

## گزارش کار پروژه کامپیوتری ۴

فاطمه کرمی محمدی | ۸۱۰۱۰۰۲۵۶

بخش اول:

(۱-۱) مانند تمرین قبل Mapset ساخته شده به این شکل است:

char_cell													
2x32 cell													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	a	b	c	d	e	f	g	h	i	j	k	l	m
2	00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010	01011	01100

...

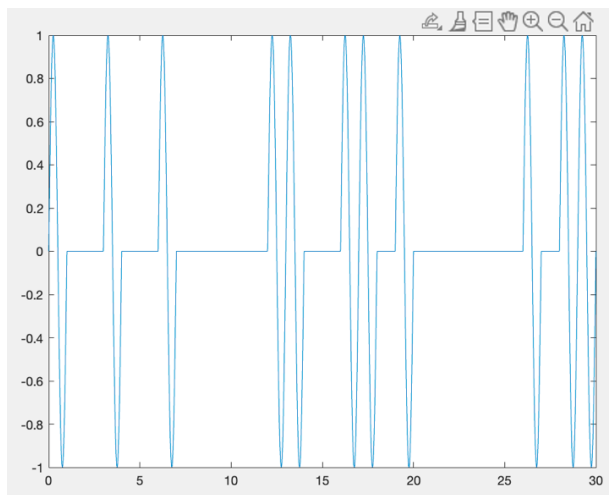
(۲-۱) این تابع ابتدا با استفاده از تابع `make_binary` رشته ورودی را کدگذاری می‌کند و سپس با توجه به `bit rate` ورودی بیت‌های کد باینری را تبدیل به تابع سینوسی کرده و به سیگنال خروجی اضافه می‌کند. فرمول تبدیل بیت‌ها به تابع سینوسی به صورت زیر است:

$$\text{Signal} = \text{sig\_dom} * \sin(2 * \pi * t)$$

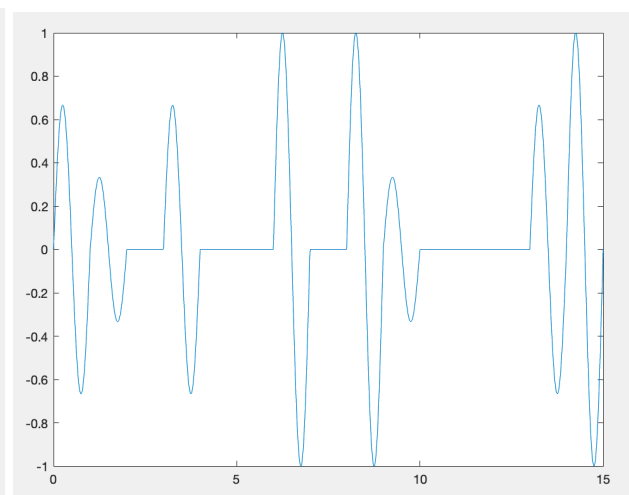
$$\text{Sig\_dom} = \text{bin2dec}(\text{bin\_str}) / (\text{pow}(2, \text{rate}) - 1)$$

`bin_str` همان بخش کد باینری است که طول آن برابر `bit rate` است. پس از تکرار این فرمول و ساخت بخش‌های سیگنال با تابع سینوسی و ادغام آنها خروجی (پیام کدگذاری شده) ساخته می‌شود.

