due to large volume of negative cases).

On the other hand, for conditions with easy diagnosis, it's entirely possible that two reviewers (or two groups of reviewers) tend to agree more often than not and therefore, the (mean) counts for 'agree' tend to be higher than those of the other agreement categories corresponding to misdiagnosis of different types.

It's better illustrated with a running example as shown in Table I (also helpful for answering the remaining questions); additionally, Table I also reveals potential problems and limitations with the existing 5 agreement values/categories.

To account for undercall and overcall cases, we can assume, for simplicity, that an ACL injury diagnosis can be classified in terms of severity degrees (say 1 – 3, mild to severe) and therefore, the diagnosis is viewed as ordinal, not just categorical. For ease of exposition, denote the control as 0 (i.e. healthy or no ACL injury) and denote undercall as UC and overall as OC.

Case	Original Reader	Reviewer	Agreement I	Agreement II (fine-grained)	Count	Note
A	0	1	FN	FN(1)	7	FN with an "disagreement differential" of 1
B	0	3	FN	FN(3)	2	FN with disagreement differential of 3; i.e. Case B has a more significant misdiagnosis error than Case A
C	1	0	FP	FP(1)	3	
D	2	0	FP	FP(2)	1	
E	2	3	UC	UC(1)	10	Minor UC disagreement
F	1	3	UC	UC(2)	5	A larger UC disagreement relative to case E
G	3	2	OC	OC(1)	8	
H	3	1	OC	OC(2)	3	
I	3	3	Agree	TP(3)	20	TP at high severity
J	1	1	Agree	TP(1)	30	TP at low severity
K	0	0	Agree	TN	11	

Table I. Different inter-reviewer agreement scenarios

As shown by the table above, I've listed a few representative scenarios (A-K) that encompass the 5 major agreement values. In medical diagnosis, there can be an arbitrary number of cases in each scenario (in the column "Count" I made up imaginary count of cases to make it easy to conceptualize). Table I also illustrates two different schemes in agreement values (differing by granularity): i.e. Agreement I vs Agreement II (see the note column for interpretation).

The above table constitutes one study (n=100). If we were to conduct another study with another 100 patients evaluated by the same two groups of reviewers, we would expect to observe a different count distributions across different agreement values. If two (groups of) reviewers consistently agree, for instance, then we would expect to observe similar counts in 'agree' as an agreement value.

Inter-reviewer variability also depends on agreement values (e.g. their scale and granularity), which leads to different effect size. As shown in Table I, the same mis-diagnosis of type FN can exhibit different subtilties. For instance, FN(3) occurs when the reviewer concludes a severe ACL injury case whereas the original reader somehow failed to diagnose it (severity difference is 3-0=3). FN(3), for instance, is a more serious mis-diagnosis case relative to FN(1) when the original reader failed to diagnose a case but only for a mild case. Please see the "note" column for interpretation of the alternative agreement value (i.e. Agreement II).

[A2] How would you assess the inter-reviewer variability? What kind of data would you need to have?

I would first consider, for example, Cohen's Kappa, since this is a case involving two reviewers (or two groups of reviewers) with a finite set of possible diagnoses. More precisely, however, these diagnoses are ordinal (instead of merely categorical) to account for the possibility of overcall and undercall. Viewing the diagnosis as an ordinal variable, I would also consider Spearman's rank correlation (where ranks can be defined in terms of severity degrees) and other rank statistics.

To use Cohen's Kappa, we will first structure the data in the form of counts like Table I, assuming that we have a finite set of diagnoses as 3 severity stages + 1 control (or non-case) for a total of 4 different diagnoses. Certainly, the data could also come in the form of continuous measurements (e.g. inflammatory biomarkers like C-reactive protein or CRP), in which case, we could discretize the value to reuse the same analysis, or use other statistics such as Pearson correlation as an agreement measure).

We can then build a **contingency table** with 4 different diagnoses, where horizontal direction represents the original readers (reviewer A) while the vertical direction represents another group of reviewers (reviewer B). We then count the number of cases for each diagnosis category, leading to 16 different counts.

