آ) 1-1 processing missing values.ipynb

روش های به کار برده شده به ترتیب :

۱. حذف سطری که مقدار گم شده دارد
۲. حذف متغیری که مقدار گم شده دارد
۳. استفاده از مشاهده قابل قبول داده قبلی / بعدی
۴. جاگذاری مقادیر گم شده با مقدار میانگین ستون
۵. جاگذاری مقادیر گم شده با مقدار میانه ستون
۶. استفاده از روش k نزدیک ترین همسایه

اول داده ها را باز میکنم. اسم ستون ها در سطر سوم آمده و دو سطر اول اطلاعات منبع داده است و ربطی به متغیرها ندارد. اسم گذاری ستون ها (متغیرها) را اصلاح می کنم. و یک دیتافریم آماده را داریم تا کار را دنبال کنم.

از اطلاعات به دست آمده ۴ مقدار گم شده داریم. ۲ تا مربوط به ستون Population growth و دوتای دیگر هر کدام متعلق به ستون های Area (sq. km) و Total population هستند.

۱. حذف سطری که مقدار گم شده دارد

از ۱۵ سطر اولیه ۴ تا حذف شد و ۱۱ تا داده قابل استفاده و کامل داریم. یعنی حدود ۲۵ درصد داده ها را حذف کردیم در حالی که ممکن بود اطلاعات ارزشمندی داخل ان باشد. سعی می کنیم از روش های بهتر استفاده کنیم.

۲. حذف متغیری که مقدار گم شده دارد :

از بین متغیرهایی که داریم، (البته هنوز وارد مرحله مدلسازی نشدیم) ولی با توجه به متن سوال Coronavirus Cases را به عنوان متغیر هدف میگیریم (در آینده) و با توجه به اطلاعاتی که داریم ٤ متغیر به دردبخور برای استفاده در مدل داریم.

اگر هر ۳ ستون را حذف کنیم، تنها ۱ متغیر برای ادامه کار می ماند و ما مقدار زیادی از اطلاعات را دور ریخته ایم که خب اصلا منطقی نیست. مثلا اگر مدا ما رگرسیون باشد، در ابتدا رگرسیون چندگانه داریم و در پایان حذف ستون ها تنها یک متغیر باقی می ماند تا مقدار هدف را تخمین بزند. پس این روش اصلا مناسب نیست. تنها زمانی استفاده از این روش ممکن است قابل قبول باشد که یک ستون داشته باشیم که که در بیشتر سطرها مقدار گم شده داشته باشد. در آن صورت می توان حذف آن را به عنوان یک گزینه مطرح کرد (اگر گزینه ی بهتری نباشد)

۳. استفاده از مشاهده قابل قبول داده قبلی / بعدی

در هر ستون مقدار گم شده ی سطر مورد نظر را با اولین مقدار ثبت شده ی قبل از آن پر می کنیم. یا از اولین مقدار بعد از آن استفاده می کنم. (فرقی ندارد)

یعنی در کل۴ مقدار تکراری به جای ۴ مقدار گم شده می گذاریم. نسبت به گزینه های قبلی نسبتا بهتر است اما لزوما مطلوب نیست. ممکن است اطلاعاتی که به دست می آید نامعتبر باشد.

۴. جاگذاری مقادیر گم شده با مقدار میانگین ستون

میانگین اندازه های هر ستوان را به جای مقدار گم شده ی ان ستوان بگذاریم. مزیت روش این است که مقدار میانگین در نهایت تغیری نمی کند. چون برای انجام این کار با ()data4.mean در pandas به مشکل خوردم از (impleImputer(strategy='mean استفاده کردم. که نتیجه کار مشابه است. اگر یکی از ستون هایی که گم شده داشت کیفی بود این روش کارایی نداشت.

۵. جاگذاری مقادیر گم شده با مقدار میانه ستون

مثل مورد قبل فقط به جای میانگین از میانه استفاده کند. برای داده های غیرکمی هم قابل استفاده ست. اگر باکس پلات داده ها را رسم کنیم، در این روش تغیری نمیکند و چنک ها سرجای قبلی می مانند.

۶. استفاده از روش k نزدیک ترین همسایه

در این روش از الگوریتم KNN استفاده می کنم تا بر اساس کل مقدار متغیرهای هر سطر که مقدار گم شده دارد، آن را با مقدار معقولی که نزدیک ترین به پاسخ گم شده ی احتمالی است جایگزین کنم. این یک روش طبقه بندی است و چون برای پیدا کردن مقدار گم شده از همه ی داده های موجود در تمام سطرها و ستون های دیگر استفتده می کند به نظرم ارجح است. و برای مراحل بعدی با همین نتیجه ادامه می دهم.