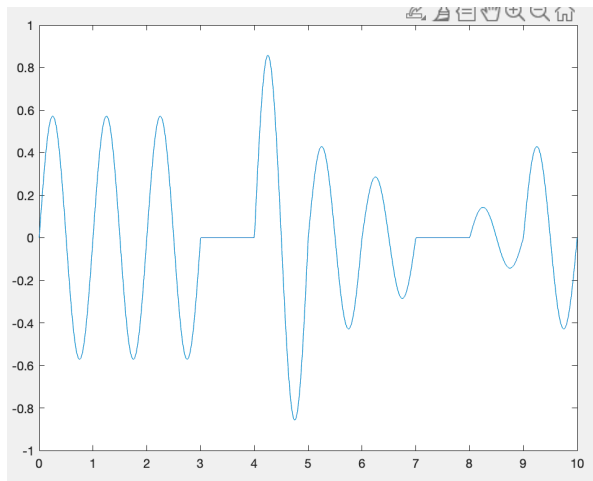
(۳-۱) خروجی تابع `coding_amp`



Bit rate = 1



Bit rate = 2



Bit rate = 3

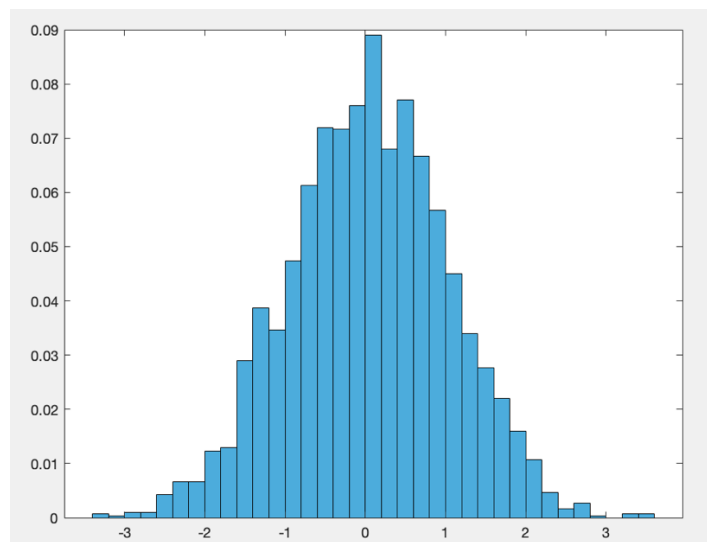
۴-۱) این تابع در یک لوپ هر دفعه بخشی از سیگنال ورودی به طول یک ثانیه را جداسازی کرده و correlation آن را با تابع  $2 \cdot \sin(2 \cdot \pi \cdot t)$  محاسبه می‌کند. سپس براساس تعداد  $\text{bit rate}$  یک مقدار error که برابر اولین مقدار  $\text{threshold}$  است ساخته  $(1/(2 \cdot (\text{bit\_rate}^2 - 1)))$  و مقدار correlation را با اعداد مورد انتظار  $(j/(\text{bit\_rate}^2 - 1))$  ,  $j = 1, 2, \dots, (\text{bit\_rate}^2 - 1)$  مقایسه کرده و فاصله آن با هر کدام از این اعداد از error کمتر بود، عدد باینری بخش مورد نظر را برابر  $\text{dec2bin}(j)$  (bit rate) در نظر می‌گیرد. در نهایت عدد باینری کامل شده را با استفاده از تابع  $\text{make\_text}$  به پیام اصلی تبدیل می‌کند.

خروجی این قسمت برای ورودی سیگنال کدگذاری شده  $\text{signal}$ :

`message =`

`'signal'`

۵-۱) با دستور  $\text{histogram}$  نمودار PDE سیگنال  $\text{randn}(1, 3000)$  به این شکل است:



مشخص است که این نمودار گوسی است.

همچنین با چاپ میانگین و واریانس (std و mean) سیگنال rand خروجی به این شکل است:

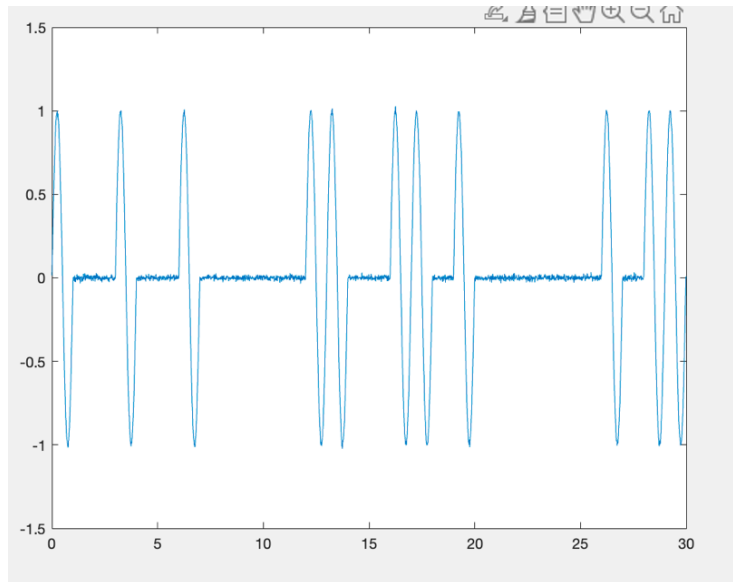
**mean: 0.040399, std: 0.990996**

مقدار mean بسیار نزدیک به ۰ و std بسیار نزدیک به ۱ است. (اگر تعداد اعداد تولید شده بیشتر شود میانگین و واریانس بیشتر به ۰ و ۱ نزدیک می‌شوند.)

