

دانشگاه صنعتي امیرکبیر  
(پلی تکنیک تهران)

دانشكده مهندسی کامپیوتر

پایان‌نامه کارشناسی

گرایش نرم افزار

تحلیل احساس روی متن فارسی

نگارش

محمدرضا شمشیرگرها

استاد راهنما

دکتر مصطفی حقیر چهرقانی

مهر 1399



اينجانب محمدرضا شمشیرگرها متعهد مي‌شوم كه مطالب مندرج در اين پايان نامه حاصل كار پژوهشي اينجانب تحت نظارت و راهنمايي اساتيد دانشگاه صنعتي اميركبير بوده و به دستاوردهاي ديگران كه در اين پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذكر گرديده است. اين پایان نامه قبلاً براي احراز هيچ مدرك هم‌سطح يا بالاتر ارائه نگرديده است.

در صورت اثبات تخلف در هر زمان، مدرك تحصيلي صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پيگيري قانوني خواهد داشت.

كليه نتايج و حقوق حاصل از اين پایان نامه متعلق به دانشگاه صنعتي اميركبير مي‌باشد. هرگونه استفاده از نتايج علمي و عملي، واگذاري اطلاعات به ديگران يا چاپ و تكثير، نسخه‌برداري، ترجمه و اقتباس از اين پایان نامه بدون موافقت كتبي دانشگاه صنعتي اميركبير ممنوع است.   
نقل مطالب با ذكر مآخذ بلامانع است.

محمدرضا شمشیرگرها

امضا

**سپاس گزاری**

از استاد بزرگوار، جناب آقای دکتر چهرقانی که راهنمای من بوده­اند، کمال سپاس و قدردانی را دارم و تلاش ها و راهنمایی های ایشان را در به ثمر رساندن این پروژه ارج می­نهم.

محمدرضا شمشیرگرها

مهر 1399

# چكيده

گسترش شبکه جهانی وب روشی جدید در بیان احساسات افراد به همراه آورده است. با افزایش روزافزون وبسایت­های فروش محصول به صورت آنلاین و کاربران آن­ها، تحلیل نظرات به صورت سنتی امری دشوار می­باشد. درهمین راستا، نیاز به یک ماشینی است که به صورت خودکار نظرات کاربران درباره هر محصول را تحلیل کند و نتیجه تحلیل را که احساس نویسنده است، در غالب یک جدول کوتاه نمایش دهد. هدف این پروژه تحلیل احساسات روی متن فارسی است، اینکار به کمک دسته­بند­ بیز ساده چند­جمله­ای، دسته­بند بردار پشتیبان خطی و شبکه عصبی پیچشی به صورت نظارتی انجام شده­است. در این پروژه میزان دقت هر­ الگوریتم بر اساس تعداد داده­های آموزشی بررسی شده است. نتیجه این تحلیل احساس می­تواند در مراحل بعدی به عنوان ورودی یک سیستم پیشنهاد­دهنده تبلیغاتی استفاده شود. برنامه­ای سمت کاربر و تحت وب به کمک Vue.js طراحی و ساخته شده است که قابلیت گرفتن یک متن فارسی، ارسال اطلاعات به برنامه سرور که به کمک Django پیاده­سازی شده است و دریافت و نمایش گروه احساسی متن را دارد.

واژه‌های کلیدی:

تحلیل احساسات، یادگیری ماشین، یادگیری عمیق، پردازش زبان طبیعی

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[چكيده ‌أ](#_Toc53577915)

[فصل اول مقدمه 1](#_Toc53577916)

[1-1- پیش گفتار 2](#_Toc53577917)

[2-1- تعریف مساله 3](#_Toc53577918)

[3-1- کارهای مرتبط 4](#_Toc53577920)

[4-1- خلاصه فصل­های بعد 5](#_Toc53577927)

[فصل دوم مقدمات، تعاریف، الگوریتم­ها 6](#_Toc53577928)

[1-2- یادگیری ماشین 7](#_Toc53577929)

[2-1-1- یادگیری با نظارت 7](#_Toc53577930)

[2-1-2- یادگیری بدون نظارت 8](#_Toc53577931)

[2-2- آماده­سازی داده­ها 9](#_Toc53577932)

[2-2-1- پیش پردازش 10](#_Toc53577933)

[2-2-2- تعبیه کلمه عصبی 11](#_Toc53577934)

[2-2-2-1- تعبیه کلمات عصبی آنلاین 12](#_Toc53577935)

[2-2-2-2- تعبیه کلمات از پیش آموزش داده شده 13](#_Toc53577936)

[3-2- الگوریتم بیز ساده 13](#_Toc53577937)

[2-3-1- دسته بند ساده گاوسی 14](#_Toc53577938)

[2-3-2- دسته بند بیز ساده چندجمله­ای 14](#_Toc53577939)