در آخر نتایج را در OUT 1-1 data(processed missing values).csv برای مرحله بعدی ذخیره می کنم.

پ.ن : البته یک روش دیگر هم در کلاس مطرح شده که مناسب تر برای داده های دسته بندی است (categorical). اینکه با بیشترین فراوانی هر ستون مقدار گم شده را پر کنم که اینجا اصلا کاربرد مناسبی نداشت.

ب) 1-2 encoding categorical values.ipynb

برای مدل رگرسیونی فقط با مقدار عددی سر و کار داریم اما به این معنی نیست که متغیر کیفی دسته ای را نمی توان با مدل رگرسیون استفاده کرد. فقط نیاز به پیش پردازش دارد تا به مقدار عددی قابل استفاده در معادلات تبدیل شود. چون به همین صورت موجود نمی توان ان را در معادلات بهینه سازی همراه با بقیه متغیرها جمع کرد وخطا را بهینه سازی کرد تا معادله رگرسیون به دست بیاید. به ازای هر گروه یا هر دسته کیفی که در این ستون International Visitors وجود دارد باید یک کد نظیر کنیم (مثلا اگر ۵ گروه داریم اعداد ۱ تا ۵) و سپس نظیر هر یک از این اعداد یک ستون به متغیرها اضافه شود با این مفهوم که " آیا این سطر دارای ویژگی عدد فلان (مثلا۱ برای اولین ستون ... تا ۵ برای پنجمین) هست یا نه" و عدد مقدار نظیر ان را ۰ یا ۱ قرار دهیم. اگر متغیر مورد نظر باینری باشد در همان مرحله اول ۰ تا ۱ داریم و کافی ست. اما اینجا متغیر ما ۴ حالته ABCD است.

A : 0

B : 1

C : 2

D : 3

دیتافریم encoded_international_visitors در نتیجه این مرحله دارای ۴ ستون است. که این ها باید به جای ستون International Visitors در دیتافریم اصلی قرار بگیرند.

البته در پایان باید یکی از اینها را حذف کرد چون که اگر به ازای ABC هر چیزی داشته باشیم مقدار ستون D وابسته به اینها بدست میاید و نیازی بهش نیست. نتیجه را در `OUT 1-2 data(categorial values encoded).csv` برای مرحله بعدی ذخیره می کنم.

ج) 1-3 feature scaling.ipynb

چون قرار است مدلی که اجرا میکنیم مدل رگرسیونی باشد، این خیلی الزام اور نیست. در حالی که مثلا برای روشی مثل k نزدیک ترین همسایه اثر بزرگی بردار متغیرها می تواند نتیجه را به کلی تغیر دهد. اما در مدل رگرسیون چون ضریبی نظیر هر یک از متغیرها (x_i ها) به دست می آید که رابطه ان را با هدف نشان می دهد، این تاثیر بعد در خود مدل حلاجی می شود و مثلا متغیری که مقادیر حدود $۱۰^۶$ دارد به اندازه ارتباط خطی اش با متغیر هدف ضریب اهمیت می گیرد و نه متناسب با اینکه مقدار آن از یک متغیر از حدود ۰٫۱ بیشتر است. پس در رگرسیون الزامی به استاندارد سازی نداریم و انواع روش های استاندارسازی هم فاییده ی چندانی ندارد مگر در شرایط خاصی که لگاریتم اندازه متغیرها تفاوت خیلی فاحسی با هم داشته باشد شاید به خاطر راحتی کار یا جلوگیری از خطای سرریز احتمالی از این روش ها استفاده کنیم. در فایل کد استفاده از توابع MinMaxScaler,Normalizer که به ترتیب اولی تمام ستون ها ی داده های کمی را به فاصله بین ۰ و ۱ نگاشت داده (براساس روش استاندارد سازی مینیمم ماکسیمم) و دومی نرمالسازی داده ها را براساس نرم ۲ برداری محاسبه کرده و قرار داده تاثیری بر مدل رگرسیونی اینده ما نمیگذارد و استفاده از انها بی مورد است. اما استاندارد سازی داده ها برای رگرسیون می تواند مفید باشد. به معنی اینکه میانگین ستون ها را صفر و واریانس انها را برابر ۱ کنیم. برای اینکه رگرسیون مدل بهتری بدهد، لازم است تا توزیع داده ها به توزیع نرمال نزدیک باشد. در درس داشتیم که اگر با میانگین ها سر و کار داشته باشیم این توزیع نرمال وجود دارد اما اینجا نتایج ستون های ما میانگین نیستند (میانگین چندتایی از نمونه برداری منظور هست) پس می توان با استفاده از تبدیل این توزیع را به توزیع نرمال نزدیک کرد تا جواب حاصل از رگرسیون چندگانه ای که در ادامه به دست می اورم تا حد بیشتری دقت و درستی داشته باشد.