۶-۱) برای این کار عدد ۰.۰۱ در سیگنال نویز تولید شده ضرب می‌شود. با اضافه کردن آن به سیگنال کدگذاری شده خروجی decoding به این صورت است:

```
noisy_message =  
'signal'
```

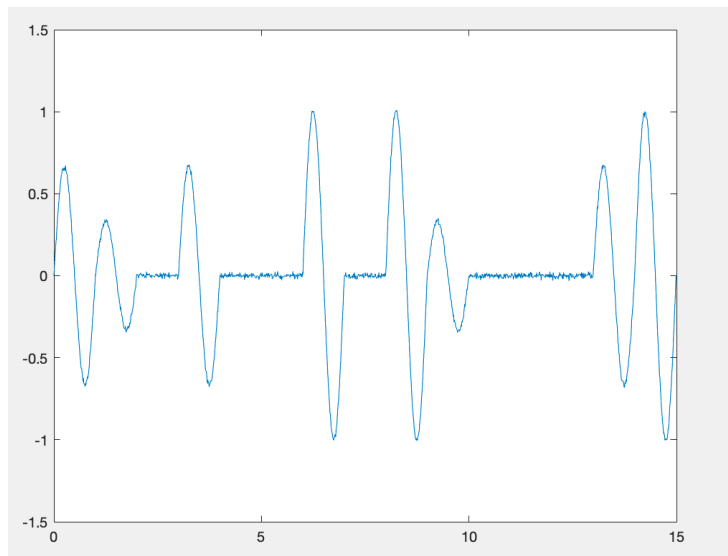
نتیجه برای ۱ = bit rate



سیگنال نویزی ۱ = Bit rate

```
noisy_message =  
'signal'
```

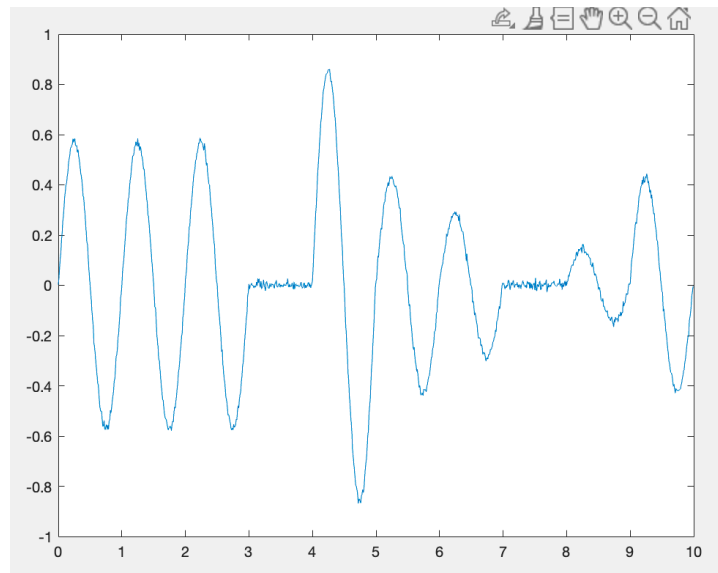
نتیجه برای ۲ = bit rate



سیگنال نویزی ۲ = Bit rate

```
noisy_message =  
'signal'
```

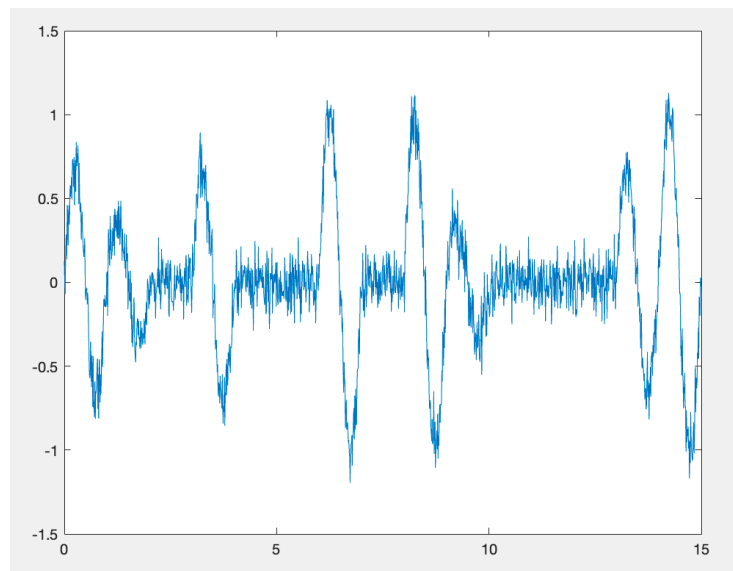
نتیجه برای  $\text{bit rate} = 3$



سیگنال نویزی  $\text{Bit rate} = 3$

