Cardiovascular Disease Data Analysis

Problem Statement

Cases of cardiovascular disease (CVD) have been on the rise throughout the world, from well-developed countries to third world countries. 17.9 million people die each year from CVDs, which make up 31% of all deaths worldwide. Of those deaths, 85% are from heart attacks and strokes. As a result, CVDs are the number one cause of death globally. Heart attacks ($12.1 billion) and Coronary Heart Disease ($9.0 billion) were 2 of the 10 most expensive conditions treated in US hospitals. Accordingly, by 2030, medical costs of Coronary Heart Disease are projected to increase by about 100%.

Our prevention plan is a mobile app, which will enable people to evaluate the possible risk of having CVD, has the potential to save cost expenses and reduce mortality rates. If an individual is informed that it's likely to be diagnosed with CVD in the future, he/she could have some kind of early intervention, such as medical treatment or a lifestyle and dietary change, to stop the condition from deteriorating. In addition, the whole evaluation process can be performed at home with the mobile app so that transportation and hospitalization costs can be reduced.

Solution

Data used

We decided to use the dataset showing the replication data for the cardiovascular disease in the SMARThealth Extend project. This dataset contains more than 20,000 records of individuals, distributed by their gender, age, BMI (body mass index), blood pressure, location, education, marital status, and other factors that affect whether they are at risk for being diagnosed with CVD (column named "highrisk").

age	sex	education	marital_status	occupation	sbp_avg	dbp_avg	bg_mgdl	bmi	smoking	village	areas	cvdrisk	highrisk	bplt	lltt	aptt
65	Female	primary	married	self-employed	140.5	78.5	156	18.902	nonsmoker	sepanjang	rural	<10%	No	No	No	No
60	Male	primary	married	self-employed	156	108.5	113	25.4767	smoker	sepanjang	rural	clinicall high risk	Yes	No	No	No
87	Male	primary	married	not working	153.5	77	91		pastsmoker	majangtengah	rural	20-30%	Yes	No	No	No
82	Female	primary	married	not working	152	76	114	23.8914	nonsmoker	majangtengah	rural	20-30%	Yes	No	No	No
55	Female	primary	widowed	casual worker	179	94	130	29.4887	nonsmoker	majangtengah	rural	clinicall high risk	Yes	No	No	No
100	Male	primary	widowed	not working	191	101	168	20.2694	pastsmoker	sepanjang	rural	clinicall high risk	Yes	No	No	No
93	Female	primary	widowed	not working	143.5	101	102	15.7651	nonsmoker	sepanjang	rural	clinicall high risk	Yes	Yes	No	No
90	Female	primary	widowed	not working	161	80.5	141	27.5862	nonsmoker	sepanjang	rural	clinicall high risk	Yes	No	No	No

Figure 1

Approach and assumptions

We assumed that the risk of having CVD can be predicted by some measurable metrics. With those metrics, we can predict if one is likely to be diagnosed with CVD with high accuracy.

Feature Engineering

Feature selection: We dropped the columns named “bplt”, “lltt” and “aptt”, which are boolean variables indicating whether the patient has follow-up treatments. Those attributes were neither self-explanatory nor aligned with our aim to save future costs for patients.

Imbalance Issue: We balanced our dataset because only 6453 rows out of 22093 rows were labeled “Yes”. After balancing the data, there were 6500 rows for “No” and 6453 rows for “Yes”.

```
# Separate majority and minority classes
df_alldata_majority = df_alldata[df_alldata.highrisk=="No"]
df_alldata_minority = df_alldata[df_alldata.highrisk=="Yes"]
```

```
# Downsample majority class
df_alldata_majority_downsampled = resample(df_alldata_majority,
                                 replace=False,    # sample without replacement
                                 n_samples=6500) # reproducible results
# Combine minority class with downsampled majority class
df_alldata_downsampled = pd.concat([df_alldata_majority_downsampled, df_alldata_minority])
# Display new class counts
df_alldata_downsampled.highrisk.value_counts()
```

```
No     6500
Yes    6453
Name: highrisk, dtype: int64
```

Figure 2

Application of Darwin

In this project, we followed the example of Darwin Supervised Classification. The process given in the example supported our data perfectly. Darwin helped us handle our missing data, transform categorical variables into numeric variables, and build and analyze the final model (XGB Classifier Model). The final model provided an excellent test result:

	precision	recall	f1-score	support
No	0.89	0.99	0.93	1293
Yes	0.99	0.88	0.93	1298
avg / total	0.94	0.93	0.93	2591

Figure 3

After working with Darwin, we found that the biggest advantage of using Darwin was how it simplifies the processes for machine learning. Moreover, Darwin has an indication for each step, making it easy to fully understand what Darwin does. Based on the example above, our favorite part of working with Darwin was the process of building and analyzing the model, because it handles model selection, the most challenging part of machine learning.