پس اول ستون ها را با ارایه numpy اماده و جداسازی میکنم. (به دلیل اورهایی که رخ داد این کار را یکباره روی کل دیتا فریم انجام ندادم و از طرفی چون ۳ ستون بیشتر نبود، یکی یکی انجام دادم) و بعد تبدیل scale2=StandardScaler() را روی تک تک این ارایه ها آنجام دادم و در اخر مقدار ستون های نظیر این ارایه ها را روی دیتا فریم اصلی بازنویسی کردم. حالا توزیع داده های روی هر یک از این ۳ ستون نزدیک به نرمال است. با میانگین ۰ و واریانس واحد.

$$z = (x - u) / s$$

جواب این تبدیل روی ستون های کمی Area (sq. km) و Total population و Population growth را در OUT 1-3 data(standard-transformed).csv برای ادامه کار ذخیره می کنم.

پ.ن: بدون این تبدیل هم برای مدل رگرسیون مشکل خاصی ایجاد نمی شد.

پ.ن۲: یک دلیل دیگر استفاده از این تبدیل در سوال بعدی و توضیح داده می شود که هدف آن کم کردن تاثیر داده های پرت احتمالی است.

د) 1-4 processing outliers.ipynb

اولین کاری که می کنم این که چک کنم در بین داده ها داده ی پرت وجود دارد یا نه و اینکه اگر هست، در کدام ستون ها و وضعیت انها چطور است. هر کدام از ستون ها را با رسم باکس پلات انها بررسی می کنم. بله ، داده پرت داریم.

در Population growth سه تا مقدار پرت داریم. و در Total population و Area (sq. km) به ترتیب یک و دو. دو روش اصلی برای پیدا کردن و حذف این ها داریم یکی IQR و دیگری Z-Score. از انجا که روش IQR هم منطبق بر این نمودارهایی است که رسم کردم و هم اینکه روش بهتری برای تشخیص داده پرت هست، از همین استفاده میکنم. هر ستون کافیست مقدار چارک اول و سوم را پیدا کنیم. اختلاف اینها را به دست می اورم. طبق این روش تمام داده هایی که در فاصله ی یک و نیم برابری این عدد بدست امده نسبت به چارک اول و سوم قرار بگیرند به عنوان داده پرت رد می کنیم.

نتیجه ی (column123>Q3+1.5*IQR) | (column123<Q1-1.5*IQR) در واقع همان داده های پرت داخل باکس پلات های اولیه را برای من پیدا کرده. حالا این ها را حذف می کنم تا منظور سوال براورده شود.

بعد از حذف ۱۰ سطر داده برای من باقی ماند (۵ تا از داده های اولیه یعنی حدود ۳۰ درصد داده ها را حذف کردم) سوال اینکه آیا این روش مناسب بود؟

در حالت کلی چون رگرسیون به داده های پرت حساس است به نظرم حذف داده های پرت می تواند روش مناسبی باشد البته اگر که اثر داده هایی که مقدار پرت دارند برای درستی مدل اهمیت نداشته باشد. مثال تراکنش بانکی که در کلاس گفته شد یکی از نمونه هایی است که نمی شود از حذف استفاده کرد. (البته معمولا در چنین نمونه هایی خود رگرسیون هم خیلی روش مناسبی نیست)

به نظرم به دلایل زیر حذف در اینجا مناسب نیست :

۱. واقعا ۳۰ درصد داده ها داده ی پرت باشد؟ همچین مقداری خیلی بیشتر از این است که بتوان از ان گذشت و بدون بررسی حذف کرد.
۲. با توجه به نوع داده هایی که باهاش سر و کار دارم خیلی بعید است که این اعداد گزارش شده خطا باشند. مثلا درمورد هند وسعت مساحت و عدد جمعیتی مشخصا درست گزارش شده. یا رشد جمعیت کشور کویت و قطر با وضعیت این کشورها مناسبت دارد.
۳. پس اینجا اطلاعات احتمالا غلط نداریم و پرت بودن مقدار این مقادیر بیشتر مفهوم دارد تا متعادل بودن انها با سایر ارقام