در این بخش چون قدرت نویز کم بوده تاثیر خاصی در سیگنال اصلی نداشته و همچنان جواب دیکود درست است. (۷-۱) با تغییر قدرت نویز به ۰.۱ همچنان این نویز تاثیر خاصی روی دیکود کردن نمی گذارد. برای مثال تصویر سیگنال نویزی شده با  $\text{bit rate} = 2$  و دیکود شده آن به این شکل است:

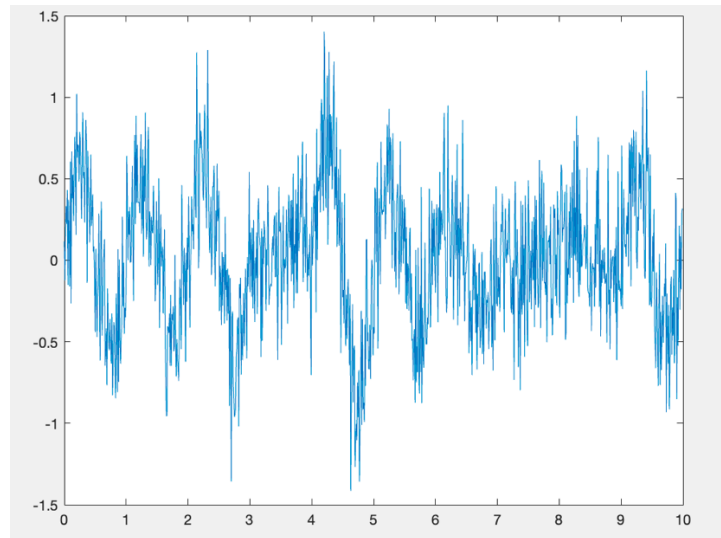
```
noisy_message =  
'signal'
```



با این قدرت نویز همچنان دیکود کردن با همه  $\text{bit rate}$  ها جواب درست می دهد.

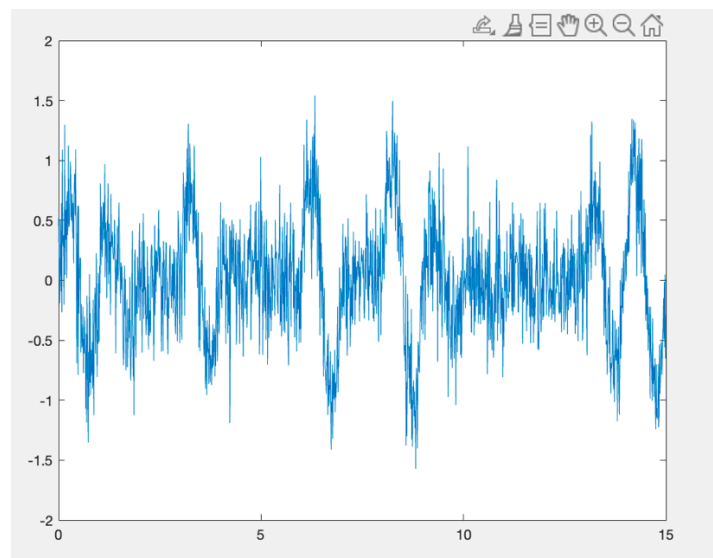
حال با افزایش قدرت نویز به ۰.۳ دیگر دیکود کردن سیگنال با  $\text{bit rate} = 3$  همواره پاسخ درستی نخواهد داد:

```
noisy_message =  
'ryonal'
```



