به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Trustworthy AI

تمرین شماره دو

نام و نام خانوادگی	سیدسروش مجد
شماره دانشجویی	دانشجوی مهمان
تاریخ ارسال گزارش	۲۵ اردیبهشت

فهرست گزارش سوالات

پرسش ۱ – SHAP

استفاده از مقادیر Shapley یکی از روش های در نظر گرفتن تاثیر ویژگی های مختلف در خروجی حاصل از همبستگی (coalition) ویژگی های دیگر است. در همین راستا، مقادیر SHAP (SHaply Additive exPlanations) یک روش کارا برای توضیح عملکرد مدل ها است.

الف: ابتدا در رابطه با مقاله SHAPبه سوالات زیر پاسخ دهید:

۱) با تعریف یک روش additive feature attribution سه ویژگی منحصر به فرد consistency، local accuracy و missingness روش SHAP را به صورت خلاصه معرفی کنید.

روش‌های Additive feature attribution مانند Lime، Explanation Modelای که دارند به صورت زیر است: (منظور از Explanation Model، مدل ساده‌تری است که برای تفسیر مدل پیچیده اصلی به کار می‌رود)

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

این مدل explanation با نسبت دادن ضریبی به هر ویژگی و جمع همه آن‌ها خروجی مدل اصلی را تخمین می‌زند. ورودی نیز (z') ساده شده ورودی مدل اصلی است (مثلا بردار ۰ و ۱ است) که با تابع h به ورودی مپ می‌شود.

برای مثال یکی از روش‌های Additive feature attribution method روش Lime یا (Local Interpretable Model-Agnostic Explanations) است که از معادله خطی بالا (g) تبعیت می‌کند. با این روش Model-Agnostic می‌توان پیش‌بینی‌های هر مدل یادگیری ماشین Black Box را با تقریب مدل به صورت محلی (Localy) با استفاده از یک مدل ساده‌تر تفسیر کرد و سپس اهمیت هر ویژگی را تخمین زد. Objective Function برای LIME:

$$\xi = \underset{g \in \mathcal{G}}{\arg\min}\ L(f, g, \pi_{x'}) + \Omega(g).$$

در این معادله f مدل پیچیده اصلی، g مدل خطی تخمین زده شده توسط Lime، pi وزن‌دهی به داده‌های مغتشش اطراف x برحسب فاصله آن‌ها از x و ترم آخر پیچیدگی مدل است که هرچه پیچیدگی کمتر باشد اینجا بهتر و مدل تفسیرپذیرتر است.

Lime ابتدا ویژگی‌های یک نمونه معین را مغتشش می‌کند و چند سمپل از روی آن نمونه و نزدیک به آن به دست می‌آورد، سپس یک مدل خطی بر روی آن داده‌های مغتشش و پیشبینی‌های مدل Black Box از آن‌ها فیت می‌کند. از اطلاعات این مدل خطی می‌توان برای ارائه تفسیر قابل فهم برای انسان در مورد اینکه چرا یک پیشبینی خاص توسط مدل در آن ناحیه (Localy) انجام شده است استفاده کرد.

Local Accuracy:

این ویژگی بیان می‌کند خروجی Explanation Model (g) برای ورودی x' با خروجی مدل اصلی (f) برای ورودی x مطابق و برابر باشد. x' ورودی ساده‌تر شده برای Explanation Model یا g است که با تابع h به ورودی اصلی (x) مپ می‌شود.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i$$

Missingness:

مطابق این ویژگی اگر برخی از ویژگی‌ها در نمونه‌هایی معین مقداری نداشته باشند، تاثیری نباید نداشته باشند.

$$x'_i = 0 \implies \phi_i = 0$$

Consistency (Monotonicity):

مطابق این ویژگی اگر مدل تغییر کند تا بیشتر بر یک ویژگی خاص تکیه کند، اهمیتی (attribution) که به آن ویژگی داده می‌شود نباید کاهش یابد. به این صورت که اگر حذف ویژگی x_i در مدل f^2 تاثیر بیشتری از f^1 تاثیر داشته باشد این ویژگی در مدل دوم تاثیر بیشتری دارد.

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$$

for all inputs $z' \in \{0,1\}^M$, *then* $\phi_i(f', x) \geq \phi_i(f, x)$.

۲) برای مقابله با پیچیدگی بالای محاسباتی مقادیر Shap، روش model-agnostic به نام Kernel Shap معرفی شده است. نحوع عملکرد این روش در مقایسه با محاسبه دقیق مقادیر SHAP بیان کنید.

هدف SHAP این است که تاثیر هر ویژگی بر پیشبینی مدل را بسنجد. در روشmodel-agnostic Kernel SHAP به جای آموزش مکرر مدل به ازای ترتیب‌های متفاوت ویژگی‌ها و انواع هم‌نشینی آن‌ها در محاسبه مقادیر دقیق SHAP که با افزایش تعداد ویژگی‌ها به صورت نمایی زیاد می‌شود، از مدلی که قبلا یک بار آموزش دیده است استفاده و به جای مقادیر خالی ویژگی‌ها به صورت رندوم از مقادیر آن‌ها در دیتا جایگذاری می‌شود. سپس مدل ساده‌تر خطی بر این فضای ویژگی فیت می‌شود. این روش به صورت local و global کار می‌کند یعنی می‌تواند اهمیت هر ویژگی به ازای یک نمونه یا کل مدل را با استفاده از مدل خطی ارائه دهد. روش Kernel SHAP از لحاظ محاسباتی بسیار کارآمدتر از محاسبه مقادیر دقیق SHAP است و تخمین خوبی از مقادیر Shapley ارائه می‌دهد. در قسمت قبل که objective function برای Lime ارائه شد می‌دانیم که مقادیر shapley تنها راه حل برای آن است به صورتی که ویژگی‌های Local Accuracy، Consistency و Missingness برآورده شود و این به Loss Function، ترم Regularization و weighting kernel بستگی دارد. روش Lime به صورت هیوریستیک این پارامترها را تخمین می‌زند و مقادیر shapley را نتیجه نمی‌دهد. در Shapey Kernel به ترتیب از بالا به پایین مقادیر ترم Regularization، Weighting Kernel و Loss Function به صورت معادله‌های زیر مقداردهی می‌شوند تا ویژگی‌های Local Accuracy، Consistency و Missingness برآورده شود. در این روش به عبارت دیگر تعدادی از ویژگی‌ها را استفاده می‌کنیم نه همه آن‌ها را. و در نهایت مدل خطی را بر آن فیت می‌کنیم.

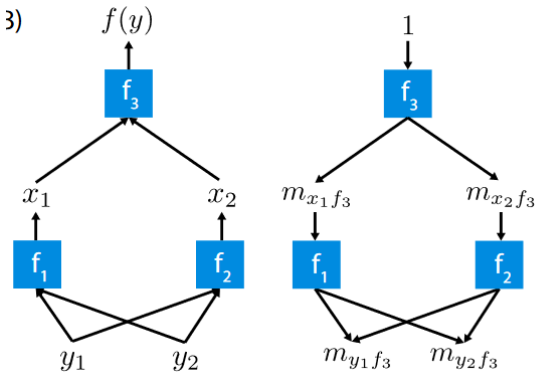
$$\Omega(g) = 0,$$
$$\pi_{x'}(z') = \frac{(M-1)}{(M\ choose\ |z'|)|z'|(M-|z'|)},$$
$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x^{-1}(z')) - g(z') \right]^2 \pi_{x'}(z'),$$

۳) در کنار روش‌های model-agnostic این مقاله روش model-specific با عنوان Deep SHAP معرفی شده است. تفاوت این روش با Kernel SHAP را بررسی کنید.

هر دو روش برای به دست آوردن مقادیر Shapley استفاده می‌شوند. روش Kernel Shap می‌تواند برای هر مدل یادگیری ماشین از جمله مدل‌های یادگیری عمیق استفاده شود ولی Deep Shap فقط برای مدل‌های یادگیری عمیق به کار می‌رود. در این روش برخلاف Kernel Shap با روش DeepLIFT که روش تفسیر پیشبینی بازگشتی است و مقادیر Shap را با در نظر گرفتن این که ویژگی‌ها مستقل از هم هستند تخمین می‌زند. ایده اصلی DeepLIFT یا Deep Learning Important FeaTures مقایسه Activation نورون‌ها در لایه‌های پنهان مدل عمیق برای یک ورودی داده شده با Activation آن‌ها برای ورودی Baseline است. با مقایسه Activation نورون‌ها برای ورودی و Baseline، DeepLIFT تاثیر هر ویژگی ورودی را در پیشبینی مدل محاسبه می‌کند.Deep LIFT چون Additive Feature Attribution Method است Local Accuracy و Missingness را برآورده می‌کند و مقادیر Shapley تنها مقادیری را ارائه می‌کند که Consistency برآورده شود.

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o,$$

منظور از o خروجی مدل و سمت راست معادله f(x)-f(r) و C تاثیر ست کردن ورودی با Reference Value می‌باشد. Deep SHAP مقادیر SHAP محاسبه شده برای بخش‌های کوچک‌تر شبکه را ترکیب کرده تا مقادیر SHAP برای کل شبکه را به دست آورد. این کار با استفاده از عبور ضرایب DeepLIFT (تعریف شده برحسب مقادیر SHAP) به شکل بازگشتی و Backward در شبکه انجام می‌دهد (شکل ۱ و معادله‌های زیر).

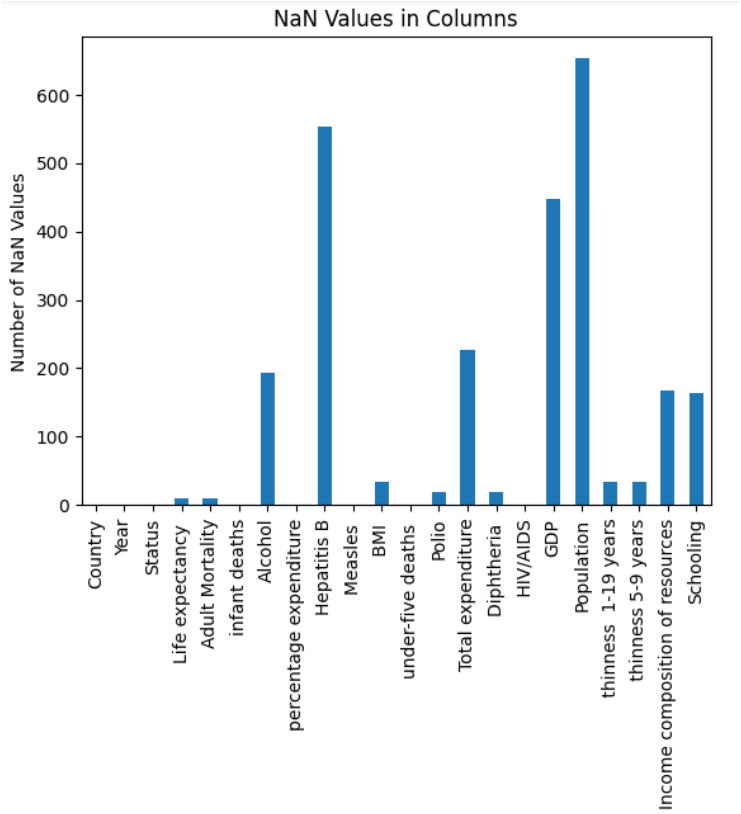


شکل ۱: **DeepSHAP**

$$m_{x_j f_3} = \frac{\phi_i(f_3, x)}{x_j - E[x_j]}$$
$$\forall_{j \in \{1,2\}} \quad m_{y_i f_j} = \frac{\phi_i(f_j, y)}{y_i - E[y_i]}$$
$$m_{y_i f_3} = \sum_{j=1}^{2} m_{y_i f_j} m_{x_j f_3}$$
$$\phi_i(f_3, y) \approx m_{y_i f_3}(y_i - E[y_i])$$

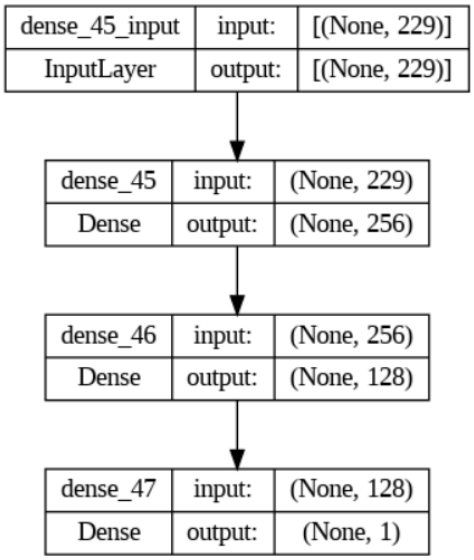
روش Deep Shap از روش‌های انتخاب هیوریستیک برای خطی سازی مولفه‌ها اجتناب کرده و به جای آن یک خطی‌سازی موثر از مقادیر SHAP محاسبه شده برای هر جز به دست می‌آورد.