One can imagine if two groups of reviewers tend to agree with each other, then the diagonal cells will have relatively larger counts. Cohen's Kappa allows us to compare the observed agreement (i.e. sum of counts in the diagonal divided by total cases) against chance agreement; the closer to 1, the higher the agreement.

We can certainly devise our own statistic by giving different weights to different types of agreements (e.g. agreement with an early onset of a hard-to-diagnose disease is a better accomplishment than agreement on severe cases, which are easy to diagnose anyway).

To use Spearman's correlation, we will structure the data in the form of a table that compares two reviews (say reviewer A, and B as two rows) against the patients (say, n=100 i.e. p1, p2, … p100), where each cell represents the raw diagnosis (0 for control, 1-3 for different severity degrees). We can then compute the Spearman's rank correlation as a measure of agreement: if two reviewers tend to agree, then they tend to rank the severity degrees consistently.

In practice, I tend to use multiple metrics instead of relying on only one metric.

[A3] Can you think of any use cases for the inter-reviewer variability?

One immediately similar use case is to test if two classifiers tend to agree in their predictions. Going back to the ACL injury diagnosis example, we can indeed view the two reviewers as being analogous to two classification algorithms (e.g. SVM vs Gradient Boost Tree), which produces the severity degrees as predictions. Certainly, we can generalize the setting two regression algorithms that output continuous values as predictions; in this case, we will then use an appropriate statistic as a measurement of agreement such as Pearson's correlation.

Another similar but different use case is to test if two datasets (with potentially different labeling consistency and hence different data quality) tend to result in consistent classification results by the same classifier. This is a use case in data-centric AI (notably advocated by Andrew Ng and his group at Deeplearning.AI), where instead of striving to come up with a better model (as in model-centric AI), you try different ways to improve the data quality, leading to versions of data sets, and hopefully the iteratively improved data quality can drive better predictive performance (without changing the model). Inter-reviewer agreement can then be measured in a similar fashion by comparing two versions of the data (labeled by two different labeling standards, strategies, groups of experts etc.) and observing if predictive results (e.g. severity degrees) are consistent.

Other use cases share the same pattern. E.g., comparing movie ratings from two different customers (do they rank the same set of movies consistently?); by symmetry, we can also compare two movies to see if they are consistently rated by a set of users.

We can, of course, generalize the inter-reviewer agreement to more than two reviewers (say, we want to compare multiple classifiers and their predictive consistency), it's a matter of using different statistics such as Fleiss' Kappa (as a generalization over Cohen's Kappa for mutli-rater settings), or, we could choose to answer a simpler (but in a sense weaker) statistical question: Is there a difference in performance among the set of classifiers and is the difference statistically significant? i.e. a yes-and-no question rather than quantifying to which degree their predictions agree with each other. In this case, **Friedman test** followed by an appropriate post-hoc test (e.g. **Nemenyi test**) is a candidate approach to finding which (subset of) classifiers performan consistently better.

[A4] What are the limitations associated with the approach you proposed in question (2)?

Cohen's Kappa for example is originally designed to compare two reviewers, each classifies a set of N items (e.g. patients) into m categories (e.g. diagnoses). But of course, one can think of these as more of the property of Cohen's Kappa rather than limitations: For multiple reviewers, we use a generalized statistics such as Fleiss' Kappa; for ordinal variables, we can use Spearmen's correlation instead, and for continuous variables, Pearson correlation is an option.

Given that the diagnoses are actually ordinal variables (to account for the possibility of overcall and undercall), using statistics like Cohen's Kappa is at best an approximation for the true statistics required for ordinal setting; we may wish to incorporate rank statistics instead.

Cohen's Kappa also does not work well with skew datasets such as the diagnosis of rare diseases; it tends to underestimate rare categories (e.g. misdiagnoses of type FN tends to be much smaller than type FP given that positive examples are disproportionally small) and hence the resulting agreement tends to be overly conservative.