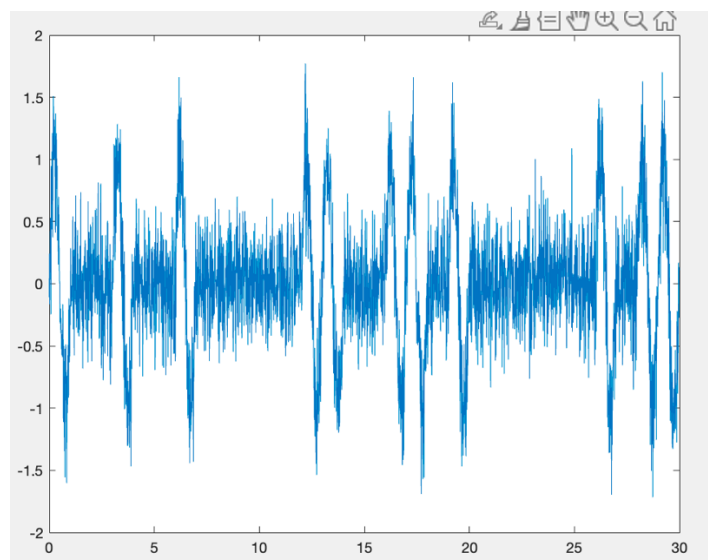
اما با  $\text{bit rate} = 2, 1$  همچنان پاسخ سیگنال در اکثر مواقع درست خواهد بود:

```
noisy_message =  
'signal'|
```



Bit rate = 2

```
noisy_message =  
'signal'
```



Bit rate = 1

نتیجتاً  $\text{bit rate}$  های کمتر مقاومت بیشتری نسبت به نویز داشتند. به این دلیل که هرچه  $\text{bit rate}$  بالاتر برود تعداد بیت‌های ارسالی در ثانیه بیشتر خواهد شد و در نتیجه تعداد  $\text{threshold}$  ها برای تشخیص در دیکودینگ بیشتر می‌شود و فاصله آنها کمتر خواهد شد. حال با اضافه کردن نویز اگر قدرت نویز به اندازه کافی زیاد باشد می‌تواند سیگنال اصلی را به قدری تغییر دهد که در بخش دیکودینگ بخش مورد نظر از سیگنال از  $\text{threshold}$  درست رد شده و تشخیص به اشتباه دچار شود.

(۸-۱)

برای  $\text{bit rate} = 3$ : ضریب ۰.۲۵ (واریانس ۰.۰۶۲۵)

برای  $\text{bit rate} = 2$ : ضریب ۰.۵ (واریانس ۰.۲۵)

برای  $\text{bit rate} = 1$ : ضریب ۱.۲ (واریانس ۱.۴۴)

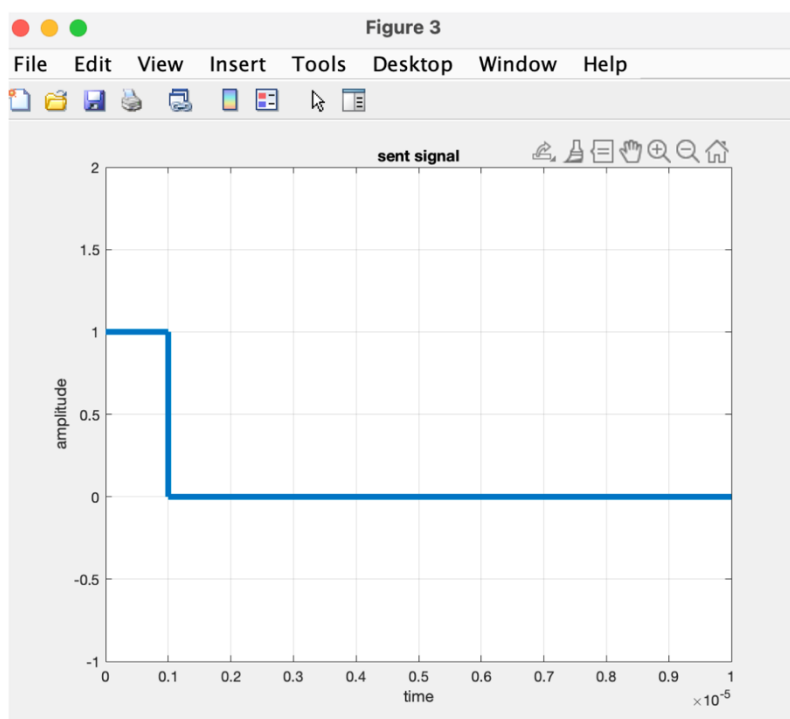
(۱۰-۱) اگر نویز نباشد  $\text{bit rate}$  تا حدود ۷ می‌تواند افزایش یابد.

(۱۱-۱) خیر. ضریب ۱۰ در  $\text{correlation}$  گرفتن صرفاً ضریب سیگنال نویز دار را افزایش می‌دهد و این ضریب در مقدار نویز هم ضرب می‌شود. بنابراین تغییری در نتیجه نخواهد داد.