ب) ابتدا در مرحله پیش‌پردازش، در دیتافریم مجموعه‌داده Life Expectency مقادیر Nan در ستون‌های GDP و Population مقدار میانه و در ستون‌های Alcohol و Hepatitis B میانگین آن ستون‌ها جایگذاری شدند. سپس بقیه ستون‌ها که تعدادی کمی مقدار Nan دارند را با مقادیر رندوم از همان ستون جایگذاری کردیم. سپس مقادیر string تبدیل به numerical و مقادیر Categorical مانند سال و کشور و status تبدیل به one-hot شدند. برای status دو ستون one-hot با نام‌های Developed و Developing در دیتافریم وجود دارد.



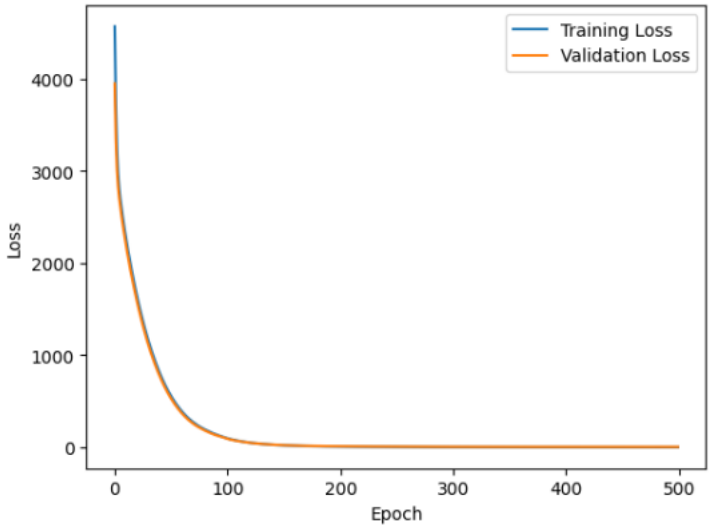
شکل ۲: تعداد مقادیر **Nan** به ازای هر ویژگی

سپس ۱۰ درصد از داده‌ها را برای تست و ۹۰ درصد برای آموزش جدا کرده به نحوی که شرط وجود حداقل یک نمونه از سه کشور در یک قاره برای داده تست برقرار باشد. از یک مدل عصبی Dense با هدف رگرسیون و برای سنجش عملکرد Deep SHAP و Kernel SHAP بهره بردیم. برای اضافه کردن non-linearity به مدل برای دقت بیشتر در رگرسیون در خروجی لایه دوم از تابع فعال‌ساز tanh استفاده شد. لایه اول هم تابع فعال‌ساز Relu دارد. معماری مدل عصبی در فریم ورک تنسورفلو در شکل زیر نشان داده شده است:



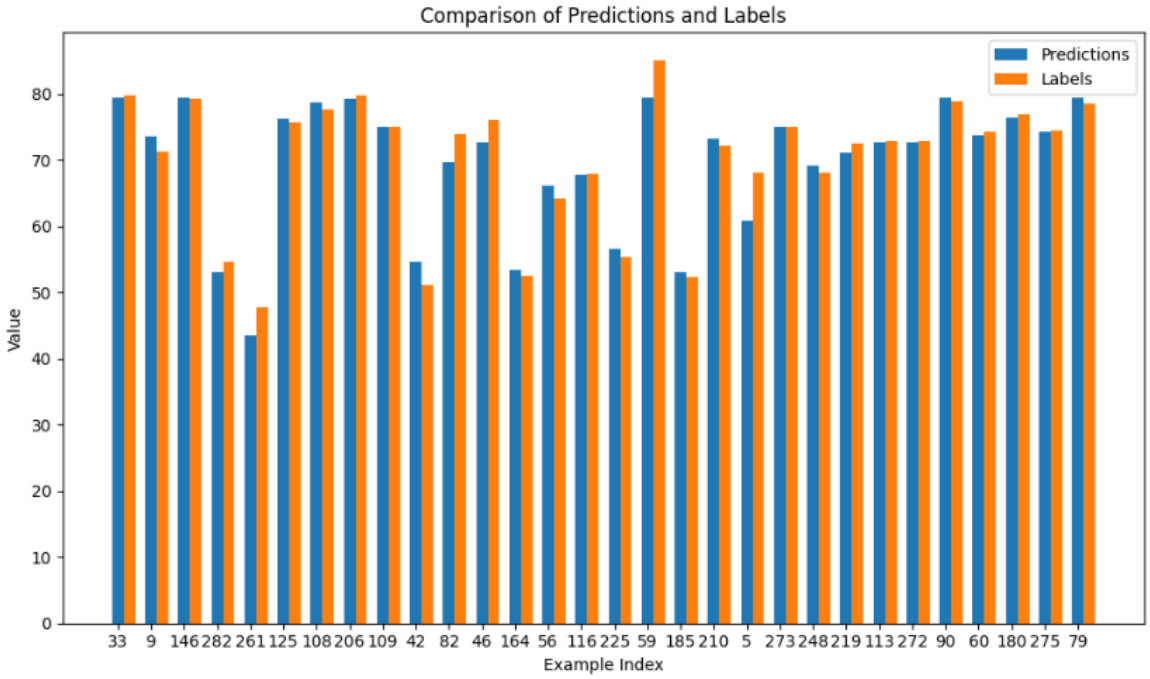
شکل ۳: معماری مدل عصبی

برای آموزش از تابع MSE Loss و اپتیمایزر ADAM استفاده شد. فرایند آموزش در شکل زیر مشاهده می‌شود. همچنین مقادیر دیتافریم Scale شدند و بین ۰ و ۱ رفتند تا وارد شبکه عصبی شوند.



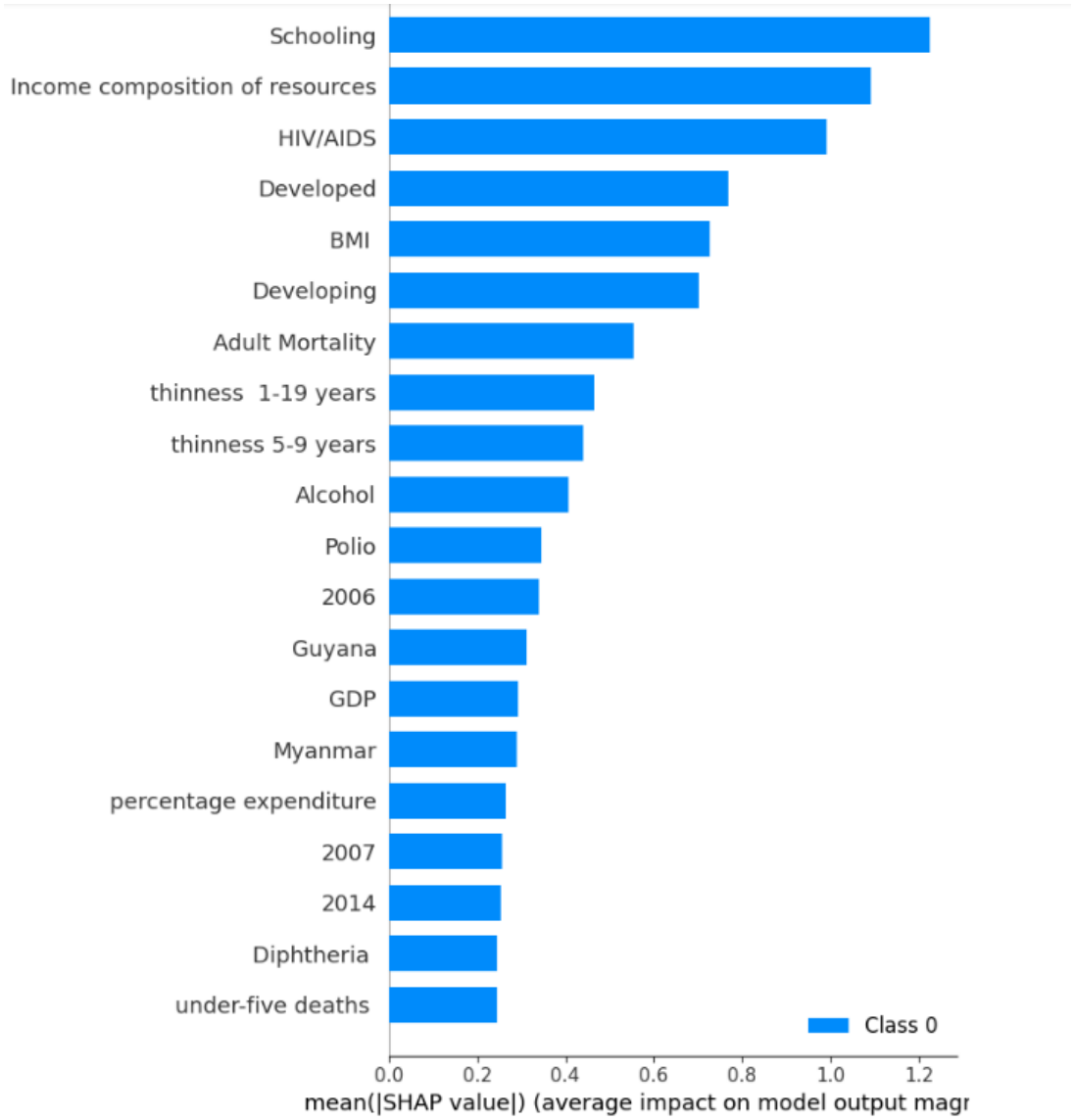
شکل ۴: فرایند آموزش مدل عصبی برای رگرسیون

با معیار **r2_square** عملکرد مدل بر روی مجموعه تست ۹۴ درصد شد. نمونه‌ای از پیشبینی‌ها برای اطمینان صحت عملکرد مدل را در شکل زیر مشاهده می‌کنید. مدل با تابع هزینه **MSE** و اپتیمایزر **Adam** آموزش دید و در نهایت **train loss** برابر با ۱.۶۴ و **test loss** برابر با ۶.۳۲ به دست آمد.



