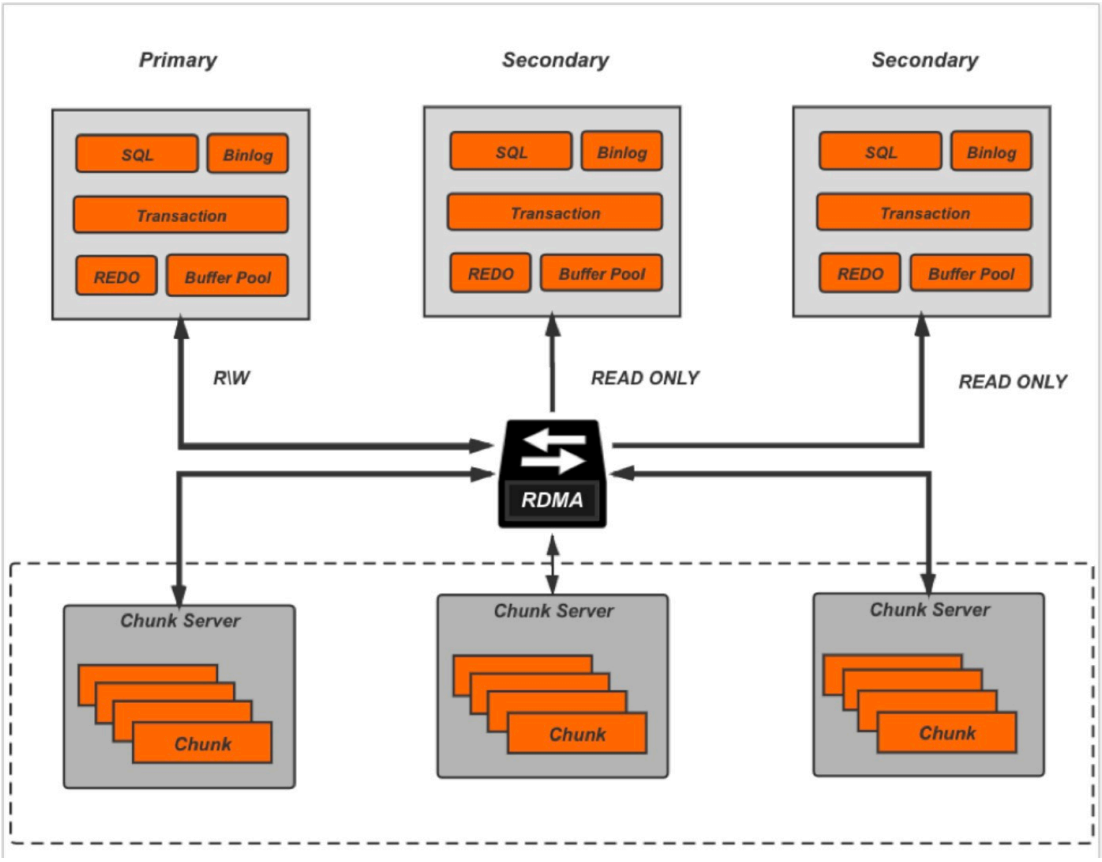
Name: Xin Peng Yuchen Zhao
Date: 12/08/2022
NYU NetID: xp2083, yz8759
Section: 3
Total in points: Professor's Comments:

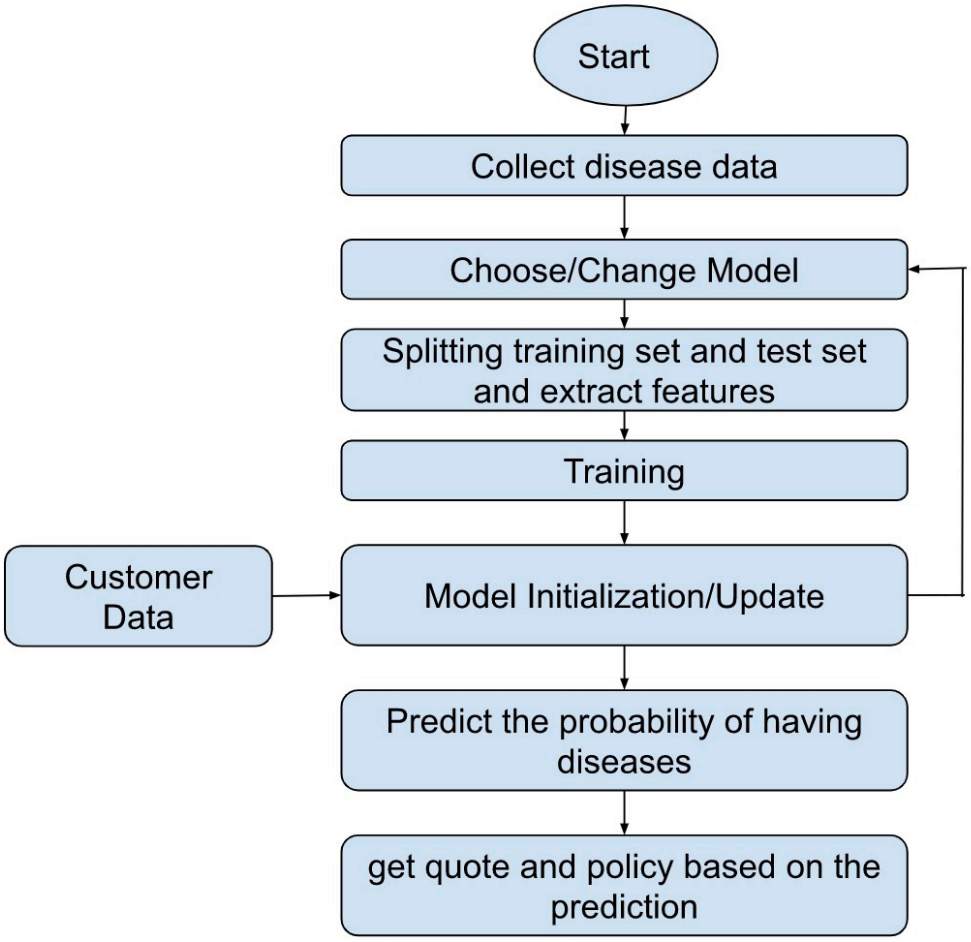
Project 3
Affirmation of my Independent Effort: Xin Peng, Yuchen Zhao