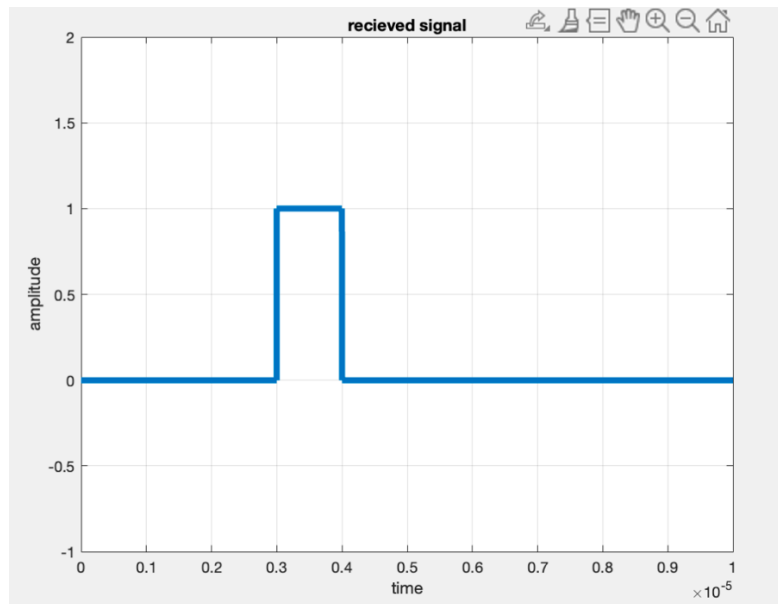
(۱۲-۱) سرعت اینترنت  $\text{adsl}$  خانگی بین ۱ تا ۱۶ مگابیت بر ثانیه است. در این تمرین اطلاعات با سرعت ۱ و ۲ و ۳ بیت بر ثانیه ارسال شده است. (حدوداً یک میلیونم سرعت  $\text{adsl}$ )

**بخش دوم:**

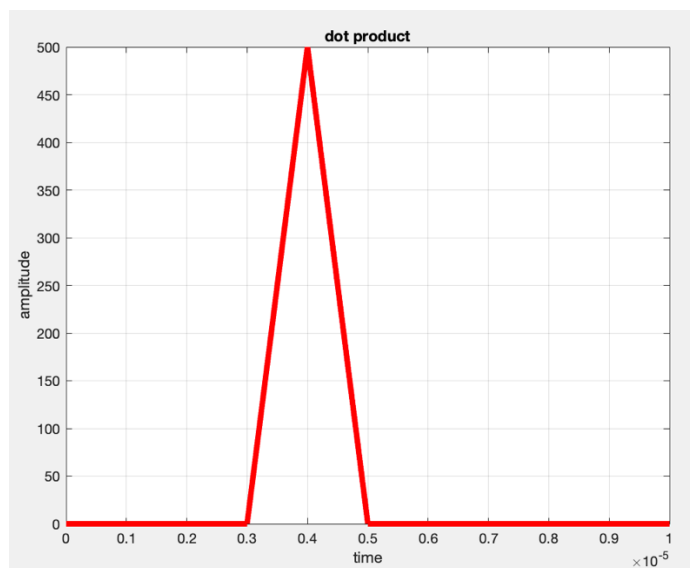
(۱-۲) سیگنال ارسالی با فرضیات گفته شده:



سیگنال دریافتی با فرض  $R = 450$  معادلاً شکل زیر است:

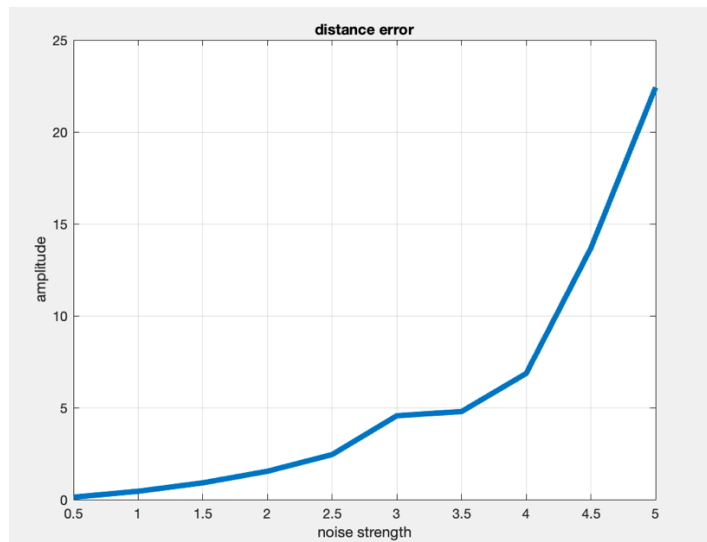


۲-۲) بله. در این بخش یک `test_signal` که از ۱ تا  $N$  مقدار ۱ و باقی مواقع مقدار ۰ دارد ساخته می‌شود. سپس کانولوشن این سیگنال با `test_signal` گرفته شده و در `ro` ذخیره می‌شود. `ro` به این شکل خواهد بود:



پس از پیدا کردن لوکیشن پیک تابع `ro` در آرایه `t` (مثلا `t0`) باید این لوکیشن منهای  $N$  شود چون در کانولوشن یکی از آرایه‌ها برعکس شده و `t0` تا شیفت داده شده تا کانولوشن به ماکسیمم برسد. پس به علت این وارونگی `t0` منهای  $N$  شده تا زمان درست پیک آرایه `ro` مشخص شود.

۲-۳) نمودار زیر نشان‌دهنده مقدار خطا بر حسب قدرت نویز است. این شکل نشان می‌دهد اگر قدرت نویز را از حدود ۳ بیشتر کنیم خطای فاصله بیش از حد زیاد می‌شود.

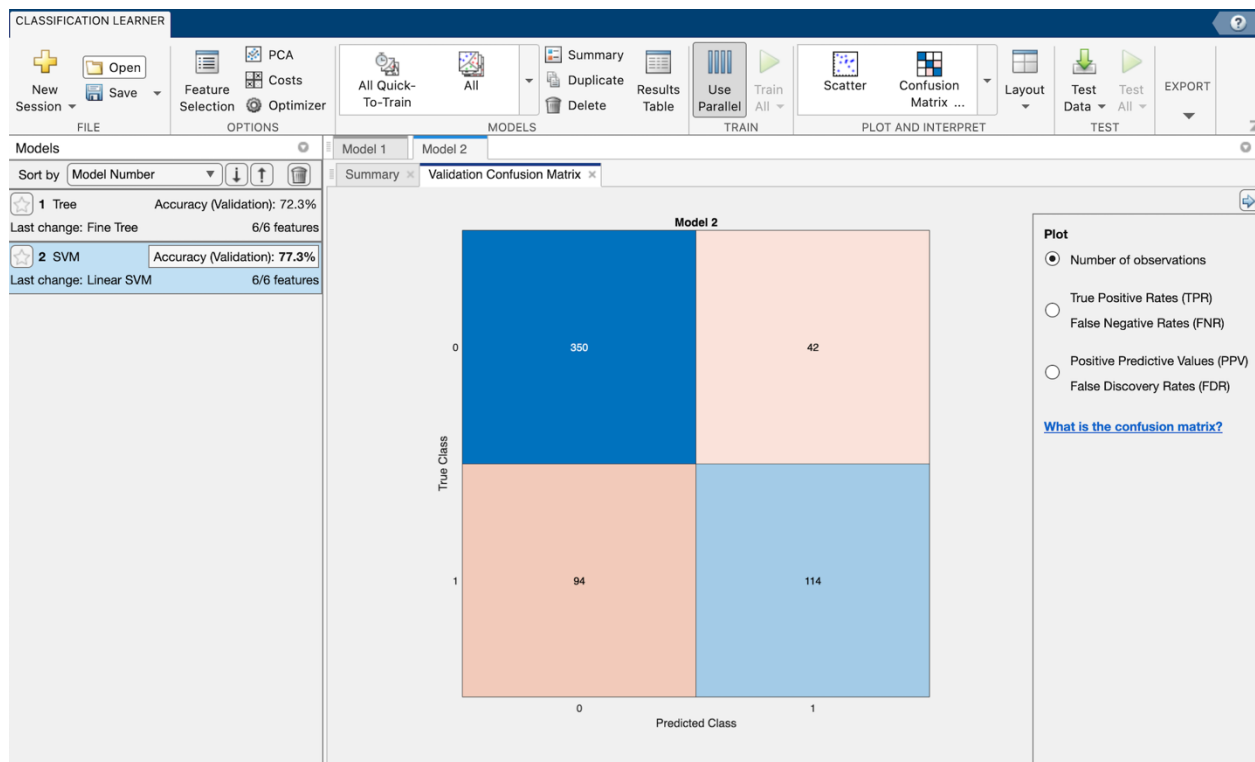


بخش سوم:

در این بخش پس از ضبط صدا، با استفاده از اسکریپت `make_sounds` متغیر `sounds` که یک `map` از کلمات فارسی به صداهاست ساخته و ذخیره می‌شود. سپس در تابع `calling_customer` این `map` لود شده و پس از چک کردن درست بودن اعداد ورودی صدای مورد نظر پخش می‌شود.