شکل ۵: پیشبینی مدل برای چند نمونه مجموعه تست

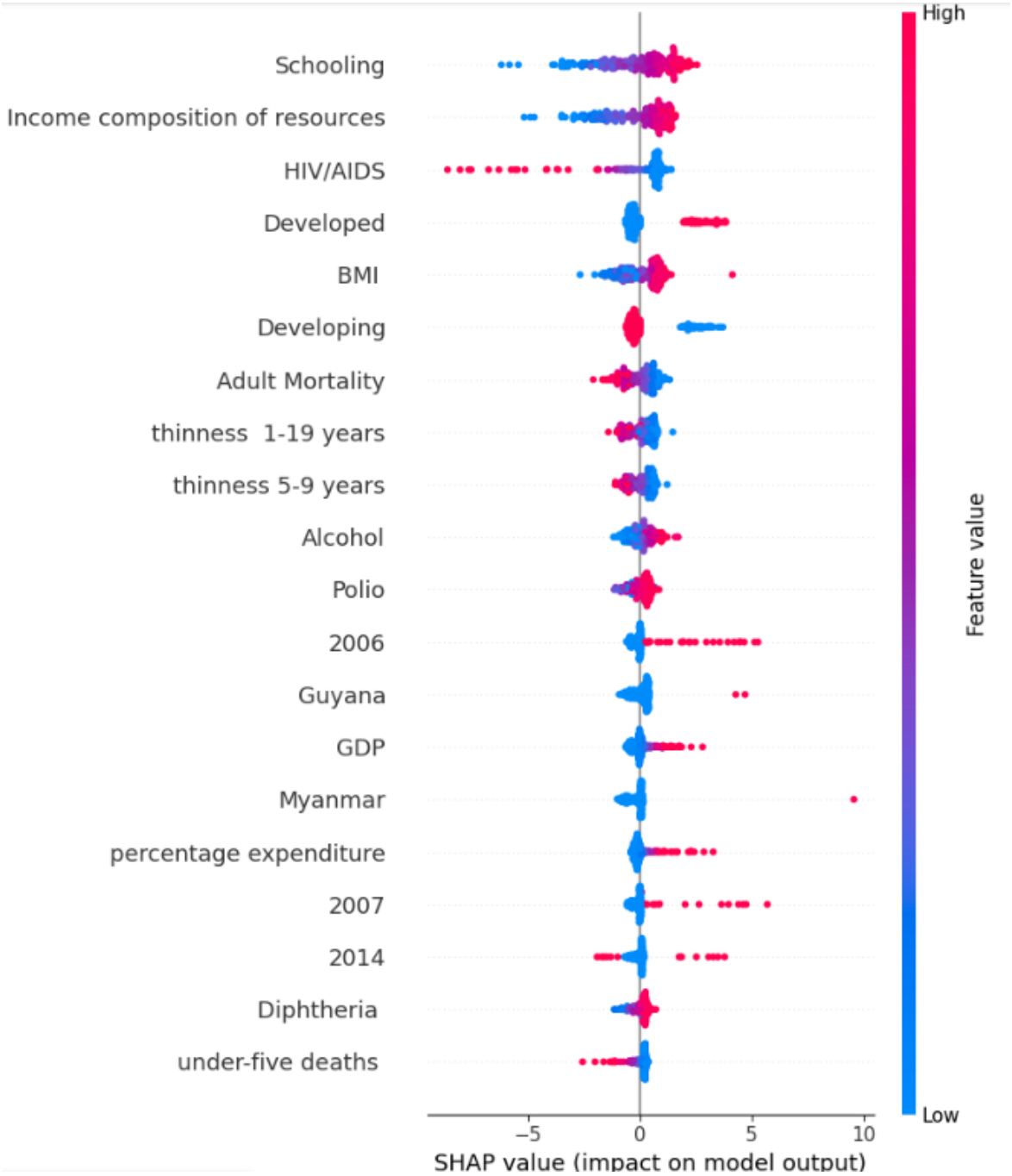
رسم نمودار summary_plot (“bar”) برای تفسیر مدل عصبی با Deep SHAP و نمایش اهمیت هر ویژگی که در شکل زیر نشان داده شده است:



شکل ۶: رسم **summary_plot ("bar")** برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Deep SHAP

ویژگی‌ها به شکل نزولی بر اساس اهمیت هر ویژگی در شکل بالا نشان داده شده‌اند. مشاهده می‌شود که ویژگی‌های income composition of reources، adult mortality، HIV/AIDS، Thinness 1-19 years، under five deaths، Developed، Developing و Schooling بیشترین تاثیر را در پیشبینی سن امید به زندگی دارند.

رسم نمودار summary_plot (“dot”) برای تفسیر مدل عصبی با Deep SHAP:

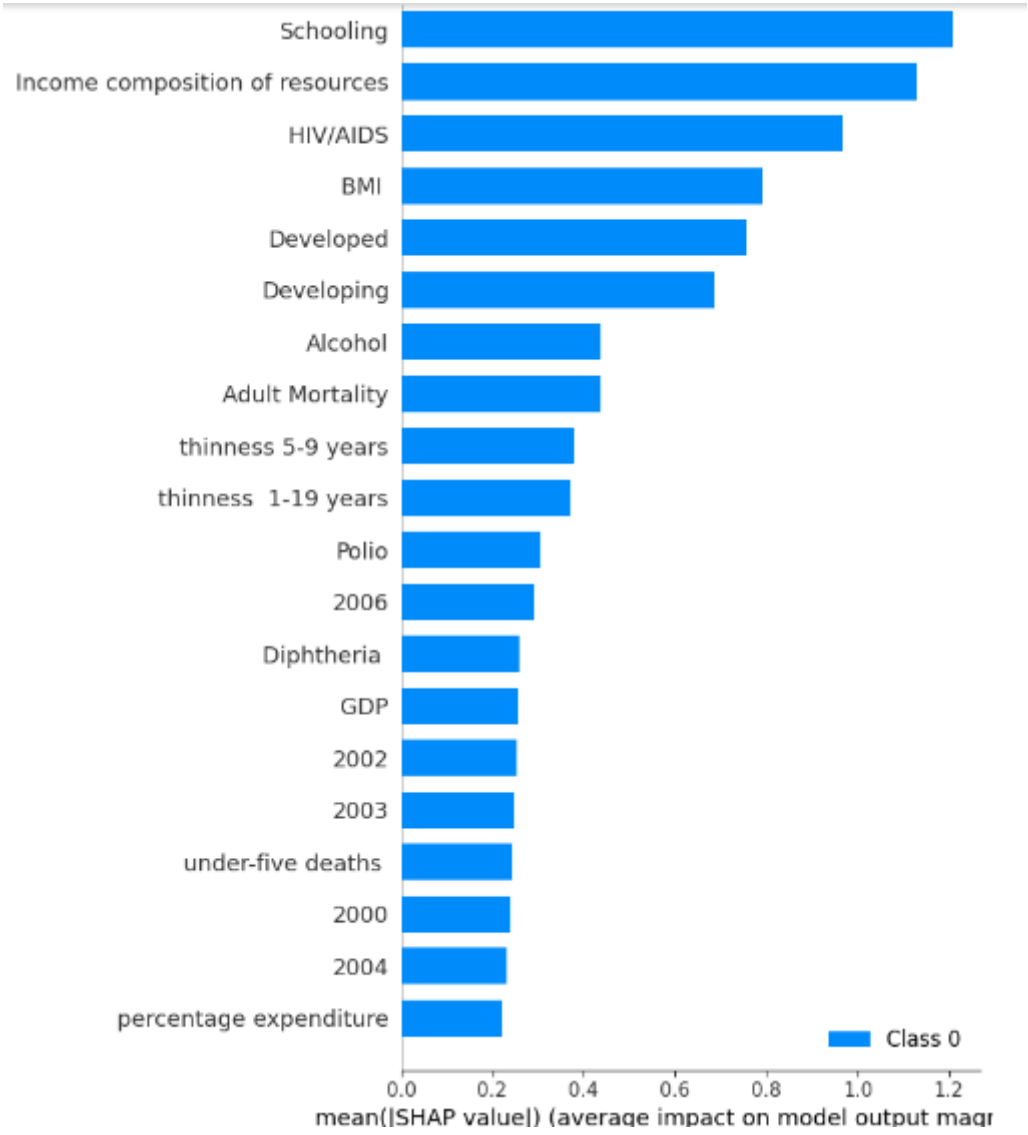


شکل ۷: رسم **summary_plot** (**"dot"**) برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Deep SHAP

در نمودار بالا مشاهده می‌شود مقادیر Value کم (آبی‌تر) برای ویژگی مانند Adult Mortality باعث افزایش سن امید به زندگی و مقادیر Value زیادش (قرمزتر) باعث کاهش سن امید به زندگی می‌شود. هرچه Income Composition of resourses، Schooling، BMI، Percentage Expedinture بیشتر باشد سن امید به زندگی بیشتر است و کمتر بودن آن‌ها تاثیر منفی بر سن امید به زندگی دارد. همچنین

مشاهده می‌شود یک بودن ویژگی Developed (یعنی کشورهای پیشرفته‌تر) در بیشتر شدن سن امید به زندگی تاثیر زیادی دارد و یک بودن ویژگی Developing (کشورهای در حال توسعه) تاثیر منفی در سن امید به زندگی دارد. همچنین برای ویژگی‌های HIV/AIDS، Thinness 1-19، Thinness 5-9 هرچه مقادیر value آن‌ها کمتر باشد تاثیر مثبتی بر سن امید به زندگی وجود دارد و زیاد بودن آن‌ها تاثیر منفی بر آن دارد. با بررسی Skewness نیز متوجه می‌شویم بیشتر کشورها Schooling، income composition of resources و BMI بالایی دارند و همچنین Status بیشتر کشورها Developing است و کشورهای کمتری Developed می‌باشند.

