به نام خدا

مهرانه مقتدائی فر ۹۷۲۲۲۰۸۶

تاریخ: ۱۴۰۰/۱۱/۰۵

گزارش تمرین سری ۵ – تمرین امتیازی (سری زمانی) درس: مبانی علوم داده استاد درس: دکتر خردپیشه

در این تمرین هدف، بررسی سری های زمانی است. این سری ها، دنباله‌ای از داده ها هستند که در یک بازه زمانی جمع آوری شده‌اند. هدف از بررسی اینگونه داده ها، ساختن مدل آماری برروی آنها است تا بتوانیم پیشبینی درمورد آینده آنها داشته باشیم.

بر روی این نوع داده ها میتوان آنالیز های زیادی انجام داد، تا اطلاعات خوبی بدست بیاوریم. مانند داده های معمولی، برروی آنها تست های مخصوصی انجام میشود تا بتوان از این اطلاعات بدست آمده استفاده کرد و با آنها مدل های شبکه عصبی و یا مدل های خاصی که برای اینکار استفاده میشوند را پیاده سازی کرد و پیشبینی انجام داد.

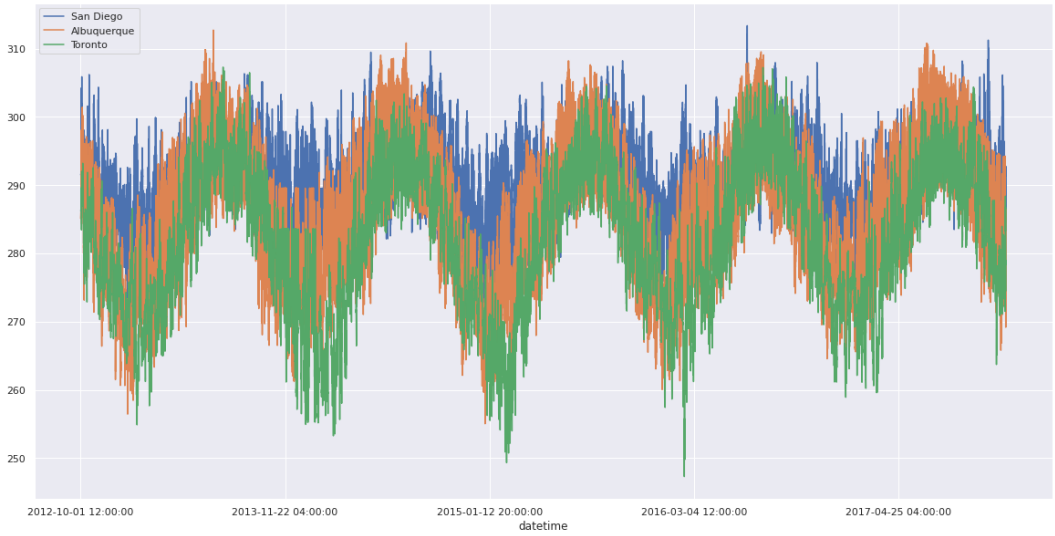
در این تمرین خواسته شده که این بررسی ها برروی داده های آب و هوا، مانند تعیین دما، فشار، سرعت باد، کیفیت هوا و ... انجام شود.

داده انتخاب شده مربوط به دیتاست Kaggle است که اطلاعات آب و هوا چند شهر را از سال 2012 تا 2017 در اختیار دارد.