Real-World Data (20 points)

Suppose that you have used claims data to assess the impact of high quality in radiology on downstream costs. As before, downstream costs are defined as the yearly overall medical costs that follow the initial image. Based on this analysis you have found that patients imaged in high quality radiology centers have, on average, lower downstream costs. These results are statistically significant (when considering a test size of 0.05). However, further data exploration suggests that patients imaged in a high quality radiology center were more likely referred by a highly specialized provider (such as an orthopedic surgeon) while patients imaged in low quality centers were more likely referred by non-highly specialized providers (such as a nurse practitioner).

1. What steps would you take before communicating any conclusions on the nature of the association between radiology quality and downstream costs?

This is an example that illustrates how confounding variables, such as *provider*, can impact the association between explanatory variable (e.g., *radiology quality*) and response variable (e.g., *downstream cost*). That is, the confounder – provider specialization (z) – influences both radiology quality (x) and downstream cost (y), leading to a potentially spurious association between x and y.

Specifically, this problem can be formulated as a regression problem, where the radiology qualify (x) is an explanatory variable and downstream costs as the response (y) that we are trying to predict. A high degree of association is then reflected by a high correlation between x and y, in which case, x's coefficient would have a small p-value, suggesting strong evidence of being "away" from the null hypothesis (that x and y is not correlated).

If a variable z confounds the relationship between x and y, then by introducing z to the model (i.e. predicting y given x while adjusting for z), we would expect to observe a drop in the correlation/association between x and y. This is also the notion of "explaining away" in that by introducing z, the association of x and y will no longer be significant (or at least become less significant), as reflected by x's coefficient having a large p-value, pushing toward the null hypothesis.

Therefore, the steps of verifying the association between radiology quality (x) and downstream cost (y) is to first brainstorm what other variables could potentially influence radiology quality (x), the cost (y) or both (aka **disjunctive cause criterion** in casual inference)). A simple way to do this is perhaps to identify a large set of variables (e.g. age, gender, population) and systematically and iteratively build multiple regression models, adjusting for these new variables, and observe the change of x's coefficient (in terms of magnitude and p-value) before and after adjusting for these variables.

More sophisticated way would involve hypothesizing candidate Bayesian networks and control for confounders (between quality and cost) by blocking all backdoor paths (as much as possible), which is also known as **backdoor path criterion** in causal inference

Machine Learning (20 points)

In radiology, misdiagnoses can have different implications. Some misdiagnoses may have clinically impactful consequences while others will not impact a patient's clinical outcomes. Suppose that you are building a model to predict a rare but clinically impactful misdiagnosis. You can assume that in your sample you have 500,000 patients that have been correctly diagnosed and 10,000 that have been misdiagnosed.

1. Describe a specific set up to train and test your model.
2. How would you assess the performance of your model?
3. How would you communicate your findings to a non-technical audience?

[A1]
For ease of exposition, denote the dataset associated with correct diagnosis as Dc and the dataset associated with misdiagnosis as Dm.

Since we wish to focus on identifying the misdiagnosed cases (rather than the typical case-versus-control setting), as a baseline, we can train a binary classification model (e.g. logistic regression) with the goal of predicting whether a patient, represented by a given set of clinical variables (X), is likely to be misdiagnosed (i.e. a hard case) or not.

As usual, we need to first identify clinical variables of interest (X) that characterize our patients and help to predict our target variable y. The label y, in this setting, essentially represents the notion of hard case (y=1) versus easy case (y=0).

Further, a probabilistic classifier that produced well-calibrated conditional probability estimates, P(y=1|X), is highly desirable for this task; for instance, logistic regression and SVM with Platt scaling, etc., are among the candidate classifiers. If P(y=1|X) is high, we have a nice interpretation that tells us that the case is likely a hard case – and therefore, this patient would likely be misdiagnosed by the medical experts. By contrast, a low conditional probability suggests that the case is likely an easy case – and we have a relatively high confidence that this patient would be correctly diagnosed; by symmetry, this confidence level can be more intuitively quantified by 1 - P(y=1|X), the higher the value, the better the confidence (of X being an easy case).

For the purpose of training this baseline model, Dm and Dc are combined as a single training data set (D) where Dm corresponds to positive examples (y=1, hard case) while Dc corresponds to negative example's (y=0, easy case). The model parameters, i.e., coefficients of covariates in X (of a logistic regression), are estimated via, say, a 10-fold cross validation (CV) within D.

