پاسخ تمرین سری ۱

محمدرضا عزیزی
۹۸۱۳۱۰۲۲
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر (پلی‌تکنیک تهران)
mrazizi@aut.ac.ir

۱ راهکاری برای پیش‌پردازش داده‌ها

داده‌ها با استفاده از کتابخانه pandas بارگذاری کرده و از آن‌جایی که ستون‌های داده‌ها نام ندارد، ابتدا با توجه به توضیحات دیتاست، یک لیست برای نام ستون‌ها تعریف کرده و این نام‌ها را به ستون‌های دیتافریم بارگذاری شده اضافه می‌کنیم. لیست این نام‌ها به ترتیب برابر است با:

```
["age", "workclass", "fnlwgt", "education", "education-num",
"marital-status", "occupation", "relationship", "race", "sex",
"capital-gain", "capital-loss", "hours-per-week", "native-country", "label"]
```

ستون آخر که مربوط به برچسب داده‌است با با نام label نام‌گذاری کرده‌ایم.

راه حلی که در ابتدا ممکن است به ذهن برسد این است که یک دیکشنری تعریف کنیم که هر دوتایی کلید/مقدار آن، خود یک دیکشنری است. برای مثال به ازای کلید label، دیکشنری زیر را داریم:

```
'label': {'<=50K':0, '>50K': 1}
```

به ازای تمامی مقادیر تمامی ستون‌هایی که مقدار عددی ندارند، این دیکشنری را تعریف کرده و به هر مقدار اسمی، یک عدد نسبت دهیم. این اعداد از برای هر متغیر از ۰ شروع شده و یک واحد یک واحد افزایش می‌یابد.

در نهایت با استفاده از تابع replace از کتابخانه pandas، طبق دیکشنری تعریف شده، مقادیر اسمی را به مقادیر عددی تبدیل کنیم.

در ابتدا روش ذکر شده را پیاده‌سازی کردیم و بارگذاری داده به صورت زیر بود:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	label
0	39	5	77516	0	13	2	8	3	0	1	2174	0	40	0	0
1	50	1	83311	0	13	0	4	2	0	1	0	0	13	0	0
2	38	0	215646	3	9	1	6	3	0	1	0	0	40	0	0
3	53	0	234721	2	7	0	6	2	4	1	0	0	40	0	0
4	28	0	338409	0	13	0	5	0	4	0	0	0	40	12	0

اما این روش کدگذاری داده‌های اسمی یک مشکل بزرگ دارد. برای مثال به کدگذاری متغیر relationship دقت کنید:

```
'relationship': {'Wife':0, 'Own-child':1, 'Husband':2, 'Not-in-family':3,
                 'Other-relative':4, 'Unmarried':5},
```

در این حالت به یک فرد که ازدواج نکرده است عدد ۵ نسبت داده می‌شود. به یک فرد که نقش شوهر دارد عدد ۲ و به فردی که نقش زن دارد، عدد ۰، در حالی که به وضوح در دیتاست این مساله، هیچ تناسبی بین این افراد وجود ندارد. یعنی رابطه‌ای از این جهت که فرد ازدواج نکرده فاصله عددش با زن یا شوهر چقدر باید باشد، وجود ندارد.

بنابراین از ادامه دادن مساله با این روش منصرف شده و در ادامه به سراغ روش One hot encoding می‌رویم.

قبل از کدگذاری با استفاده از این روش، باید راه‌حلی برای داده‌های گم‌شده پیدا کنیم. در این مساله، ما برای داده‌های عددی، میانگین هر ستون را جایگزین مقدار گم‌شده کرده و برای داده‌های اسمی، مقداری که بیشترین تکرار را دارد جایگزین مقدار گم‌شده می‌کنیم. این کار با استفاده از کلاس DataFrameImputer انجام شده‌است.

می‌دانیم که one hot encoding به این صورت است که به ازای هر مقدار یک متغیر، یک ستون جدید ایجاد می‌شود و ردیف‌هایی که آن مقدار را دارند، در آن ستون ۱ و در دیگر ستون‌ها مقدار ۰ خواهند گرفت. دو تابع one_hot_input و one_hot_output را برای تبدیل مقادیر ورودی و خروجی شبکه عصبی تعریف کرده‌ایم که در هر یک OneHotEncoder کتابخانه sklearn استفاده شده است.

۲ بارگذاری داده‌ها و انجام پیش‌پردازش

پس از بارگذاری داده‌ها و انجام پیش‌پردازش‌های ذکرشده در بخش ۱، ستون آخر را به عنوان label در نظر گرفته و بقیه ستون‌ها را به عنوان ورودی شبکه عصبی در نظر می‌گیریم.

ابعاد ورودی و خروجی نهایی ما به صورت زیر خواهد بود:

```
x_train_enc shape: (32561, 22141)
y_train_enc shape: (32561,)
```

مجموعا ۳۲۵۶۱ داده داریم (که در آینده به عنوان داده آموزش و ارزیابی استفاده خواهد شد) و هر داده، ۲۲۱۴۱ بعد (یا ویژگی) دارد.

۳ طراحی مدل