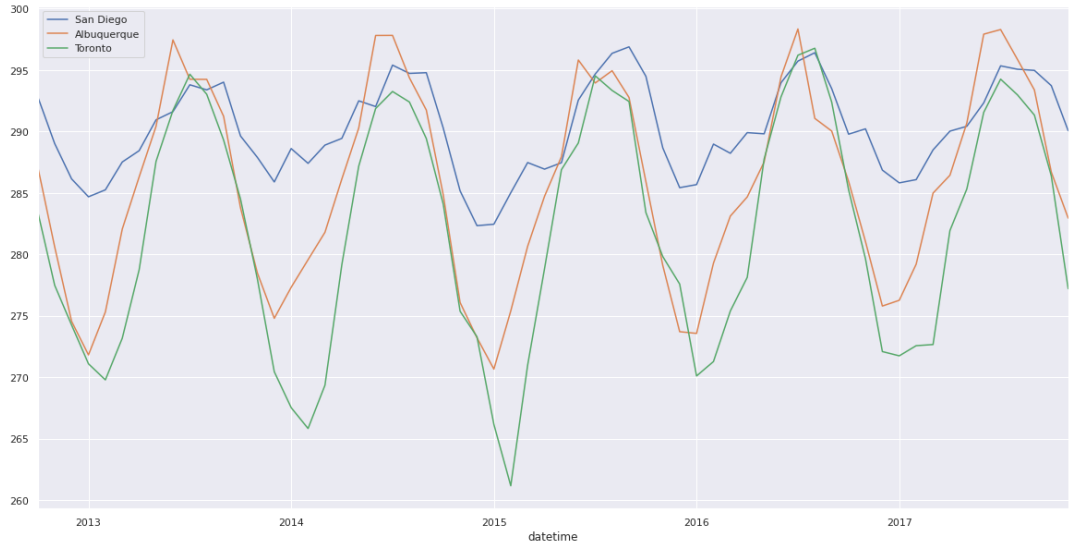
بررسی داده

در این دیتاست، اطلاعات متفاوتی از دمای هوا، فشار، سرعت باد، جهت باد، رطوبت و .. برای چندین شهر مختلف در سال های مختلف وجود دارد. بررسی هایی که در ادامه انجام داده شده برروی "دمای هوا" بوده است و برای سه شهر San Diego, Albuquerque, Toronto انجام شده است.

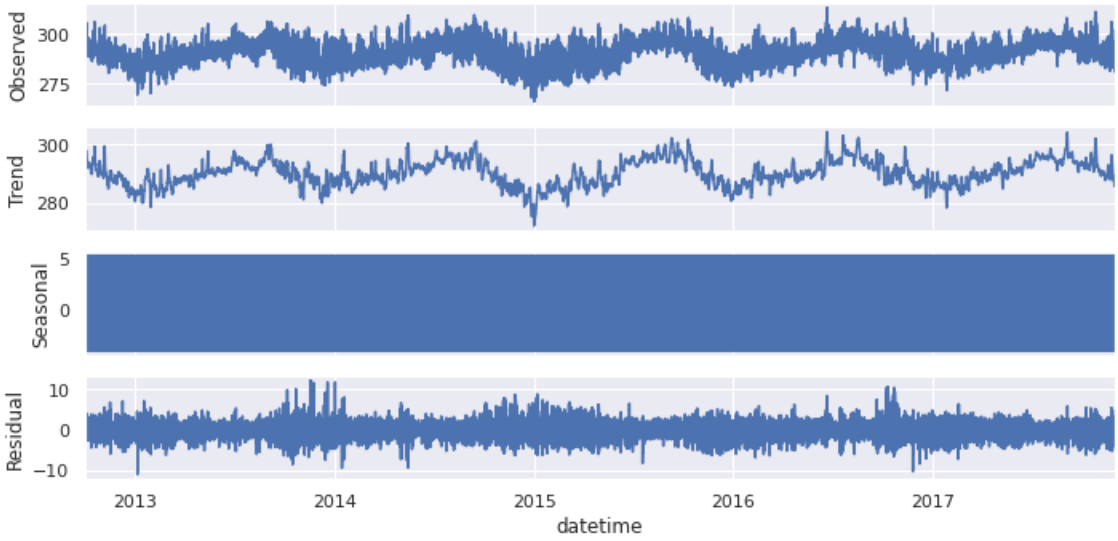
با توجه به داده های این سه شهر و همچنین ستون datetime یک دیتاست جدید میسازیم. با بررسی آن متوجه میشویم که تنها ۱ سطر از آن مقدار ندارد که در تاریخ **2012-10-01 12:00:00** بوده است. از آنجایی که تنها یک سطر است ، آن را از دیتاست حذف میکنیم. حال پس از آنکه اندیس را برابر با time قرار دادیم، میتوانیم نمودار زیر را که نشان دهنده تغییرات دمای هوا برای این سه شهر در سال های 2012 تا 2017 است را داشته باشیم:



نمودار دیگری برای بررسی trend و seasonality در زیر میبینیم. این نمودار نشان میدهد که در این ۵ سال، در هر ۱۲ ماه میانگین تغییر دمای هوا چگونه بوده است:



نمودار زیر نیز با استفاده از یکی از پکیج های tsa رسم شده است، که برای شهر San Diego به طور خاص، trend و تغییرات دما را نشان میدهد.



معیار ADF نیز برای تغییرات دما هر سه شهر محاسبه شده است، برای San Diego برابر با 7- ، Albuquerque برابر با 6.6- و Toronto برابر با 6.7- است. مقدار p-value برای آنها نیز محاسبه شده و برابر با 0 است. این نشان دهنده stationary بودن داده ها میباشد. البته stationary بودن در نمودار تغییرات نیز کاملا مشخص است، زیرا تغییرات به صورت سینوسی است بدین معنا که در یک بازه زمانی کم میشود، دوباره افزایش میابد و دوباره کاهش و همین روند تکرار میشود. پس یک روند تغییرات منظم دارد که همین به معنای Stationary بودن این داده ها است.

در ادامه به داده ها، دو ستون سال و ماه را بر اساس تاریخ پیدا کرده و بر اساس ماه، فصل را نیز اضافه کردیم. به ستون سال در ادامه نیاز خواهیم داشت.

پیاده سازی مدل ها

برای پیاده سازی مدل ها، ابتدا نیاز است داده های train و test را جدا کنیم. این جدا کردن برای مدل های مختلف به صورت متفاوت انجام شده است. توضیحات را برای هر کدام مینویسیم.

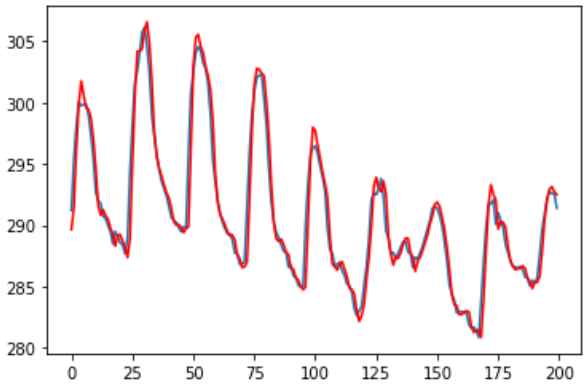
*تمامی مدل ها به طور خاص، برروی دمای شهر San Diego پیاده شده‌اند.

مدل ARIMA

برای پیاده سازی این مدل، ابتدا همانطور که گفته شد train و test را جدا میکنیم. برای اینکار، داده های سال 2017 را که سال آخر در دیتاست هست را به عنوان test و بقیه را به عنوان train در نظر گرفتیم. ابتدا مدل را برروی داده های آموزشی، fit میکنیم و سپس برروی داده های test آن را امتحان میکنیم.

نکته قابل توجه آن است که در اینجا، علاوه بر داده های train، یک history نیز برای داده هایمان در نظر گرفتیم که این history در واقع نتایج پیشین مدل برروی داده های train را در خود نگه میدارد و بدین ترتیب است که مثلا برای داده 100م، 99 داده ی قبل نیز برروی نتیجه ما تاثیر گذار خواهد بود. همینکار باعث میشود تا مدل برروی داده های تست عملکرد بهتری داشته باشد.