The next step is the model evaluation. Since the sample size between the two classes is skewed (with less misdiagnosed cases), we have a class imbalance problem, which suggests that a performance metric that trades off precision and recall is a better choice (see [A2]). AUC and accuracy for instance can be misleading in a class imbalance situation since we are more interested in capturing true positives than true negatives.

Following the model evaluation stage, we can then perform an error analysis by selecting different facets of the data (e.g. age brackets, gender, ethnicity) and observe their respective classifier performance. That is, we have an overall performance by estimated from within D, but additionally, we will compare how the model performs from within different facets of the data (e.g. male vs female) and this helps us identify the reasons behind misclassified cases and identify potential biases in our dataset: E.g. the data may contain more younger patients than old patients; the misdiagnosed cases may concentrate in certain age backets, gender or ethnicity. This information could give us valuable information as for what to do next. If, for instance, the data is intrinsically biased toward younger patients, then we may want to gather more data points for older patients.

From data preparation, model training and evaluation to error analysis, a ML task is an iterative process.

[A2]
Since we are dealing with a highly imbalanced dataset, a metric that trades off between precision and recall is a better option. For instance, using F1 score along with a precision-recall curve is a viable option. Alternatively, we could use a more general metric such as F-β that has the freedom of assigning a different weight to precision and recall.

In practice, I also often used "F-max" by identifying the probability threshold that leads to the maximum F score; F-max is essentially an F score under the optimal probability threshold (above which is positive and below which is negative).

To be more rigorous, I would compute sensitivity (recall), PPV (precision), F-measure within positive examples and specificity, NPV, F-measure within negative samples. Further, I'd find out these performance metrics within different subgroups (or facets) of the data, say different age groups, gender and ethnicity.

[A3] Model interpretation and data visualization are especially important for communicating the findings to non-technical audience. Using interpretable modeling approach is often more desirable in this setting. Logistic regression, for instance, has a nice interpretation for its coefficients; perhaps the log odds interpretation is too technical for laymen but roughly speaking, we can say that, the higher the magnitude, the more important that clinical variable is for predicting hard versus easy case. On the other hand, since the output, P(y=1|X), is a (approximately) well-calibrated class conditional probability, it can often be interpreted as a confidence level. To use simpler language, we can refer to a large P(y=1|X) as a risky case that medical experts tend to make mistakes; and a small P(y=1|X) as likely an easy case (see [A1] for more details).

The way we formulate the model also influences the interpretability: Instead of predicting the diagnosis itself (which leads to a multiclass classification problem), predicting a hard case versus easy case perhaps has a better advantage since the clients are likely more concerned about identifying hard cases associated with high risk of misdiagnosis, rather than the specific diagnoses for a patient. Of course, this depends on the audience and the clinical question they are interested in.

I'd probably incorporate a substantial amount of graphics to present the findings, including feature importance, performance metrics, identified biases in the data set (from error analysis) and so on. Shapley values for instance come with a nice model interpretation, when it comes to feature importance, through its property of additive importance and force plots, etc. This is to be demonstrated in the Jupyter notebook in the last question.

Model Deployment (10 points)

Suppose you have prototyped a machine learning model that successfully exceeds desired performance metrics, and now you are tasked with deploying the model into a production environment. A colleague suggests deploying the model onto AWS Lambda, a serverless compute platform that allows users to invoke functions on-demand. A dissenting colleague argues for deploying the model to a server running on AWS EC2, Amazon's cloud-hosted compute service.

1. Assuming one of Lambda and EC2 is the optimal solution, what information would you need to know about the model and its end users to decide the best option?
2. In general what best practices do you consider when deploying a model to a production environment?

[A1] I'd make a checklist examining the following criteria: use cases, scalability, cost/pricing, and performance comparisons among others.