[2-3-3- دسته بند بیز ساده برنولی 15](#_Toc53577940)

[4-2- الگوریتم ماشین بردار پشتیبان 16](#_Toc53577941)

[5-2- یادگیری عمیق 26](#_Toc53577942)

[6-2- شبکه عصبی پیچشی 28](#_Toc53577943)

[2-6-1- معماری شبکه عصبی 31](#_Toc53577944)

[فصل سوم اجرا الگوریتم ها و ساخت مدل 35](#_Toc53577945)

[1-3- نتیجه­گیری 36](#_Toc53577946)

[فصل چهارم پیاده سازی سامانه 38](#_Toc53577947)

[1-4- انتخاب ابزار و زبان 39](#_Toc53577948)

[2-4- نمودار مورد کاربردی 39](#_Toc53577949)

[3-4- فرایند توسعه نرم افزار 40](#_Toc53577950)

[4-4- اجزا سامانه 41](#_Toc53577951)

[4-4-1- برنامه سمت کاربر 41](#_Toc53577952)

[4-4-2- برنامه سمت سرور 45](#_Toc53577953)

[فصل پنجم جمع‌بندي و پیشنهادات 46](#_Toc53577954)

[1-5- جمع­بندی 47](#_Toc53577958)

[2-5- پیشنهاد 48](#_Toc53577959)

[منابع و مراجع 49](#_Toc53577960)

[پيوست‌ها 51](#_Toc53577961)

[Abstract 52](#_Toc53577962)

|  |  |
| --- | --- |
| فهرست اشكال | صفحه |

[شكل ‏1‌-‌‌2- مراحل پیش پردازش . 11](#_Toc276969410)

[شكل ‏2‌-‌‌2- لایه تعبیه آنلاین . 12](#_Toc276969410)

[شكل ‏3‌-‌‌2- ترسیم داده‌ها در فضای n بعدی در الگوریتم ماشین بردار پشتیبان . 16](#_Toc276969410)

[شكل ‏4-‌‌2- تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۱) . 17](#_Toc276969410)

[شكل ‏5‌-‌‌2- تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 18](#_Toc276969410)

[شكل ‏6‌-‌‌2- تشخیص خط راست صحیح با محاسبه حاشیه در الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 19](#_Toc276969410)

[شكل ‏7-‌‌2- تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان (سناریو 3) . 20](#_Toc276969410)

[شكل ‏8‌-‌‌2- تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان هنگام وجود دورافتادگی (سناریو ۴) . 21](#_Toc276969410)

[شكل ‏9‌-‌‌2- تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان هنگام وجود دورافتادگی (سناریو ۴) . 22](#_Toc276969410)

[شكل ‏10‌-‌‌2- حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵). 23](#_Toc276969410)

[شكل ‏11‌-‌‌2- حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵) . 24](#_Toc276969410)

[شكل ‏12‌-‌‌2- حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵) . 25](#_Toc276969410)

[شكل ‏13‌-‌‌2- آموزش یک پرسپترون . 29](#_Toc276969410)

[شكل ‏14-‌‌2- آموزش یک پرسپترون. 30](#_Toc276969410)

[شكل ‏15‌-‌‌2- آموزش یک پرسپترون. 31](#_Toc276969410)

[شكل ‏16‌-‌‌2- معماری شبکه عصبی پیچشی . 32](#_Toc276969410)

[شكل ‏1‌-‌‌4- تمودار کاربرد . 40](#_Toc276969410)

[شكل ‏2‌-‌‌4- Unified Process . 41](#_Toc276969410)

[شكل ‏3‌-‌‌4- برنامه سمت کاربر. 42](#_Toc276969410)

[شكل ‏4‌-4- برنامه سمت کاربر – پاسخ الگوریتم بیز ساده . 43](#_Toc276969410)

[شكل ‏5‌-‌‌4- برنامه سمت کاربر – پاسخ الگوریتم ماشین بردار پشتیبان . 44](#_Toc276969410)

[شكل ‏6-‌‌4- برنامه سمت کاربر – پاسخ الگوریتم شبکه عصبی پیچشی . 45](#_Toc276969410)

[شكل ‏5‌-‌‌2- تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 18](#_Toc276969410)

[شكل ‏6‌-‌‌2- تشخیص خط راست صحیح با محاسبه حاشیه در الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 19](#_Toc276969410)

[شكل ‏7-‌‌2- تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان (سناریو 3) . 20](#_Toc276969410)

[شكل ‏1‌-‌‌2- مراحل پیش پردازش . 11](#_Toc276969410)

[شكل ‏2‌-‌‌2- لایه تعبیه آنلاین . 12](#_Toc276969410)

[شكل ‏3‌-‌‌2- ترسیم داده‌ها در فضای n بعدی در الگوریتم ماشین بردار پشتیبان . 16](#_Toc276969410)