ابتدا تمامی 7900 داده تست را به مدل داده بودیم اما زمان بسیاری برای پیاده شدن احتیاج داشت، به همین علت برروی ۲۰۰ تای آخر مدل را test کردیم. نتایج را در زیر میبینیم:



خط آبی نمایش داده های اصلی است و خط قرمز داده های predict شده و همانطور که مشخص است اختلاف آنها خیلی کم است پس یعنی عملکرد مدل خوب است. همچنین خطای آن (MSE) برابر با 1.37 است.

مدل RNN, LSTM, GRU

برای این مدل ها، به گونه دیگری داده های آموزشی و تست را جدا میکنیم. برای اینکار ابتدا داده را shift میدهیم. یعنی به عنوان مثال داده های هرروز، شامل داده های ۱۰ روز قبل میشود.(در اینجا شیفت را برابر با 10 قرار میدهیم)

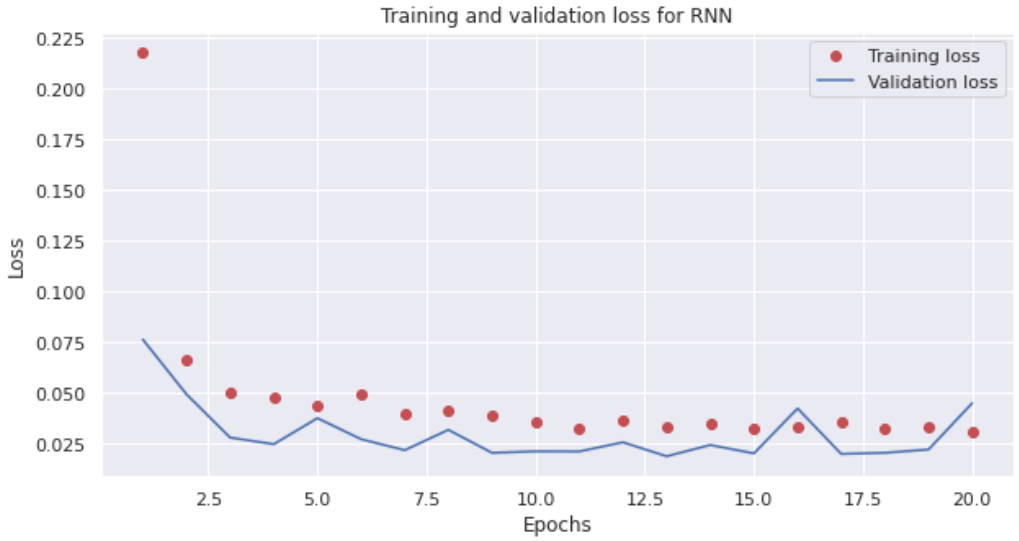
بدین ترتیب داده های x و y را میسازیم. درواقع y همان target ما است که در اینجا برابر با دمای San Diego است و x داده های ۱۰ روز قبل است. برای روز های اولیه این مقادیر null میشوند به عمین علت آن ها را drop میکنیم.

سپس با استفاده از TimeSeriesSplit داده های x و y را به train و test جدا میکنیم، سپس این داده ها را scale کرده و شروع به ساختن مدل های بازگشتی میکنیم. 1000 داده TRAIN و 800 داده TEST در نظر گرفتیم.

مدل اول RNN است.

با استفاده از مدل های sequential مدل RNN را با لایه های مختلف میسازیم. ورودی آن یک بردار ۱۰ تایی از داده های train است و خروجی ۱ است. برای جلوگیری از overfit شدن مدل از earlystopping استفاده کردیم و پس از 20 تکرار مدل متوقف شد. همچنین برای این مدل به عنوان Optimizer از adam استفاده کردیم.

loss مدل را در تصویر زیر مشاهده میکنیم:



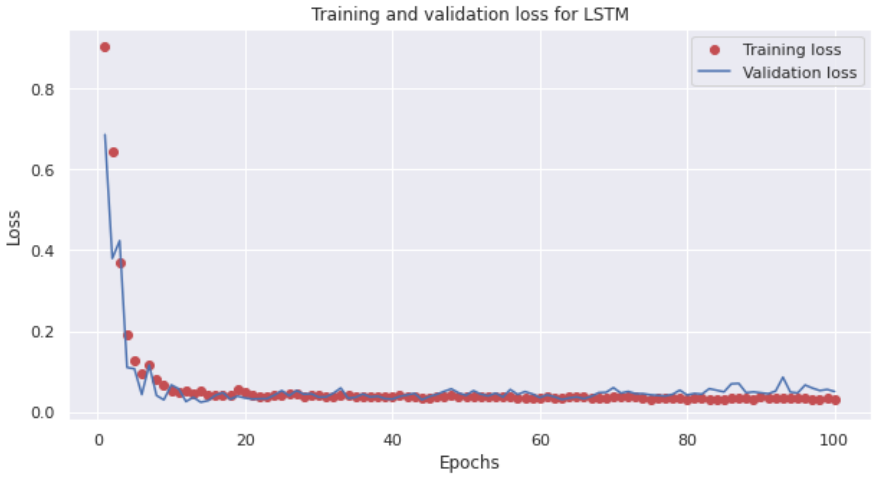
مقدار MSE برای داده های predict شده برابر با 0.041 شده است.

مدل دوم GRU:

این مدل را نیز مانند قبلی، با تعداد لایه های مختلف و تعداد نورون های متفاوت، پیاده کردیم. نتایج زیر را مشاهده میکنیم:

این مدل نیز پس از ۲۰ تکرار متوقف شد و مقدار MSE گزارش شده برابر است با: 0.037 که از مدل قبلی کمتر است.

مدل بعدی LSTM است که مقدار LOSS آن را در نمودار زیر میبینیم:

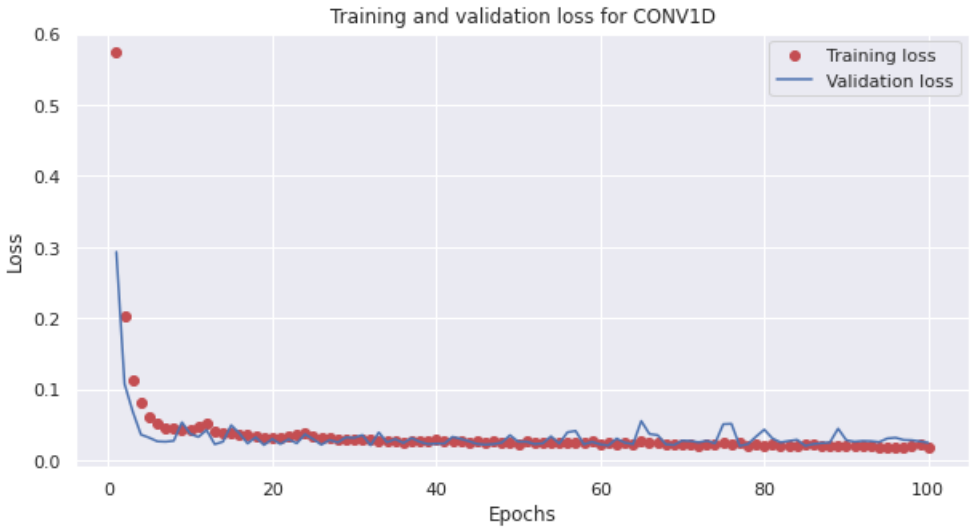


مقدار خطای MSE برابر با 0.035 است که از دو مدل قبلی کمی کمتر است و نتایج بهتری را نشان میدهد.

مدل CONV1D

این مدل را نیز مانند مدل های قبلی با استفاده از لایه های مختلف ساختیم. در این مدل convolutional از لایه های maxpooling و لایه flatten نیز برای بهتر شدن عملکرد مدل استفاده کردیم که مدل های قبلی از آن برخوردار نبودند.

نتایج آن در زیر مشخص است:



مقدار MSE آن برابر با 0.038 شده است.