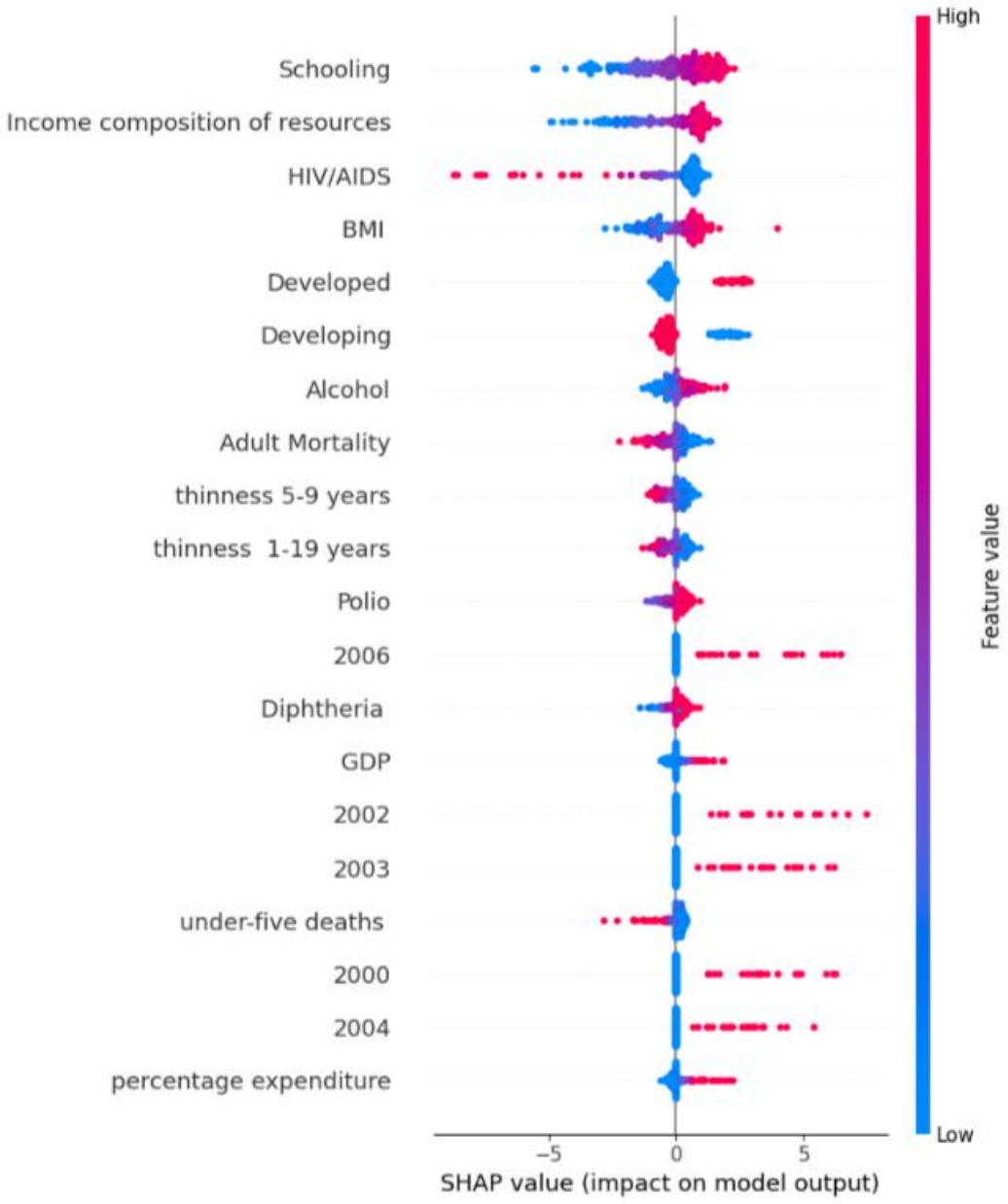
رسم نمودار summary_plot (“bar”) برای تفسیر مدل عصبی با Kernel SHAP و نمایش اهمیت هر ویژگی که در شکل زیر نشان داده شده است:



شکل ۸ رسم **summary_plot ("bar")** برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Kernel SHAP

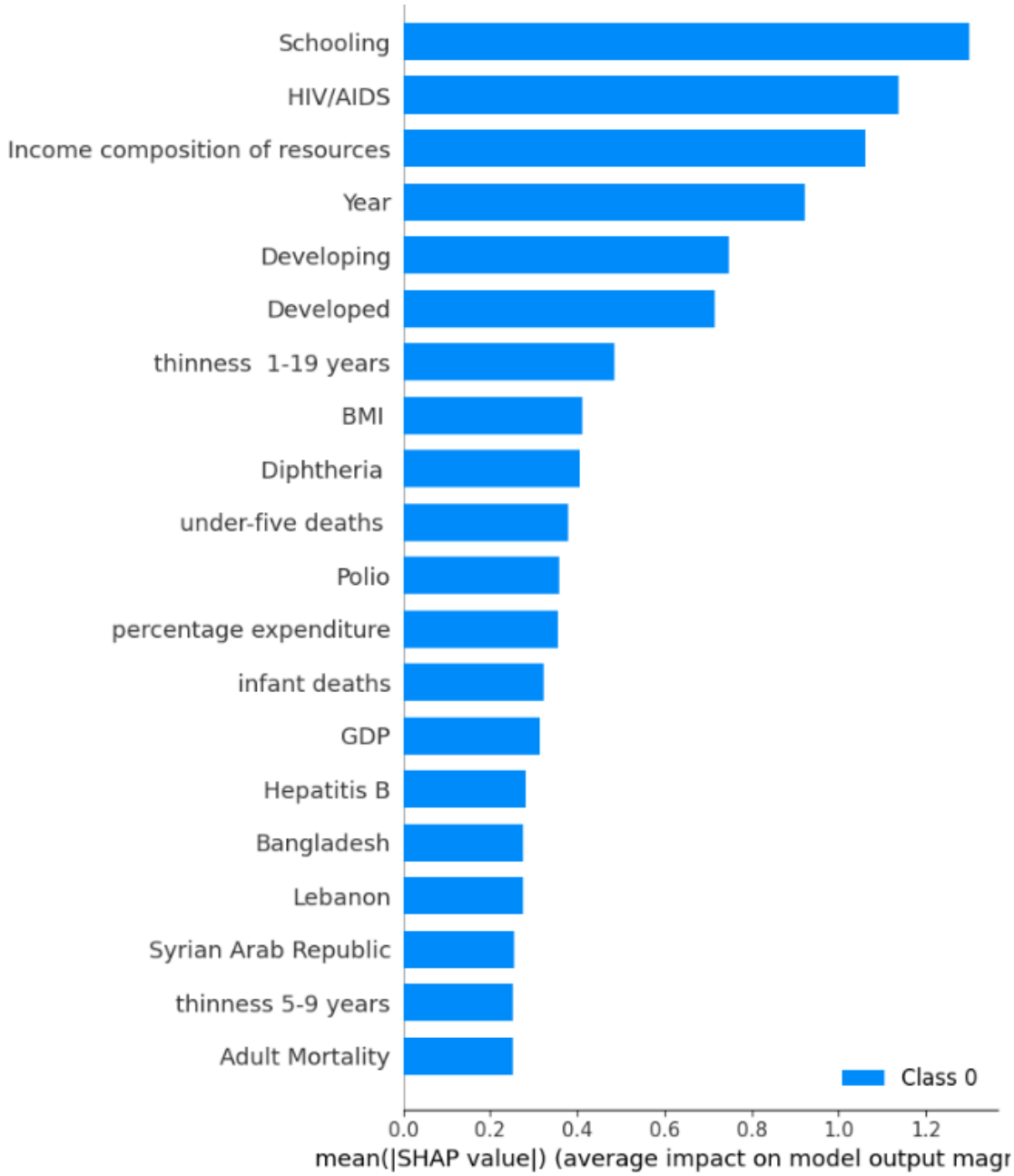
مشاهده می‌شود که ویژگی‌های مهم به مقدار خیلی کمی نسبت به روش Deep SHAP جا به جا شده‌اند. برای مثال BMI بالای Adult Mortality قرار گرفته است و ویژگی Alcohol در روش Kernel SHAP در مقایسه باDeep SHAP اهمیت بیشتری نسبت به ویژگی‌های Thinness 1-19 years و under-five deaths و Adult Mortality دارند ولی در کل Kernel SHAP و Deep SHAP مانند یکدیگر عمل کردند.

رسم نمودار summary_plot (“dot”) برای تفسیر مدل عصبی با Kernel SHAP:



شکل ۹: رسم **summary_plot ("dot")** برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Kernel SHAP

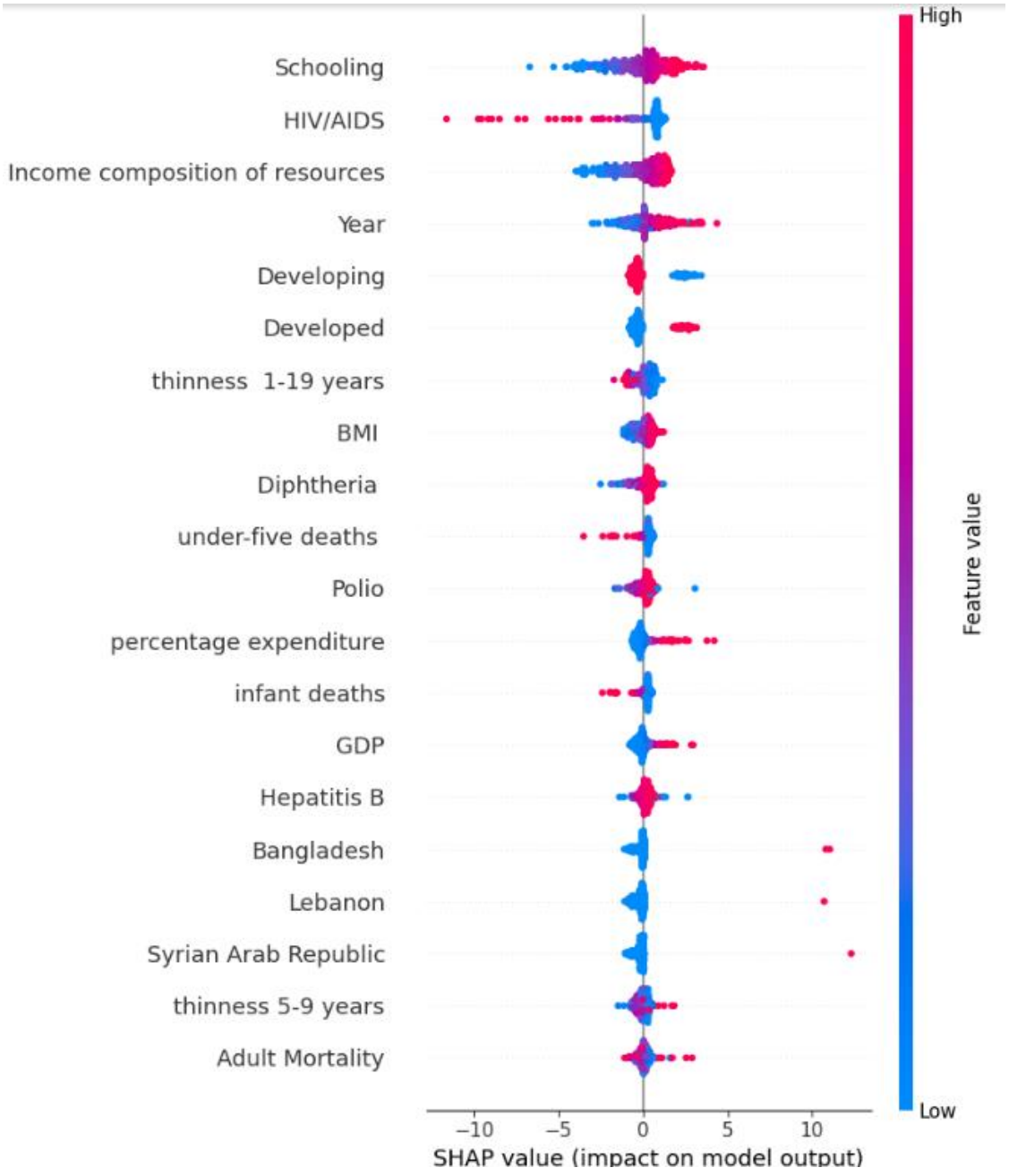
رسم نمودار summary_plot (“bar”) برای تفسیر مدل شبکه عصبی با Deep SHAP و در نظر گرفتن ویژگی Year به صورت Numerical. نمایش اهمیت هر ویژگی که در شکل زیر نشان داده شده است:



شکل ۱۰: **summary_plot ("bar")** برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Deep SHAP

مشاهده می‌شود که اگر ویژگی Numerical Year در نظر گرفته شود جزو ویژگی‌های تاثیرگذار شناسایی می‌شود.