1. about the physical database design

For the index at the logical level, we add a primary key to every table in the data warehouse. Then Aliyun Dataworks will automatically build indexes using the primary key for each table.
For partitioning at the logical level, we add a field called “Date” for each table in the data warehouse. Then Aliyun Dataworks will automatically create one date partition for each date and in each date partition stores all records whose Date field is the same as the date partition.
At the underlying structure of Aliyun Dataworks, it mainly uses the PolarDB. See the figure below for its structure. PolarDB uses Share Storage as overall architecture. In a cluster, there are many Chunk Servers which are connected by high-speed internet and controlled by a Remote Direct Memory Access(RDMA) module to provide service for upper calculation nodes. A cluster can support one Primary node and multiple Secondary nodes. The Primary node can perform read and write operations whereas the Secondary node can only perform read operations. The Primary node and Secondary nodes are linked to Chunk Servers by RDMA [3].
The calculation nodes are connected to PolarStores by libpfs. The data are split into Chunk and distributed to related Chunk Server by local PolarSwitch. Each Chunk Server maintains a group of Chunk copies and maintains the consistency of copies by ParallelRaft. PolarCtr is in charge of maintaining and updating the metadata of the whole cluster [3].



2.The pipeline that enables a customer to obtain an insurance quote and a policy.



Before construction of the ML model, assume there exists some rules of predicting a customer's probability of getting a disease. The rules may be the result of market research or past experiences.

Then we start to construct a ML model.

(1) collect some disease data from the internet.

(2) based on the data distributions and the prediction result we want to get, decide which predictive model to use.

(3) extract features from the data and select useful features when there are many features.

(4) perform model training.

(1-4) is a loop. With more and more collected data coming in, there may be the need to change the model. There may be more features. And the model will be updated continually.

(5) When there is a new customer waiting for quotes/old customers needing to change policy, the ML model will give a result of the probability of the customer getting the disease.

(6) A calculation will then be performed to provide the suggested quote and policy based on the result.

After a certain number of collected data and training examples, the model will be updated to a satisfying status so that it can provide accurate predictions. Using the ML model result is more accurate than the initial rules. It will give more accurate quotes and policies so that customers can see whether they can buy insurance for a specific disease and improve customer experience. Also, insurance sellers can decide whether they can create a disease policy and how much should the quote be for a customer and avoid paying large amounts of compensation and improve organizational excellence.

3. Machine Learning Model Creation

3.1 Collect disease data

Here we use the diabetes Dataset from Kaggle [4]. It has 768 body examination records. For each record, there are 8 features. The features include Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction and Age. The goal is to predict whether people will have diabetes.

3.2 Choose model

As there is a certain goal, it is suitable to use a supervised ML model. As the goal is to predict whether people will have diabetes, here we should choose classification models instead of regression models.
As there are few data and features, it is better to use shallow models instead of deep models. Because deep models need a lot of data to make sure it can converge.
As the features provided have relatively explicit and linear relations with diabetes, Logistic Regression [5] will be a suitable model to do this job and it was widely used in the healthcare field, such as TRISS [1], colonic peritonitis [2], and so on.

3.3 split training set and test set

Use 90% of records as training set, and the other 10% of records as test set

3.4 Extract features

To better capture the data's relation with diabetes, we use three kinds of features including original features, one-hot features and statistical features.

(1) original features
Use the original features in the data set.
Use data standardization technique to make sure each original feature is transformed into values whose average value is 0 and standard deviation is 1.