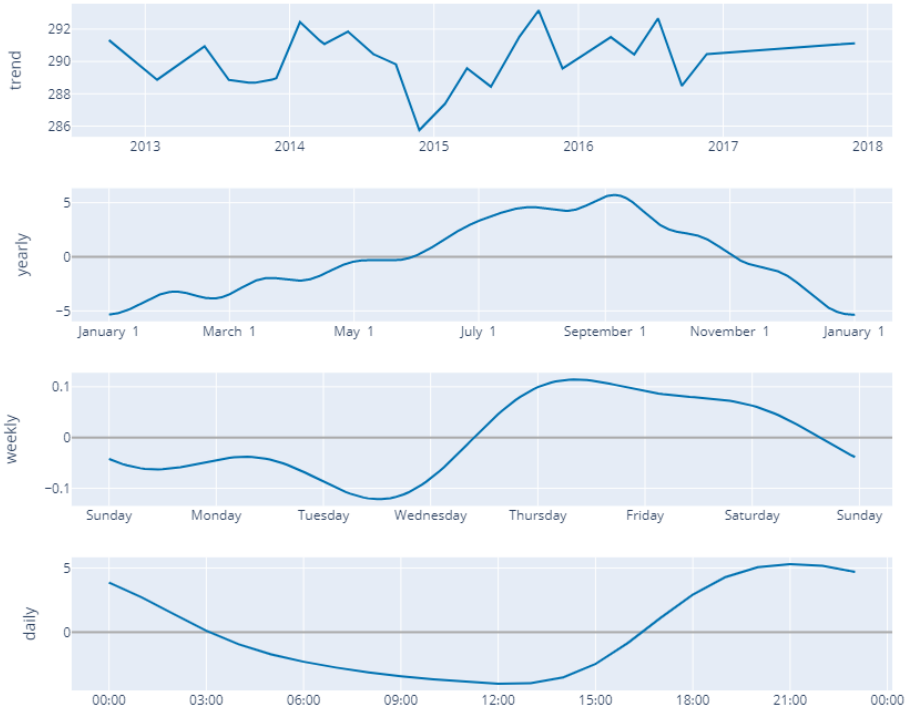
در این مدل ها، که مدل های مبتنی بر شبکه های عصبی بوده اند، استفاده از OPTIMIZER ها نتایج و عملکرد مدل را بهتر کرد. همچنین با استفاده از روش earlystopping از Overfit شدن مدل جلوگیری کردیم تا بتوانیم نتایج واقعی مدل را برروی داده های تست ببینیم.

در این مدل ها adam optimizer از بقیه Optimizer ها نتیجه بهتری برای مدل داشت.

مدل PROPHET

مدل آخری که پیاده سازی شده است، مدل PROPHET است این مدل در کتابخانه مخصوص خود قرار دارد. برای پیاده کردن آن نیاز است که دو ستون Y و DS که نشان دهنده target و تاریخ است داشته باشیم. این دو ستون را از دیتاست خود جدا میکنیم و به عنوان ds , y به دیتاست جدید میدهیم.

با استفاده از این مدل میتوانیم forecast را پیشبینی کنیم. نتایج زیر برخی trend ها را برای target ما به خوبی نشان میدهد.



همانطور که میبینیم، تغییر دما، به طور سالیه، ماهانه و روزانه نیز مشخص شده است

همچنین خطای mse گزارش شده برای این مدل برابر با 9.20 است.

نتیجه گیری:

برای بررسی داده های سری زمانی، مدل های شبکه عصبی که شامل مدل های بازگشتی و همچنین مدل convolutional است نتایج بهتری داشتند. این مدل ها حتی با تعداد لایه های کم و تعداد نورون های کم نیز نتایج خوبی داشتند زیرا داده های ما خیلی پیچیدگی نداشتند به عمین علت Loss و میزان خطای آنها برروی داده های تست نیز بسیار پایین است.

مدل prophet نیز میتواند نمودار های مفیدی را در اختیار ما قرار دهد اما MSE مناسبی به ما نمیدهد.