رسم نمودار summary_plot (“dot”) برای تفسیر مدل شبکه عصبی با Deep SHAP و در نظر گرفتن ویژگی سال به صورت Numerical:

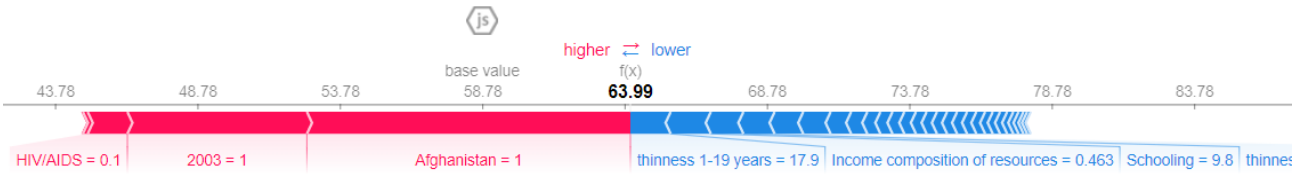


شکل ۱۱: **summary_plot ("Dot")** برای مدل شبکه عصبی و تمام نمونه‌های تست و تمام ویژگی‌های مدل به کمک Deep SHAP

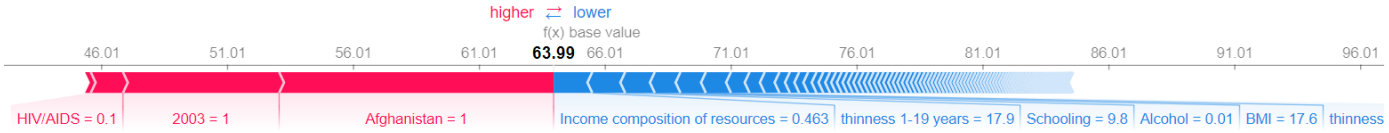
همانطور که مشاهده می‌شود با افزایش سال سن امید به زندگی نیز افزایش می‌یابد و با منطق اینکه در طی گذر سال‌ها به دلیل پیشرفت تکنولوژی سن امید نیز افزایش می‌یابد مطابقت دارد.

سپس برای دو کشور ژاپن و افغانستان نمودار force_plot را به صورت زیر به دست آوردیم:

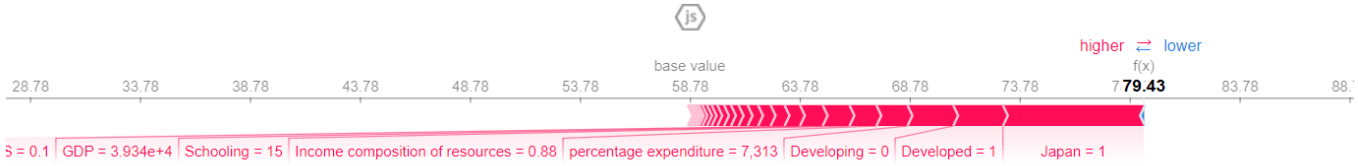
F(x) پیشبینی مدل و Base Value میانگین پیشبینی بر روی کل دیتاست است. فیچرهایی که باعث افزایش مقدار پیشبینی شده می‌شوند به رنگ قرمز و فیچرهایی که باعث کاهش آن می‌شوند به رنگ آبی هستند. سن امید به زندگی برای کشور افغانستان در نمونه انتخاب شده ۵۹.۵ است که شبکه ۶۳.۹۹ پیشبینی کرده است. همچنین برای ژاپن ۸۳ است که ۷۹ پیشبینی شده است.



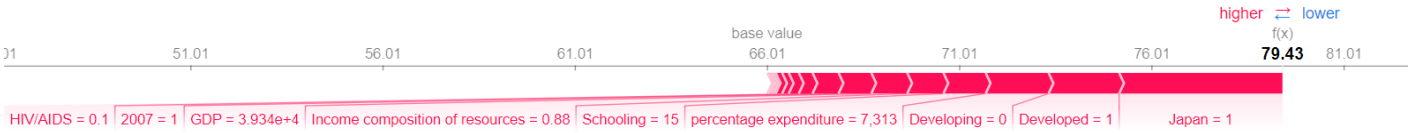
شکل ۱۲: نمودار **force_plot** برای کشور افغانستان با **Deep SHAP**



شکل ۱۳: نمودار **force_plot** برای کشور افغانستان با **Kernel SHAP**



شکل ۱۴: نمودار **force_plot** برای کشور ژاپن با **Deep SHAP**



شکل ۱۵: نمودار **force_plot** برای کشور ژاپن با **Kernel SHAP**

با این نمودارها نتیجه‌گیری می‌شود برای کشور ژاپن تقریبا همه مقادیر ویژگی‌ها باعث افزایش سن امید به زندگی شده و برای کشور در حال توسعه افغانستان دقیقا برعکس و اکثر ویژگی‌ها تاثیر منفی بر سن امید به زندگی دارند. هرچه تاثیر ویژگی برای نمونه بیشتر باشد فلش آن عرض بیشتری دارد و مقادیر Value ویژگی‌ها نیز در زیر آن فلش‌ها نوشته شده است. در این دو شکل نیز مشاهده می‌شود Deep

SHAP و Kernel SHAP تفاوت خیلی زیادی در شناسایی تاثیر ویژگی‌ها برای این دو نمونه ندارند و صرفا کمی ترتیب اهمیت ویژگی‌ها جا به جا شده است.

پرسش ۲ – Knowledge Distillation

۱) خلاصه ای از مزایای مدل معرفی شده در این مقاله نسبت به شبکه های عصبی را بیان کنید و علت آن بیان کنید.

مدل معرفی شده در این مقاله، درخت تصمیم گیری نرم آموزش داده شده با استفاده از شبکه عصبی است که دارای مزایایی نسبت به شبکه های عصبی است. یکی از مزیت‌هایش این است که درخت تصمیم نرم تفسیرپذیرتری بیشتری را ارائه می‌کند و به ما این امکان را می‌دهد تا بهتر درک کنیم که مدل چگونه Classification کرده است. شبکه های عصبی عمیق در انجام وظایف Classification موثر هستند مخصوصا زمانی که داده‌های ورودی ابعاد بالایی دارند یا رابطه بین ورودی و خروجی پیچیده است و یا تعداد نمونه‌های آموزشی با لیبل زیاد است. ولی تفسیر عملکرد شبکه‌های عصبی در Classification ممکن است دشوار باشد زیرا محاسبات پیچیده‌ای در لایه‌های نورون‌ها وجود دارد. اما تفسیر درخت تصمیم چون ساده‌تر است راحت‌تر می‌باشد ولی نمی‌تواند دقت بالایی فراهم کند و به دلیل اینکه نودهای میانی تنها بخشی از داده‌ها را می‌بینند ممکن است به overfit منجر شود. هدف اصلی این مقاله این است که دانشی که از شبکه عصبی به دست آمده و در نتیجه تعمیم‌پذیری بالا را به درخت تصمیم کوچک انتقال دهد. در واقع درخت تصمیم نرم را با Stochastic Gradient Descent و پیشبینی-های شبکه عصبی می‌سازد. این درخت تصمیم نرم از فیلترهای یادگرفته شده استفاده می‌کند تا بر اساس داده ورودی تصمیمات سلسله مراتبی بگیرد و به عنوان خروجی توزیع احتمال کلاس را نتیجه دهد. این مدل نسبت به درختی که از خود داده آموزش می‌بیند تعمیم‌پذیری بیشتری دارد ولی نسبت به خود شبکه عصبی تعمیم‌پذیری‌اش کمتر ولی سرعتش بیشتر و نیازمند حافظه کمتر است. پس اگر لازم باشد پیشبینی Classification تفسیر شود می‌توان از این درخت تصمیم استفاده کرد و همچنین مزایای شبکه‌های عصبی عمیق را در آن اعمال کرد. یکی دیگر از مزایای درخت تصمیم Roubostتر است و نسبت به داده‌های نویزی مقاومت بیشتری نشان می‌دهد.

۲) چگونه این مدل به جای یک سلسله از ویژگی ها (hierarchy of features)، با یک سلسله از تصمیم‌ها (hierarchy of decisions) کار می‌کند؟

هدف این مقاله ساخت مدلی است که به راحتی قابل تفسیر باشد و به همین دلیل بر خلاف شبکه عصبی که بر سلسله‌ای از ویژگی‌ها متکی بود بر سلسله‌ای از تصمیمات متکی باشد. برای رسیدن به این هدف، درخت تصمیم گیری نرم از فیلترهای آموخته شده (w_i,b_i) برای تصمیم گیری سلسله مراتبی بر اساس یک مثال ورودی استفاده می‌کند. این فیلترها با استفاده از Stochastic Gradient Descent آموزش داده

شده و برای انتخاب یک توزیع احتمال خاص بر روی کلاس‌ها به عنوان خروجی استفاده می‌شوند. درخت تصمیم نرم در هر مسیر کل ویژگی‌ها را در نظر می‌گیرد. این سلسله از فیلترها هر نمونه را به یک biggot، تخصیص می‌دهد (مدل ترکیب سلسله مراتبی از expertها است [Jordan and Jacobs, 1994] که هر expert یک bigot است که بعد از آموزش به داده نگاه نمی‌کند و همیشه یک توزیع را تولید می‌کند) و هر bigot توزیع آماری ساده برروی کلاس‌های خروجی یاد می‌گیرد. با تصمیم گیری در سطح انتزاع، درخت تصمیم نرم می تواند روابط پیچیده بین ورودی‌ها و خروجی‌ها را بهتر به دست آورد. در واقع برای اینکه متوجه شد چرا مدل پیشبینی خاصی کرده است می‌توان فیلترهای یاد گرفته شده در مسیر ریشه تا برگ Classification را بررسی کرد. احتمال انتخاب branch راست:

$$p_i(\mathbf{x}) = \sigma(\mathbf{x}\mathbf{w}_i + b_i)$$

توزیع احتمالاتی expertها: (Q^l_k توزیع احتمالاتی در برگ l و فی پارامتر یاد گرفته شده در برگ است)

$$Q_k^\ell = \frac{\exp(\phi_k^\ell)}{\sum_{k'} \exp(\phi_{k'}^\ell)},$$

شیوه تصمیم‌گیری به دو صورت استفاده از مسیر احتمالاتی برگ با بالاترین احتمال مسیر و یا میانگین از توزیع احتمالاتی کل برگ‌ها است.

۳) در رابطه با تابع هزینه مدل بحث کنید و تفاوت آن ها را با تابع هزینه cross-entropy مقایسه کنید.