In particular, is the service on-demand or expected to be constantly running? For an application that checks to see if a radiology report is reliable (and if not, subsequent actions are triggered such as sending the report back to the medical doctors for further investigations), I would image it is more an real-time-on-demand type of service and therefore, it may not need to be up and running all the time (but only when queries are sent, i.e. event trigger activated) – and probably requires constant updates in ML modeling as well (deployment is an iterative process to be discussed in [A2]) given that "diagnosing a disease" is a perpetual open question (and that's why even medical experts tend to make mistakes). If this is indeed the case, using Function-as-a-Service (FaaS) makes more sense, i.e. Amazon Lambda would likely be a better option.

Also, it's helpful to figure at which level of deployment (e.g. partial automation with human in the loop, shadow mode, or as an AI assistant) is best suited for a given application and from there, we can then decide which deployment solution to go. Does it require a high-performance setting (HPC) i.e. minimizing latency in response? What happens when the service crashes i.e. fast recovery? Or, does it involve a lot of real-time, on-demand endeavors such as modeling updates, data updates (using different versions of the features, new data transformation, data quality improvements, etc.).

[A2] Model deployment, like the ML model development, is an iterative process involving multiple stages: Continual monitoring, performance analysis and at times, going back to model training and error analysis (i.e. early stage of the ML pipeline) before deploying the new model again.

After developing a model with a given training dataset, it is often not sufficient to stop at achieving a good performance metrics in the test set. In particular, we need to consider the external dataset whose distribution could be quite different. This is in relation to i) concept drifts; i.e. the mapping from X to y changes and ii) data drifts i.e., X changes, and at times prior (class) distribution P(y) could also change. Data quality can also be an important factor in the deployment stage, because labeling inconsistency, bias, etc. can affect the model performance.

As shown in the first question/example indicates that there can be inter-reviewer variability in data annotation and very often, improving the labeling consistency itself will increase the model performance without changing the model itself (say, training the same neural network architecture but with a data set with less noises, and errors). Bias is yet another key factor that influences model performance; for instance, certain facets of the data may not be sufficiently represented (e.g. gender, ethnicity, social-economic status). In the case of disease analytics and

phenotyping, we may not have sufficient data for certain pathological conditions useful for drawing conclusions for diagnosing a disease (and hence impacted the model performance for predicting, say, the risk of this disease). Data augmentation by further gathering the data slices that has either low sample sizes or are error-prone, can help to train a more accurate and robust model. All of the these are in relation to the notion of data-centric modeling briefly mentioned earlier.

Having a reliable **ML pipeline** (as in the concept of MLOps) from data processing, model training and evaluation, error analysis to deployment, monitoring, registry, integration with other software system (so-called CI/CD practices) can help to facilitate a better iterative deployment cycle. At each stage of the pipeline, it's a good practice to consider model lineage and artifact tracking, including the versioning of the data, model artifacts (the output from one stage serving as the input to the next stage in an ML pipeline), code & hyperparamters, algorithms and frameworks, docker images, packages & libraries, among others. Maintaining a feature store is also important that speeds up model development and results sharing (e.g. various versions of feature representations, BERT word embedings trained on different datasets, different image feature maps for pre-training and fine-tuning practices).

Data Analysis (30 points)

The dataset 'radiology_costs.csv' has information associated with patients who had a knee MRI. This dataset contains the following information:

- age: patient's age,
- female: indicator if the patient is female,
- population: the population associated with the patient's home zip,
- hospital: binary feature that takes the value one if the patient was imaged at a hospital and zero if the patient was imaged at a community clinic
- upstream_costs: the yearly overall medical costs prior to the initial knee MRI
- downstream_costs: the yearly overall medical costs that follow the initial knee MRI
- er: indicates if the image took place at an Emergency Room (ER) visit

You can answer either of the following questions:

- **Option A (Inference):** What features drive higher downstream costs? What assumptions did you make to answer this question? What are the limitations of your analysis?

- **Option B (Prediction):** Build a model to predict downstream costs. How do you assess your model's predictive performance? What assumptions did you make when building your prediction model? What are the limitations of your analysis?

Summarise your findings in a short paragraph (less than 200 words). Make sure to include assumptions, limitations and any patterns that you have observed in the data.