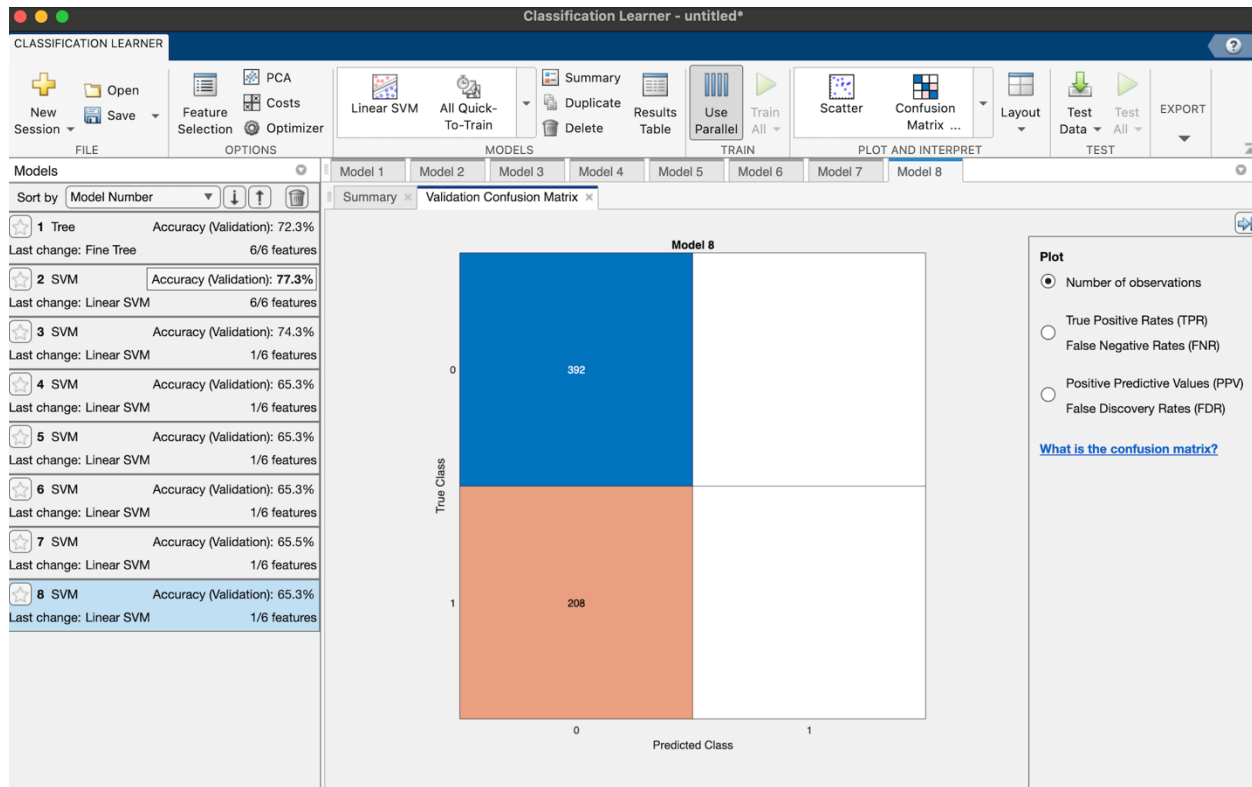
بخش چهارم:

۴-۱) دقت به دست آمده برای روی داده‌ها: ۷۷.۳٪



۴-۲) دقت به دست آمده برای هر فیچر:





- گلوکز: ۷۴.۳٪
- فشار خون: ۶۵.۳٪
- کلفتی پوست: ۶۵.۳٪
- انسولین: ۶۵.۳٪
- BMI: ۶۵.۵٪
- سن: ۶۵.۳٪

گلوکز به دیابتی بودن بیشتر ربط دارد چون استفاده از آن در train کردن مدلا باعث دقت بیشتری در گرفتن پاسخ می‌شود.

۳-۴ در این بخش label ۷۷.۵٪ داده‌ها به درستی تشخیص داده شد که حدوداً برابر پاسخ قسمت ۴-۱ است.

۴-۴ دقت فاز تست (ارزیابی) ۷۸.۰٪ است.

train accuracy percentage: 77.50  
test accuracy percentage: 78.00