Cross-entropy تابع هزینه‌ای است که تفاوت بین دو توزیع احتمال را اندازه‌گیری می‌کند. در Classification و اغلب برای اندازه‌گیری تفاوت بین توزیع احتمال پیش بینی شده و توزیع احتمال واقعی استفاده می‌شود. با به حداقل رساندن cross-entropy loss، مدل یاد می‌گیرد که پیش بینی‌های خود را با تنظیم پارامترهای خود بهبود بخشد تا توزیع پیش بینی شده به توزیع واقعی نزدیک شود. به جای محاسبه یک مقدار cross-entropy برای همه کلاس‌های خروجی که در شبکه‌های عصبی انجام می‌شود، آموزش درخت تصمیم نرم با به حداقل رساندن cross-entropy بین هر برگ (وزن شده بر اساس احتمال مسیر آن) و توزیع هدف (T) انجام می‌شود. این تابع هزینه نسبت به cross-entropy بار محاسباتی بیشتری دارد. مقدار تابع هزینه برای ورودی x:

$$\mathrm{L}(\mathbf{x}) = -\log\left(\sum_{\ell \in LeafNodes} P^{\ell}(\mathbf{x}) \sum_{k} T_k \log Q_k^{\ell}\right)$$

P احتمال تعلق نمونه ورودی به نود برگ l، T توزیع هدفمان و Q توزیع خروجی مدل است. دلیل استفاده از log برای این جمع وزن‌دار احتمالا این است که با افزایش تعداد برگ‌ها افزایش نمایی خواهیم داشت. با این تابع هزینه درخت تصمیم می‌آموزد رفتار شبکه‌های عمیق را تقلید کند ولی اگر تفسیرپذیری دغدغه اصلی نباشد می‌توان از همان cross-entopy و شبکه عمیق استفاده کرد.

۴) علت اضافه کردن ترم regularization در این مدل چیست؟

دلیل اضافه کردن آن جلوگیری از گیر افتادن در راه‌حل های ضعیف در طول آموزش است. بدون regularization، درخت تمایل دارد در فلات‌هایی گیر کند که در آن یک یا چند گره داخلی تقریباً همیشه تمام احتمالات را به یکی از زیر درخت‌های چپ یا راست خود اختصاص می‌دادند. این باعث شد که گرادیان لجستیک بسیار نزدیک به صفر باشد و یادگیری مدل را دشوار کند. اگر آلفا به صورت زیر تعریف شود:

$$\alpha_i = \frac{\sum_{\mathbf{x}} P^i(\mathbf{x}) p_i(\mathbf{x})}{\sum_{\mathbf{x}} P^i(\mathbf{x})}$$

P^i احتمال رسیدن به نود i و p_i احتمال رفتن به زیردرخت راست است. می‌خواهیم توزیع احتمالی α و 1-α نزدیک به ۵۰ ۵۰ باشد و هر نود داخلی . ترم C با استفاده از cross-entropy این هدف را اعمال می‌کند و لاندا اهمیت این ترم را اعمال می‌کند:

$$C = -\lambda \sum_{i \in InnerNodes} 0.5 \log(\alpha_i) + 0.5 \log(1 - \alpha_i)$$

این ترم هر گره داخلی را تشویق می‌کند تا از هر دو درخت فرعی چپ و راست استفاده کند. همچنین به جلوگیری از گیرکردن در فلات‌ها کمک می‌کند و توانایی آن را برای تعمیم خوب بهبود می‌بخشد.

پرسش ۳ – D-RISE

در این سوال قصد داریم تا به بررسی Object Detectorها با استفاده از Saliency Mapها بپردازیم. بدین منظور ما مقاله D-RISE را انتخاب کرده‌ایم.

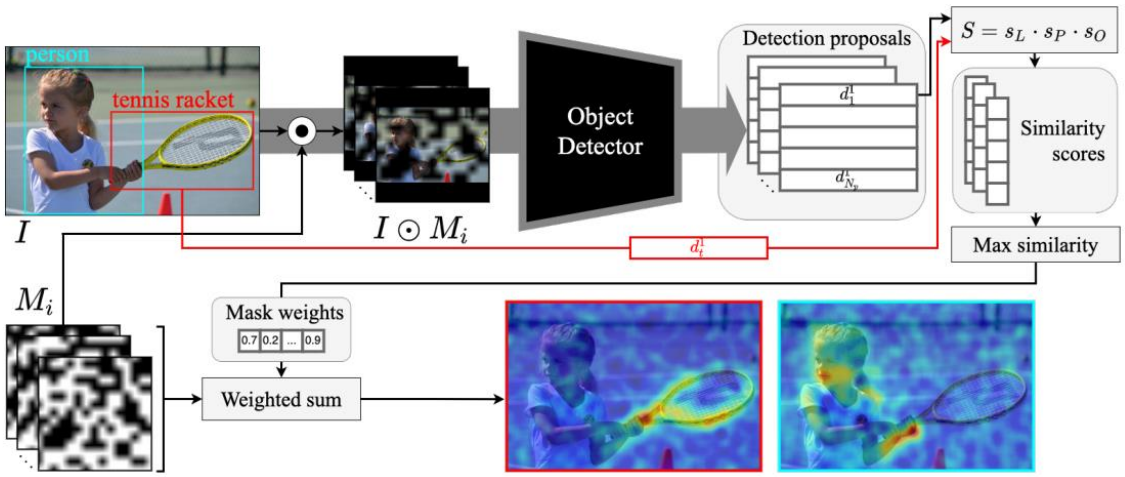
a) ابتدا مقاله را مطالعه کرده و یک خلاصه‌ای از ایده کلی و متمایز کننده آن نسبت به روش‌های دیگر ارائه دهید. همچنین بیان کنید که علت دست یافتن به این روش چه بوده است و روش‌های مشابه مبتنی بر شبکه‌های عصبی چه مشکلاتی داشته‌اند.

ایده کلی پشت D-RISE ارائه تفسیر تصویری مبتنی بر Saliency Map برای مدل‌های تشخیص اشیاء یا Object-Detectorها است. این روش از یک متریک تشابه منحصر به فرد استفاده می‌کند که جنبه‌های Localization و Categorization تشخیص اشیاء را در نظر می‌گیرد تا Saliency Map تولید کند که مناطقی از تصویر که بیشترین تأثیر بر پیش‌بینی مدل دارند را برجسته می‌کند. در مقایسه با سایر روش‌های مبتنی بر گرادیان برای تولید توضیحات بصری، D-RISE قابل تعمیم‌تر است و نسبت به مدلی که بر روی آن اعمال می‌شود Agnostic است و نیاز به عملکرد و معماری مدل ندارد. در نتیجه می‌تواند در بسیاری از مدل‌های تشخیص اشیاء اعمال شود. مشکل اصلی روش‌های مشابه مبتنی بر شبکه‌های عصبی عدم تفسیرپذیری و شفافیت آن‌ها است. تفسیر و اشکال‌زدایی این روش‌ها دشوار و شناسایی منابع خطا یا بایاس در رفتار مدل سخت است. هدف D-RISE غلبه بر این محدودیت‌ها با ارائه توضیحات بصری تفسیرپذیر است که می‌تواند به کاربران در درک بهتر نحوه پیش‌بینی مدل کمک کند. در مقاله گفته شده است که با روش آن‌ها برای مثال در تشخیص لپ‌تاپ، لوگوی سیب تاثیر بیشتری داشته و مشخص می‌شود خیلی وقت‌ها اطلاعات خارج Bounding BOX به تشخیص اطلاعات داخل آن کمک می‌کند.

در مقاله گفته شده تکنیک‌های قبلی مانند Attribution که در تسک Classification، Saaliency Map تولید می‌کردند (مانند Grad-Cam) و معمولا به معماری مدل وابسته هستند برای تسک تشخیص اشیاء مناسب نیستند. همچنین روش‌های دیگر که برای تفسیر مدل‌های تشخیص اشیاء مبتنی بر شبکه عصبی ارائه شده‌اند برعکس D-RISE که به صورت Black-Box با مدل کار می‌کند، مبتنی بر روش‌های White-Box هستند. این روش مثلا به سادگی می‌تواند برای Object Detectorهای مختلف مانند YOLOv3 و Faster-RCNN که به ترتیب یک مرحله‌ای و دو مرحله‌ای می‌باشند اعمال شود.

روش این مقاله به این صورت است که تشخیص Targetهایی که می‌خواهیم تفسیر کنیم را به بردار d_t برده و Nتا ماسک باینری نمونه می‌گیریم. سپس با Object Detector و تصاویر ماسکه شده پروپوزال‌های

D_p را نتیجه می‌گیریم. در مرحله بعد شباهت بین Target و پروپوزال را برای محاسبه وزن ماسک‌ها به دست آورده و از جمع وزن‌دار این ماسک‌ها به Saliency Map می‌رسیم. در شکل بعدی این روش نشان داده شده است:



شکل ۱۶: روش **D-RISE**

b) الگوریتم استفاده شده در این مقاله برای تولید Mask را بیان کنید و توضیح دهید.

الگوریتم مورد استفاده در این مقاله برای تولید ماسک یا Mask Generation شامل سه مرحله است:

۱. از N ماسک دودویی با اندازه $h \times w$ (کوچکتر از اندازه تصویر $H \times W$) با تنظیم هر عنصر به طور مستقل بر روی ۱ با احتمال p و ۰ با احتمال باقی مانده، نمونه می‌گیرد.

۲. همه ماسک‌ها را به اندازه C_W (w+1) $\times$ C_H (h+1) با استفاده از درون‌یابی Upsample Bilinear می‌کند، که در آن $C_H \times C_W$ اندازه سلول در ماسک Upsample شده است.

۳. بریدن مناطق $H \times W$ با آفست رندوم یکدست با دامنه‌ای از (0,0) تا (C_H, C_W)

این رویکرد از مقاله RISE گرفته شده است. در این مقاله گفته شده ماسکه کردن پیکسل‌ها به شکل مستقل ممکن است تاثیر Adversary داشته باشد و یک تغییر جزئی بر Confidence Score مدل تغییرات ایجاد کند. همچنین ماسکه کردن به شکل مستقل یک فضای ماسکه کردن 2^{W+H} خواهد داشت و فضای بزرگتر برای تخمین خوب به نمونه‌های بیشتری نیاز دارد. برای این مشکل ابتدا ماسک‌های باینری کوچک‌تر نمونه‌برداری شده و سپس با Bilinear Interpolation به رزولوشن بالاتر upsample می‌شود. Bilinear Upsampling لبه‌های تیز تولید نمی‌کند. بعد از Interpolation ماسک‌ها باینری نیستند و مقادیری بین ۰ و ۱ دارند. همچنین برای انعطاف بیشتر ماسکه کردن، ماسک پیکسل‌ها در هردو جهت مکانی شیف با مقدار رندوم شیفت داده می‌شوند.

c) معیار Similarity استفاده شده در این مقاله را توضیح دهید و علت انتخاب این روش را به اختصار توضیح دهید.