سوال اصلی این است که حالا این مقادیر عجیب چقدر باید در مدل ما تاثیر داشته باشند؟ اگرقرار به حذف باشد، یعنی کلا آن ها را تاثیر ندهم که به نظرم دور ریختن اطلاعاتی است که می توانند مفید باشند. میتوان داده های پرت را با مقادیری مثل میانگین یا مثلا با روش های خوشه بندی از روی داده های مشابه یا حتی مثل داده های گم شده با الگوریتم نزدیک ترین همسایه جایگزین کرد تا مثلا اگر یک فیلد داده پرت داریم اما اطلاعات سایر فیلدها برایمان قابل استفاده باشد و اثر دور ریختن داده ها کمتر محسوس باشد. اما به نظرم بازهم به دلیل هایی که گفتم این عددها معنادار هستند و انجام این کارها اینجا مناسب نیست. کاری که می شود کرد اینکه با استفاده از تبدیل هایی مثلا همان تبدیل که برای نزدیک کردن داده ها به توزیع

نرمال در سوال قبلی استفاده شد استفاده کنیم تا این عدد ها هرچند که معنی دارند کاملا حذف نشود بلکه تاثیر انها روی جوابی که در نهایت به دست می اید کمتر شود. می شود به طور مشابه از RobustScaler هم استفاده کرد که میانه را صفر میکند و مقادیر را بر فاصله میان چارکی که محاسبه کردیم (بالا) تقسیم می کند که از این راه تاثیر مقدارهایی که خارج ان بازی ۱٫۵ برابری ان قرار دارند را کم می کند و این در نتیجه کاهش مقدارشان و مثبت و منفی شدن انها از هر دو طرف میانه صورت میگیرد . من هر دو را انجام دادم (اولی را در سوال قبل) ولی نهایتا به ترجیح دادم که همان نتیجه ی سوال قبلی یعنی حاصل StandardScaler() را برای ادامه کار استفاده کنم که دو مزیت دارد : هم تاثیر داده های پرت را تا حدی کمتر می کند و هم میانگین صفر و واریانس واحدی که ازش نتیجه میشود برای همگرا شدن مدل رگرسیون به یک مدل دقیق تر را راحت میکند. جدا از این، روش RobustScaler در این مثال خاص چون تمام outlier ها در یک طرف باکس پلات قرار گرفته اند و مقدارشان هم کوچک است تاثیر چندانی روی کل ماجرا ندارد.

نتیجه ی این قسمت همان فایل مرحله قبلی است که البته ان را برای راحتی ادامه کار با نام OUT 1-4 data(outlier_passed).csv دوباره ذخیره کردم.

پ.ن: برای اینکه مطمئن شوم حذف نکردن اوت اتیر ها بهتر بوده در مرحله آخر یک بار مدل رگرسیونی را با این داده ها و یک بار بعد از حذف کردن اوت لایرها انجام دادم. نتیجه حذف نکردن بهتر بود پس همین روش را ادامه دادم. البته گزارش این ازمایش را نیاوردم چون تعداد داده ها خیلی کم بود گزارش کردن نتایج و نمودارها نتایج کاملا بی معنی داشتند. ولی علاوه بر دلایلی که بالا گفته شد از این روش هم برای پاسخ به این سوال می توان استفاده کرد که حذف اوت لایر مناسب است یا نه.

۵) 1-5 multiple regression.ipynb

برای تولید مدل از lrm=LinearRegression() استفاده میکنم این مدل مقدار Coronavirus Cases را بر اساس مقدار ۶ متغیر ورودی براورد میکند. برای بدست اوردن اطلاعات مدل شامل ضرایب و غیره از جدولی که از statsmodels.formula.api میگیرم استفاده میکنم. و در نهایت مقدار خطاها را هم حساب میکنم (توی جدول نبود).

اطلاعات به صورت زیر :

سطر اول مقدار عرض از مبدا و سطرهای بعدی هر یک ضریب متغیر نوشته شده نظیر ان را در معادله رگرسیون خطی چندگانه نشان میدهد.