Darwin is helpful in many aspects, and the only thing we wish Darwin included is a set of improved functions to balance data. When we built our first model, the imbalanced data caused a low precision and recall value, and we had to import scikit-learn to handle this problem.

Team Engagement

The members of our team are Jiayan Han, Shimin Zhang, Bo Yu, and Hassan Sheikh. Based on our objectives for completing the project, we divided our work into coding and delivering. One objective was to use Darwin and other data analysis tools to present the data. The other was to finish up the technical report for this project:

Coding:
Jianyan Han - Dataset Research
Bo Yu - Feature Engineering
Shimin Zhang - Modeling
Delivering:
Hassan Sheikh - Report Writing

General Challenges

One of the challenges we faced was handling the imbalanced data. The imbalanced data caused a low precision and recall value for our first few models. To overcome that problem, we balanced the data using Pandas Package first and then passed the data into Darwin. Another challenge we had was handling the feature selection. We found that diastolic blood pressure and systolic blood pressure were dominant in all the features, so we tried out using only those two features to predict the risk, but the result turned out different than what we expected. We also tried to drop those two features and only use the rest of the features, but the result was decent. Below is the result for the training dataset:

	precision	recall	f1-score	support
No	0.67	0.61	0.64	5231
Yes	0.64	0.70	0.67	5131
avg / total	0.66	0.65	0.65	10362

Figure 4

However, the standard for a valid prediction in the healthcare industry is higher than in other industries. Under this circumstance, those two features were kept because we believed this was the model that would generate the most accurate result.

Next Steps

We plan to create an app, which will help people predict the intensity of their CVD risk based on the features in our model: whether they have low risk, medium risk, or high risk of CVDs. We also might consider doing the same for calculating one's BMI based on the factors that affect it in terms of their gender, height, weight, and age. With all these measurements, we can predict one's risk of having CVD with 93% accuracy. Since the diastolic blood pressure and systolic blood pressure are the most important features in our model, for business purposes, the advertisement of sphygmomanometers can also be included in the app, after showing the prediction for CVD risk. Below is a simple UI for the App:

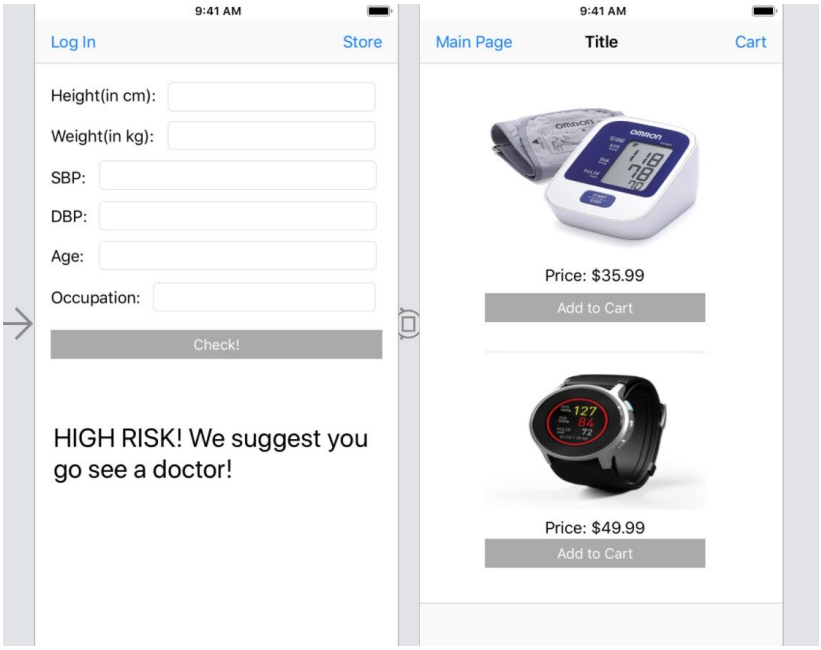

Figure 5