[Solution]

I shall focus on Option A although I have also implemented a (nearly) complete baseline solution for Option B since both A and B are interrelated.

Specially,

An Jupyter notebook (Covera - Radiology Costs.ipynb) was created to illustrate all the key steps including relevant thought process leading up to the related demo. Some details of the implementations are abstracted away through supporting modules, including:

data_pipeline.py: handles all the data processing routines including the creation of training data.

utils.py: contains useful helper functions in general
model.py: includes related functions for model training, hyperparameter tuning, model evaluation, etc.

The notebook is structured as follows:

1. Exploratory data analysis (EDA)
2. Model training and evaluation
3. Analyze feature importance via SHAP
4.

The demo in the notebook was tested on **Google Colab**. Although it can be run locally, I'd recommend running it on the Colab to minimize library dependency issues (particularly the shap library).

A common way to identity the features that influence the target variable (downstream cost), is to conduct an association study by finding out the correlations between candidate features and the target. In this case, plotting the correlation heatmap would be a good option for both the analysis and visualization.

In this demo, however, I opted for Shapley values, or more precisely SHAP (SHapley Additive exPlanations), as a more scalable implementation for Shapley values. SHAP values, however, are computationally expensive to compute for general black-box models; in the case of trees and forests there exists a fast polynomial-time algorithm (see the TreeShap paper, or "Consistent Individualized Feature Attribution for Tree Ensembles", by Scott M. Lundberg et al. for more details).

For this reason, I decided to use a tree-based regression model such as random forest regression for predictive purposes in addition to potentially identifying non-linear relationships between the candidate covariates and the downstream cost. Due to the property of the "additive feature importance" that SHAP is able to capture, it also helps us identify the feature subset that drives the cost:

If the variables are positively correlated to the cost (e.g. being imaged at a hospital, being a female, older patients), then by increasing their values, we'll expect to observe a higher downstream cost; by contrast, if the variables are negatively correlated to the cost, then increasing their values will drive down the cost (although I did not observe such variables in this dataset). On the other hand, there are variables that do not seem to contribute to either positive or negative responses (e.g. population) and as a result, their feature importance will be lowest. From the perspective of feature selection, it may be of interest to drop such variables to increase model performance (and alleviate overfitting and increasing training efficiency, etc.)

SHAP can also assign feature importance correctly even under the presence of correlated features. A likely case would be 'er' and 'hospital', although this demo does not cover the analysis of identifying the correlations among the covariates.

Below include the key observations regarding which features drive the cost:

We can view feature importance both globally and locally (individually); that is, there are features that are generally important in driving the cost but in the meantime, we can also observe how each feature contributes to individual prediction on a per-patient basis.

1a. In terms of the global feature importance, 'hospital', 'female' and 'age' are the top 3 most important features.

'upstream_costs' and 'er' also moderately impact the cost.

'population' however does not provide a strong signal and does not seem to be correlated to the cost at least in an average sense. See the notebook for more details; also see the following files for example results from the summary plot:

feature-importance-bar.tif
summary_plot-global.tif

1b. In terms of the local feature importance, we can use a force plot to observe which features positively drives the downstream cost while the other negatively drives the cost. Two example cases are shown in the notebook/demo. The individual trends may be quite different than the global trend in that the global feature importance is computed in the average sense (on average, by removing a specific feature from the feature set, how much performance differential do we observe). In the force plot, we observe that for patient #1, being a female with the image taken in the emergency room in a hospital setting pushes the cost higher, while her age (relatively younger) drives down the cost.

I randomly selected 50 patients for the analysis of local interpretability and demonstrated two of them.

2. The dependence plot, on the other hand, shows what the marginal effect of features has on the predicted outcome of the random forest regression. It tells us whether the relationship between the target and a feature is linear, monotonic or more complex.

‘age’ for example is generally positively correlated with the cost

‘population’, on the other hand, exhibits no apparent correlation (neither positively nor negatively) with the cost. This could also be due to the fact that the zip code was not cleaned. According to the EDA section, there might be a mixture of standards present for the zip code, which would then distort the signal in its association with the cost.

Please refer to the notebook for other observations.