The standardization equation is:

x= (x-)/

The final original features for training set include:

feature_Preg_orig_train_f (standardized Pregnancies values),

feature_Glucose_orig_train_f (standardized Glucose values),

feature_Blood_orig_train_f (standardized BloodPressure values),

feature_Skin_orig_train_f (standardized SkinThickness values),

feature_Insulin_orig_train_f (standardized Insulin values),

feature_BMI_orig_train_f (standardized BMI values),

feature_Func_orig_train_f (DiabetesPedigreeFunction value which is already in standardized values)

It should be noted that during the standardized procedure, for each feature, the average value and standard deviation value used will be calculated by training data only. This is because if it is calculated including test data, it will make test data patterns involved in the training data. This situation needs to be avoided because it will make the test data predictions more accurate than it should be which will lead to biased models. Therefore, the standardization of both training set and test set only use the average value and standard deviation value calculated from the training set.

Note that the standardization of the feature is used to make sure that the model can converge.

The original features for the training set:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	0.650112	0.853433	0.162026	0.908410	-0.690624	0.218511	0.627
1	-0.838381	-1.097649	-0.147541	0.533732	-0.690624	-0.661245	0.351
2	1.245509	1.937367	-0.250729	-1.277215	-0.690624	-1.075987	0.672
3	-0.838381	-0.973771	-0.147541	0.159053	0.125941	-0.472726	0.167
4	-1.136080	0.512768	-1.488995	0.908410	0.768768	1.412466	2.288
...	...	...	...	...	...	...	...
686	-0.242984	0.295981	-0.250729	-1.277215	-0.690624	-1.101123	0.314
687	-0.838381	-0.416319	-0.973051	-0.090733	-0.690624	-0.447590	0.181
688	-0.838381	0.605677	0.265215	0.346392	0.873011	-0.975443	0.828
689	-0.838381	0.729555	0.677970	1.595321	0.873011	1.789504	0.335
690	1.245509	-0.416319	0.574781	-1.277215	-0.690624	-0.912604	0.856

(2) one-hot features

First, map age to different age gaps at the gap of 10. Concretely, ages from 0 to 10 are mapped to 1, ages from 11 to 20 are mapped to 2, …, and ages from 91 to 100 are mapped to 10. All other ages are mapped to 11. Secondly, based on the mapped age, transform it to one-hot features.

The final one-hot feature for the training set is train_data_age.

	age_3	age_4	age_5	age_6	age_7	age_8	age_9
0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
2	0	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...
686	1	0	0	0	0	0	0
687	1	0	0	0	0	0	0
688	1	0	0	0	0	0	0
689	0	0	1	0	0	0	0
690	0	1	0	0	0	0	0

(3) statistical features

The basic idea is to calculate for each age gap, what is the probability that people have diabetes. Then for each record, check its age gap, and use the relative diabetes probability of this age gap as the feature.

The probability equation is:

p(gap)=no of patients in the gap/no of all patients in training set

As the probability lies within 0-1, it needs not to be standardized.

The final statistical feature for the training set is train_age_prob.

0	5	0.073806
1	4	0.099855
2	4	0.099855
3	3	0.120116
4	4	0.099855
...	...	...
686	3	0.120116
687	3	0.120116
688	3	0.120116
689	5	0.073806
690	4	0.099855

3.5 Training of the model

Use the training set to train a logistic regression model and check the prediction accuracy using the test set.
Check each feature's importance using model.coef_ values. It turns out that one-hot values of ages contribute little to the model. Then take them off from the features.
At last, the model achieves an accuracy of 0.78 in the training set and accuracy of 0.81 in the test set.

4. Provide quote suggestions

Associates need quote suggestions to help their decisions on premiums and policies. Our database stores huge amounts of personal data and claims, which can be a good source of information. The quote will be provided based on the result of our predictive model. In this part, we simply define a linear computation to represent the relation that the higher risk a person may have diabetes, the higher premium he will pay. However, in reality, this relationship will be acquired by data mining and analyzing.
The mock relationship is as below:

$$\text{Premium} = 1000 * \text{Prob} + 10000$$

5. Model updating

By training and testing, we can provide relatively precise predictions based on current datasets. However, new data is continuously added to the database in everyday activities, and the market can change quickly due to inflation, COVID and so on. That's why we need to continuously update the predictive model and keep giving the most accurate and up-to-date predictions. In order to make changes easily to present, we update our model every time new data comes in. The retraining procedure is presented in part 2 and the model will be updated instantly.

Reference

[1] BOYD, CARL R. M.D., F.A.C.S.; TOLSON, MARY ANN R.N., M.S.N., CCRN; COPES, WAYNE S. PH.D.. Evaluating Trauma Care: The TRISS Method. The Journal of Trauma: Injury, Infection, and Critical Care: April 1987 - Volume 27 - Issue 4 - p 370-378

[2] Biondo, Sebastiano MDa,*; Ramos, Emilio MDa; Deiros, Manuel MDa; Ragué, Juan Martí MDa; De Oca, Javier MDa; Moreno, Pablo MDa; Farran, Leandre MDa; Jaurrieta, Eduardo MDa. Prognostic Factors for Mortality in Left Colonic Peritonitis: A New Scoring System. Journal of the American College of Surgeons: December 2000 - Volume 191 - Issue 6 - p 635-642 doi: 10.1016/S1072-7515(00)00758-4

[3] https://help.aliyun.com/document_detail/426488.html

[4] https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset

[5] https://en.wikipedia.org/wiki/Logistic_regression