Intercept	*50366.943585*
Population growth	-7295.630057
Total population	-2979.059785
Area (sq. km)	11846.317535
A	130108.758225
B	-28201.448627
C	-41830.434172

خطاها به ترتیب برابر :

MSE	*405381115.26471084*
RMSE	*20134.078455809962*

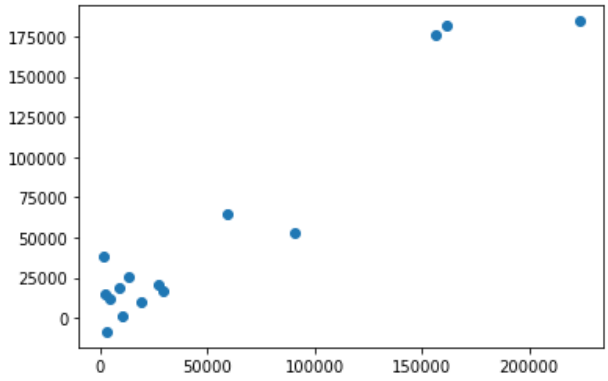
معیارهای ارزیابی مدل :

اول برای آزمون مقدار پی :

گفتیم هرچه مقدار پی کوچکتر از الفا باشد میتوان فرض اچ صفر (اچ صفر : برابری ضریب نظیر متغیر با عدد صفر ، به معنی نبود رابطه خطی بینy و x_i) در جدول زیر مقدار پی های نظیر هر سطر به جز B و عرض از مبدا از الفا بزرگتر شده (در صورتی که الفا را ۰،۰۵ مطابق معمول بگیریم) که نشان می دهد مدل خوبی نیست (رابطه ی بین متغیرها با متغیر هدف خطی نیستند)

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.037e+04	2.04e+04	2.466	0.039	3276.929	9.75e+04
Q("Population growth")	-7295.6301	7829.727	-0.932	0.379	-2.54e+04	1.08e+04
Q("Total population")	-2979.0598	7919.727	-0.376	0.717	-2.12e+04	1.53e+04
Q("Area (sq. km)")	1.185e+04	8053.118	1.471	0.179	-6724.207	3.04e+04
Q("A")	1.301e+05	2.72e+04	4.785	0.001	6.74e+04	1.93e+05
Q("B")	-2.82e+04	2.32e+04	-1.215	0.259	-8.17e+04	2.53e+04
Q("C")	-4.183e+04	2.58e+04	-1.618	0.144	-1.01e+05	1.78e+04

البته درست که همه ی این مقدارها نزدیک صفر هستند اما به اندازه کافی کوچک و نزدیک صفر نیستند پس مدل ما زیاد اطمینان بخش نیست. (در نمودار هم میشود تشخیص داد شکل بدست امده خیلی شبیه خط نیست) . علاوه بر این درست هست که با بازه اطمینان های بدست امده کار چندانی نداریم اما بد نیست توجه کنیم که در بعضی از این ها، صفر داخل بازه هست و همین باعث میشود که فرض اچ صفر قابل رد نباشد.



تا اینجا البته برای رد مدل کافیست ولی برای کامل شدن همه ی قسمت ها :

معیار ارزیابی دیگری که می توان استفاده کرد ازمون ضریب تعین یا ضریب رگرسیونی هست یا همان سطر دوم جدول زیر که برابر است با ضریب همبستگی به توان ۲ که برای رگرسیون چندگانه استفاده می کنیم. اگر این مقدار نزدیک ۱ باشد مدل خوبی داریم. اینجا ۰٫۸

R-squared:	0.914
Adj. R-squared:	0.850
F-statistic:	14.25
Prob (F-statistic):	0.000698
Log-Likelihood:	-169.94

البته در صورت تعارض این دو تا معیار گفته بودیم که اعتباری که به ازمون پی میدهیم بیشتر است. بنابراین مدل همچنان مردود هست. با بقیه اعداد این جدول در مورد رگرسیون خطی کاری نداریم.

پ.ن: البته ممکن است نتایج به خاطر این باشد که تعداد داده ها خیلی کم است.

و) 1-6 non linear regression.ipynb

از روی داده هایی که آماده کرده بودیم با ستون Total population یک دیتافریم جدید میسازیم که ستون اول ان مقدار Total population و ستون دوم ان همین مقادیر با توان ۲ باشد. و مدل رگرسیون خطی برای دو متغیر (ستون های دیتافریم جدید) به ازای متغیر هدف Coronavirus Cases ایجاد میکنم که روش کار مثل سوال قبلی هست فقط تحلیل ها و خطاها و غیره را لازم نداریم. این مدل معادل مدلی است که در سوال خواسته شده. ضرایب بدست امده از این روش این طور معنی میشوند :

ضریب بدست آمده برای متغیر نظیر Total population ← ضریب x

ضریب بدست آمده برای متغیر نظیر Total population**2 ← ضریب x^2

عرض از مبدا هم به جای خودش

پس اعداد بدست امده برای مجهولات :

a	-15440.070964
b	44642.402653
c	69393.4042975207