برای محاسبه Similarity Score بین بردارهای target و proposal و رسیدن به وزن ماسک‌ها برای ایجاد Saliency Map سه مولفه باید در نظر گرفته شوند. در مولفه اول از Intersection over Union برای اندازه‌گیری همجواری مکانی Bouning Boxها که توسط دو بردار انکد شده‌اند استفاده شده است. برای سنجش اینکه چقدر دو ناحیه شبیه به هم هستند از شباهت کسینوسی احتمال کلاس‌های مربوط به ناحیه‌ها بهره برده‌اند. نهایتا برای شبکه‌هایی که مستقیما Objective Scoreرا محاسبه می‌کنند (مانند YOLOv3) معیاری از شباهت Objective Score را نیز در متریک ضمیمه کردند. اگر مدلی Objective Score تولید نکند (مانند Faster R-CNN) بخش Objectness می‌تواند حذف شود. سه مولفه:

$$
\begin{aligned}
s_L(d_t, d_j) &= \mathrm{IoU}(L_t, L_j), \\
s_P(d_t, d_j) &= \frac{P_t \cdot P_j}{\|P_t\| \|P_j\|}, \\
s_O(d_t, d_j) &= O_j.
\end{aligned}
$$

$$
s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j),
$$

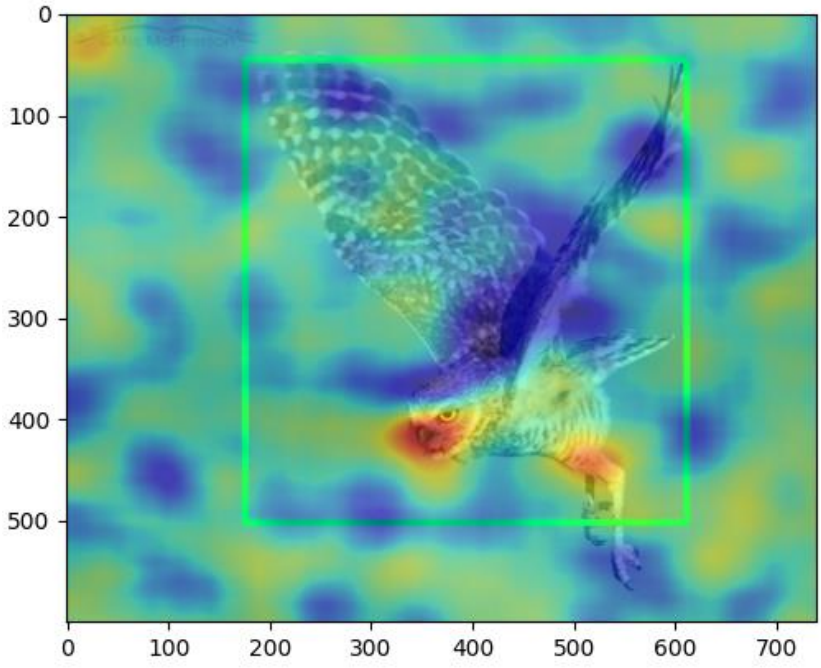
ضرب اسکالر برای این است که اگر یکی از مقادیر شباهت کم بود کل شباهت نیز کم باشد.

d) اختیاری

(e

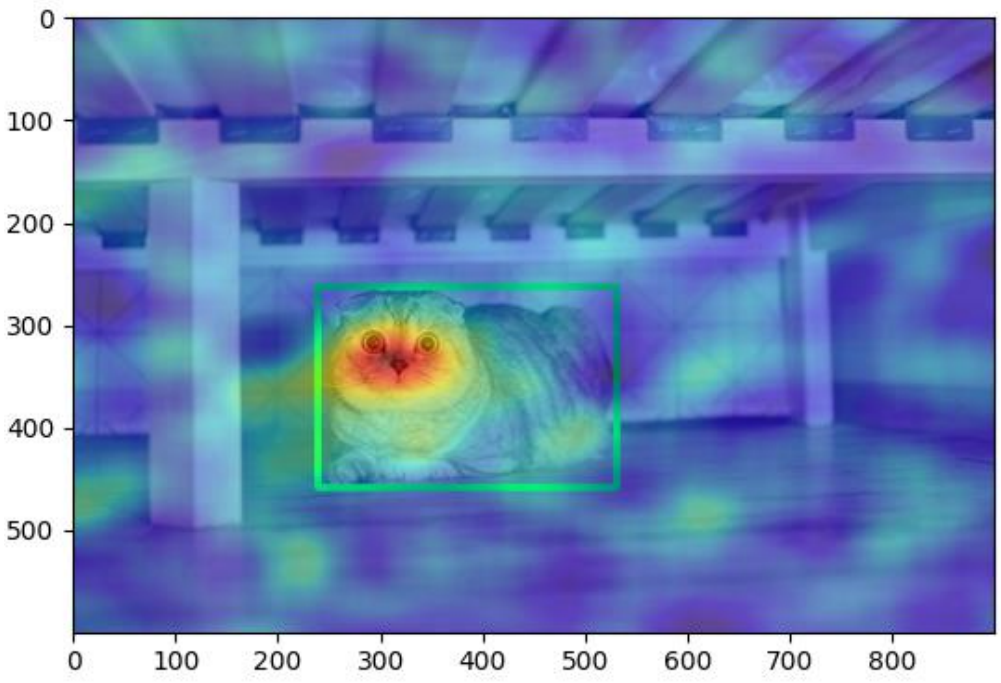
در این بخش با گوگل کلب و روش ارائه شده در گروه تلگرام تصویر Saliency را نتیجه گرفتیم. برای اجرای کد باید در فایل py, آدرس عکس ورودی مشخص شود و همچنین cellها به ترتیب اجرا شود. در نهایت عکس Saliency Map به نام Figue در content ذخیره شده و قابل دانلود خواهد بود.

شکل ۱۷: تصویر اول ورودی به مدل با کلاس **Bird**



شکل ۱۸: تشخیص **Saliency** برای تصویر اول کلاس **Bird**

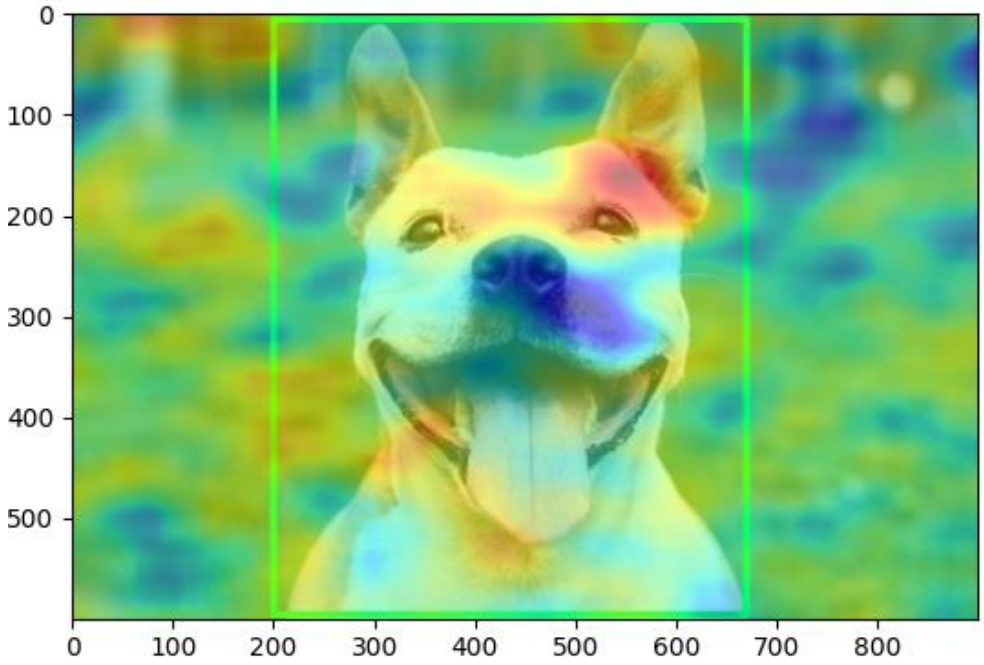
شکل ۱۹: تصویر دوم ورودی به مدل با کلاس **Cat**



شکل ۲۰: تشخیص **Saliency** برای تصویر دوم کلاس **Cat**

```
15 cat (239, 262, 531, 458) 0.99466974
```

شکل ۲۱: تصویر سوم ورودی به مدل با کلاس **Dog**



شکل ۲۲: تشخیص **Saliency** برای تصویر سوم کلاس **Cat**

```
16 dog (200, 5, 670, 595) 0.9923299
```

پرسش ۴ – LIME

در این پرسش قصد داریم تا با ساز و کار و نحوه عملکرد LIME آشنا شویم.همانطور که از اسم آن مشخص است این روش Model-agnostic هست و مدل را یک موجودیت black-box در نظر می‌گیرد. بنابراین از این روش می‌توان برای تفسیر هر مدل یادگیری ماشینی بهره برد.

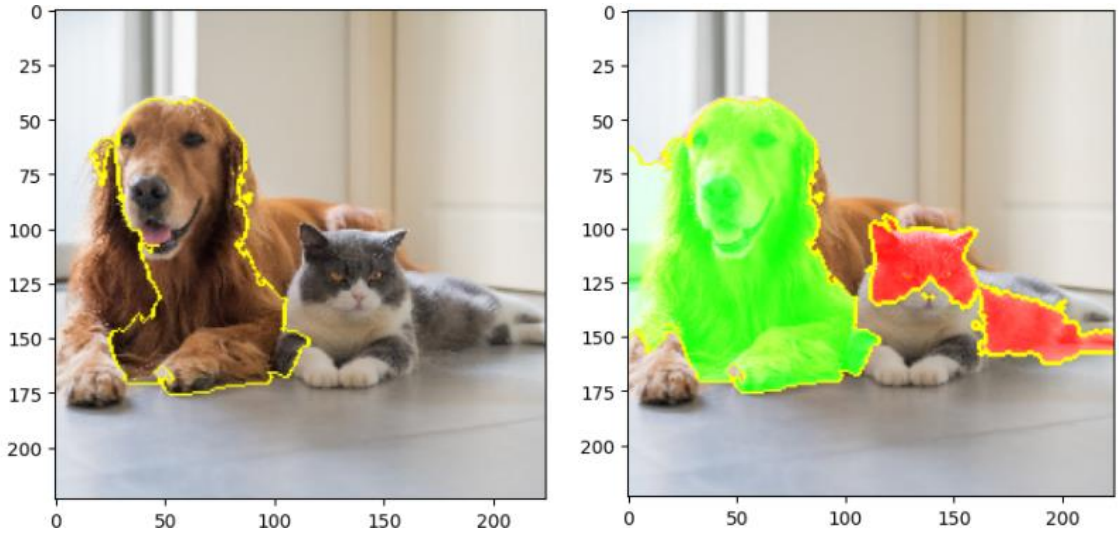
در این بخش از مدل MobileNet V2 در فریمورک تنسورفلو که با مجموعه‌داده image-net آموزش دیده استفاده کردیم تا کار Classification بر روی تصاویر دلخواه انجام داده و سپس دلیل پیشبینی مدل را تفسیر کنیم. سپس با استفاده از lime_image و mark_boundaries از skimage، boundary را بر روی تصاویر رسم کردیم. برای بررسی تاثیر super pixelها بر پیشبینی مدل نواحی Pros and cons و در انتها Heatmap را نملیش دادیم (موارد خواسته شده از a تا h). تعداد نمونه‌های مغتشش تولید شده از تصویر ورودی مدل ۱۰۰۰۰تا در نظر گرفته شد. تصاویر به ابعاد ۲۲۴ در ۲۲۴ resize شد و پس از تبدیل آن به آرایه پیش پردازش‌های اولیه برای ورود به مدل با تابع `preprocess_input` انجام شد. روش LIME در سوال یک توضیح داده شده است. **تصویر اول (سگ و گربه)**:

شکل ۲۳: تصویر سگ و گربه

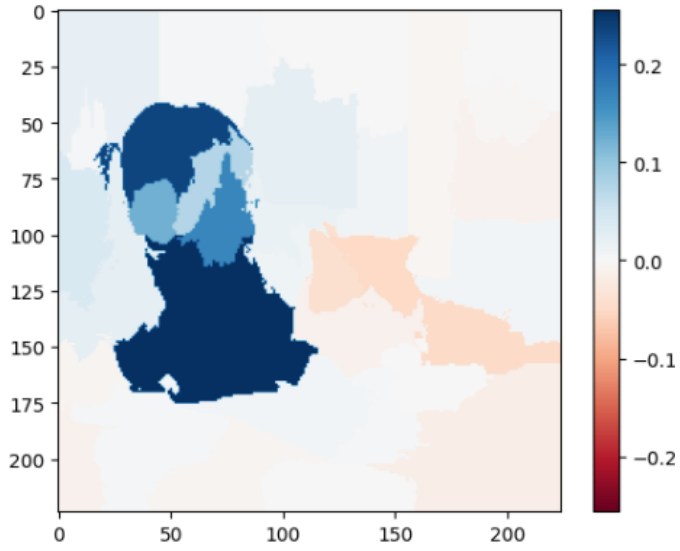
پنج کلاس پیشبینی شده و احتمال آن‌ها:

```
golden retriever: 0.6620
cocker spaniel: 0.0471
Irish setter: 0.0292
Welsh springer spaniel: 0.0067
Sussex spaniel: 0.0055
```

کلاس با بیشترین احتمال سگ golden retriever تشخیص داده شده است (نژاد سگ داخل عکس همین است) و بقیه کلاس‌ها نیز سگ‌ها با نژادهای مختلف هستند.