[شكل ‏4-‌‌2- تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۱) . 17](#_Toc276969410)

[شكل ‏5‌-‌‌2- تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 18](#_Toc276969410)

[شكل ‏6‌-‌‌2- تشخیص خط راست صحیح با محاسبه حاشیه در الگوریتم ماشین بردار پشتیبان (سناریو ۲) . 19](#_Toc276969410)

[شكل ‏7-‌‌2- تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان (سناریو 3) . 20](#_Toc276969410)

|  |  |
| --- | --- |
| فهرست جداول | صفحه |

[جدول 2-1- کلاس­های احساسی به همراه برچسب آن­ها. 9](#_Toc276969411)

[جدول ‏2‌-2- تعداد عبارت هر کلاس احساسی در مجموعه داده.. 10](#_Toc276969412)

[جدول ‏2‌-3- دقت دسته­بند ها بر اساس نسبت داده­های تست.. 36](#_Toc276969412)

# فصل اول مقدمه

## 1-1- پیش گفتار

از دیرگاه کلمه­ی «روبات» و «کامپیوتر» به سرد و خشک و بی­احساس بودن شناخته شده­اند، ولی می­توان روزی را تصور کرد که این شناخت به کلی متحول شود. تشخیص احساسات شاخه­ای از علوم کامپیوتر و پردازش زبان طبیعی است که سعی دارد ماشین و هوش مصنوعی را با احساس و عواطف انسانی آشنا و تشخیص آن­ها را میسر سازد]1[.

گسترش شبكه جهاني وب روشي جدید در بیان احساسات افراد به همراه آورده است. همچنین رسانه­ای با حجم عظیمي از اطلاعات است كه در آن كاربران مي­توانند نظرات سایر كاربران را كه در كلاس­های مختلف احساسات طبقه­­بندی شده­اند مشاهده كنند و به­طور فزاینده­ای به­عنوان یک عامل كلیدی در تصمیم­گیری در حال رشد هستند. با افزایش روزافزون وبسایت­های فروش محصول به صورت آنلاین و كاربران آن­ها، تحلیل نظرات كاربران به صورت سنتي امری دشوار می­باشد. در همین راستا، تحلیل احساسات[[1]](#footnote-1) می­تواند فرایند تشخیص و شناسایی الگوها[[2]](#footnote-2) را نسبت به روش­های سنتی تسریع بخشد.

تحلیل احساسات یا کاوش عقاید[[3]](#footnote-3)، دامنه تحقیقاتی است که به تجزیه و تحلیل عقاید، احساسات و نظرات افراد نسبت به موجودیت­هایی مانند محصولات، خدمات، سازمان­ها و مسائل می­پردازد]2[. در واقع داده متنی توسط الگوریتم ها سازمان­دهی می­شود و خروجی­های آن­ها بر­اساس نمره کسب شده توصیف می­شود. این خروجی ها همچنین به عنوان قطبیت نیز شناخته می­شوند که معمولا سه دسته مثبت، منفی و خنثی هستند؛ بنابراین این یک مساله چند کلاسه[[4]](#footnote-4) در یادگیری ماشین(ML)[[5]](#footnote-5) محسوب می­شود.

مدل­های تحلیل احساسات به دو دسته کلی با نظارت[[6]](#footnote-6) و بدون نظارت[[7]](#footnote-7) تقسیم می­شوند. مدل­های تحت نظارت از مجموعه داده­های دارای برچسب استفاده می­کنند. معمولا مجموعه­­ داده­های ذکر شده به زبان انگلیسی هستند و برای تحلیل احساسات در زبان­های دیگر به مجموعه داده­های مختص آن زبان نیاز داریم که تهیه آن نیازمند زمان زیاد و افرادی با همان زبان مادری برای تشخیص قطبیت­ها است. همچنین اینکه مدل پیشنهادی تا چه حد می­تواند درست پیش­بینی کند بحث جداگانه­ای است.

تشخیص احساس در زمان حال بیشترین کاربرد را در سایت­های عرضه محصولات کالا (حقیقی یا مجازی) دارد. برای مدیران سایت و شرکت­های عرضه کننده محصول بسیار مهم است که بدانند کدام نقد و بررسی ها از یک محصول مثبت هستند و کدام منفی و اینکه شدت منفی یا مثبت بودن چقدر است]3[.

شرکت­های بزرگی همچون آمازون و گوگل از تشخیص احساسات متن برای بررسی نظرات کاربران استفاده می­کنند. همچنین نتایج تحلیل احساسی می­تواند در مراحل بعدی به عنوان ورودی یک سیستم پیشنهاد­ دهنده استفاده شود و باور عمومی بر بی­احساس بودن ماشین­ها را کمرنگ­تر کند و در پی آن باعث آسان­تر شدن بسیاری از جهات زندگی انسان نیز خواهد شد.