شکل ۲۴: نواحی **Boundaries** در شکل چپ و **Pros and Cons** در شکل سمت راست برای تصویر سگ و گربه



شکل ۲۵: **Heatmap** برای تصویر سگ و گربه

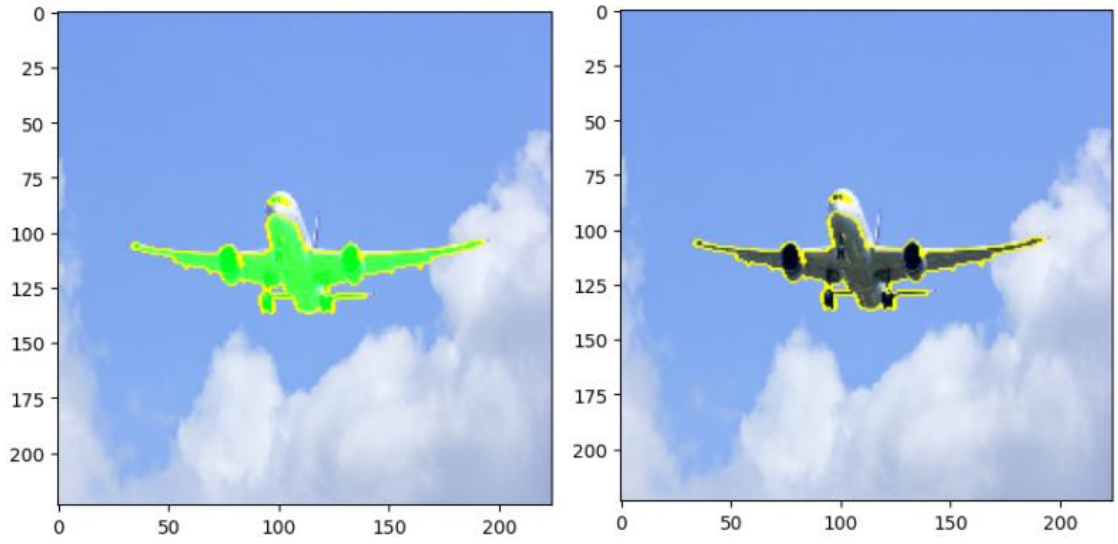
تحلیل هر سه شکل منطبق و تکمیل‌کننده یکدیگر هستند ولی Heatmap دارای طیف رنگی بیشتری است و Superpixelها را دقیق‌تر نشان می‌دهد و همچنین نسبت به pros and cons اطلاعات بیشتری ارائه می‌دهد. قسمت‌های سبز در pros and cons در پیشبینی مدل تاثیر مثبت داشته و قسمت‌های قرمز تاثیر منفی بر آن داشته‌اند. با توجه به این که پیکسل‌های قسمت قرمز مربوط به کلاس گربه هستند بدیهی است که تاثیر منفی در پیشبینی کلاس سگ golden retriever بگذارند و قسمت سبز نیز دلیل اصلی این پیشبینی است. همچنین تعداد ویژگی برای تفسیر پنج‌تا است. در شکل heatmap هرچه تاثیر در خروجی مدل مثبت‌تر باشد پیکسل آبی تر و هرچه منفی‌تر باشد پیکسل قرمزتر است.

تصویر دوم (هواپیما):

شکل ۲۶: تصویر هواپیما

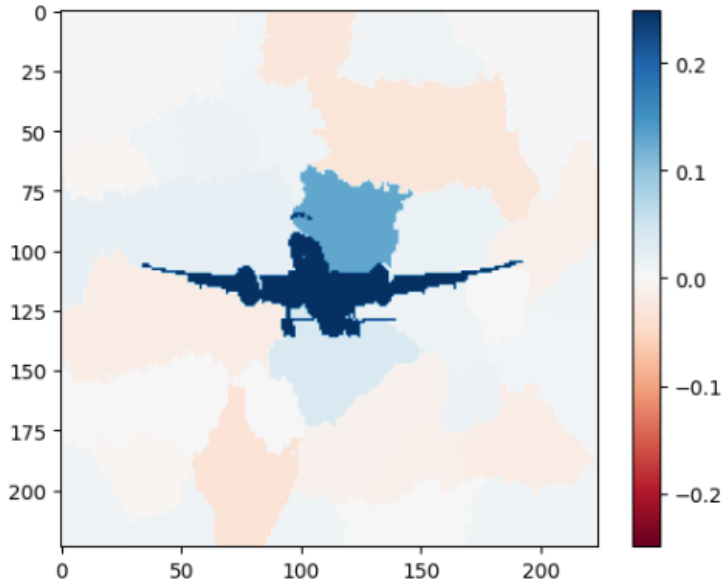
پنج کلاس پیشبینی شده و احتمال آن‌ها:

```
airliner: 0.8880
wing: 0.0361
warplane: 0.0276
kite: 0.0024
albatross: 0.0012
```



شکل ۲۷: نواحی **Boundaries** در شکل چپ و **Pros and Cons** در شکل سمت راست برای تصویر هواپیما

مشاهده می‌کنیم که پیکسل‌های هواپیما کاملا موثر در پیشبینی کلاس هواپیما تشخیص داده شده است و مدل از پیکسل‌های بال‌ها، دم، موتورها و چرخ‌ها برای این پیشبینی استفاده کرده است. Heatmap را نیز در شکل بعدی مشاهده می‌کنید. همچنین تعداد ویژگی برای تفسیر یکی است.



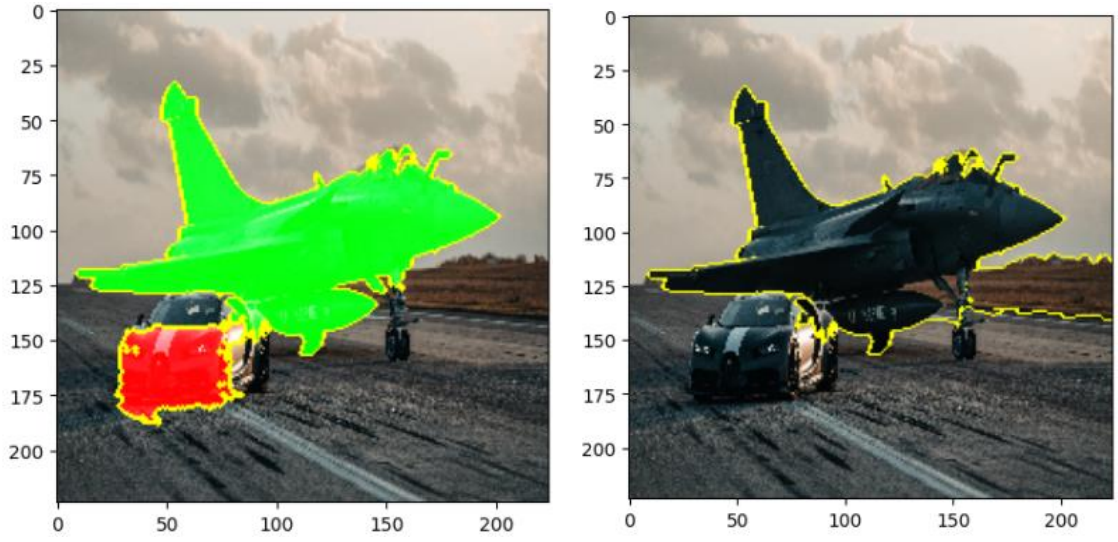
شکل ۲۸: **Heatmap** برای تصویر هواپیما

تصویر سوم (هواپیمای جنگنده و خودرو):

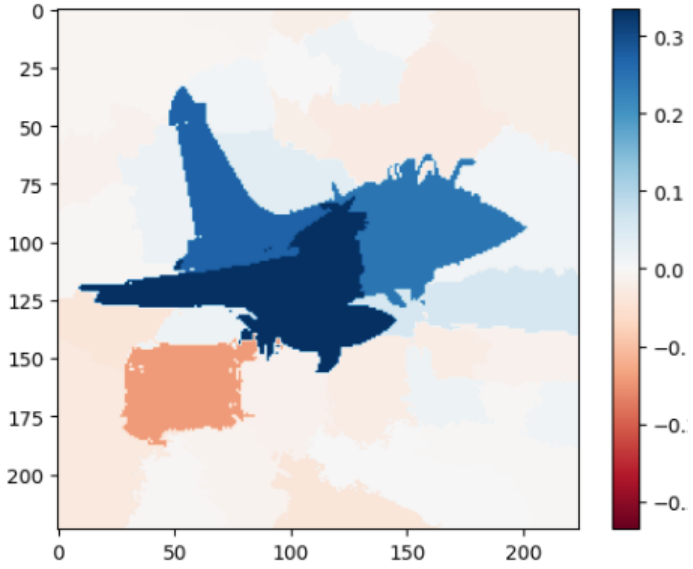
شکل ۲۹: تصویر جنگنده و خودرو

پنج کلاس پیشبینی شده و احتمال آن‌ها:

```
warplane: 0.9148
aircraft carrier: 0.0397
projectile: 0.0079
missile: 0.0068
wing: 0.0039
```



شکل ۳۰: نواحی **Boundaries** در شکل چپ و **Pros and Cons** در شکل سمت راست برای تصویر جنگنده و خودرو



شکل ۳۱: **Heatmap** برای تصویر جنگنده و خودرو

در این بخش نیز در شکل Boundary مشاهده می‌شود از تمامی پیکسل‌های هواپیمای جنگنده مخصوصا بال راست، مخزن سوخت آن، همچنین بخش سرنشین و بال عقب برای تشخیص خروجی استفاده شده است. قاعدتا همانطور که در شکل Pros and cons و Heatmap نشان داده شده است، پیکسل‌ها خودرو در پیشبینی مدل که جنگنده می‌باشد تاثیر منفی دارند. همچنین تعداد ویژگی برای تفسیر چهارتا است.