نتیجه این تحلیل احساس مي­تواند در مراحل بعدی به عنوان ورودی یک سیستم پیشنهاد دهنده تبلیغاتي استفاده شود.

## 2-1- تعریف مساله

با توجه به مطالب بیان­شده، نیاز به سیستمی که تحلیل احساس برای زبان فارسی را انجام دهد احساس می­شود. اینگونه سیستم­ها برای زبان­های غیر انگلیسی به ندرت یافت می­شوند و برای همین ما تصمیم گرفتیم که سامانه­ای طراح کنیم که متن فارسی که می­تواند یک نظر برای یک کالا در یک وبسایت فروشگاهی باشد را دریافت کند و احساسات کاربر را استخراج کرده و در غالب یک جدول نمایش دهد.

در این سیستم ما به جای سه دسته مثبت، منفی و خنثی از پنج دسته­بندی استفاده کردیم تا شدت و میزان مثبت و منفی بودن مشخص شود. این دسته­بندی­ها عبارتند از: خشمگین، عصبانی، خنثی، خوشحال، بسیار راضی.

همچنین این سامانه قابلیت انتخاب الگوریتم استفاده شده برای تحلیل احساسات و همچنین قابلیت انتخاب اینکه چند درصد از مجموعه داده­ها را به عنوان داده تست در نظر بگیریم دارا است. برای تحلیل احساسی از دو الگوریتم یادگیری ماشین به نام بیزساده[[8]](#footnote-8) (NB) و ماشین بردار پشتیبان[[9]](#footnote-9) (SVM) و یک الگوریتم یادگیری عمیق[[10]](#footnote-10) به نام شبکه عصبی پیچشی (CNN) استفاده کردیم که در فصل­های بعد به­طور مفصل شرح داده خواهد شد.



## 3-1- کارهای مرتبط

مقاله ای]4[ که توسط کیم و همکاران در سال 2019 منتشر شد یک شبکه عصبی پیچشی را پیشنهاد داد که از لایه­های مختلفی برای دسته­بندی نظرات بر روی سه مجموعه داده تشکیل شده است. در مرحله اول، آن­ها یک ماتریس تعبیه[[11]](#footnote-11) را ایجاد کردند، جایی که کلمات به صورت معنایی در فضا قرار می­گیرند و همچنین از طریق فرایند آموزش تکمیل می­شوند. در مرحله دوم، این مقاله از دو لایه کانولوشن[[12]](#footnote-12) استفاده کرده است که لایه اول اطلاعات محلی را ذخیره می­کند و لایه دوم مطابق لایه اول از کلمات متنی ویژگی می­گیرد. پس از آن، حداکثر تجمع[[13]](#footnote-13) با یک گام[[14]](#footnote-14) اختیاری برای به­دست آوردن قابل توجه­ترین ویژگی­ها استفاده شده­است. سرانجام، مقدار احتمال برای هر کلاس با استفاده از یک لایه کاملاً متصل[[15]](#footnote-15) (FC) و تابع فعال­سازی softmax محاسبه شده­است. مدل CNN پیشنهادی آن­ها با روش­های سنتی یادگیری ماشین مانند NB و SVM مقایسه شده­است و به طور خلاصه، این مقاله ادعا می­کند که مدل ارائه شده آن به ترتیب 81 و 68 درصد دقت برای دسته­بندی باینری و سه گانه را به­دست آورده است.

در میان چند مطالعه دیگر، دشتی پور و همکاران]5[ یک رمزگذار خودکار عمیق[[16]](#footnote-16) و CNN را برای یک مجموعه داده جدید فارسی تعیین کرده است. پس از طی مراحل معمول پیش پردازش، آن­ها کلمات را به کمک کتابخانه FastText ]6[ که برای نمایش متن است به بردار تبدیل کردند. پس از آن ، یک رمزگذار خودکار برای کاهش ابعاد و بازسازی ورودی استفاده شده است. معماری CNN پیشنهادی آن­ها در کل دارای 11 لایه متشکل از کانولوشن و به دنبال آن حداکثر تجمع، لایه های کاملاً متصل و تابع فعال سازی softmax است. در نهایت، پرسپترون چند لایه[[17]](#footnote-17)(MLP) با CNN و رمزگذار خودکار مقایسه شده است. این مقاله ادعا کرد که معماری CNN از عملکرد MLP و رمزگذار خودکار با دقت 82.6 درصد برای دسته­بندی باینری پیشی گرفته است.



## 4-1- خلاصه فصل­های بعد

در فصل اول مقدمه و کارهای مرتبط صورت گرفته بیان شد. در فصل دوم به بیان مقدمات و توضیح الگوریتم­ها می­پردازیم. در فصل سوم توضیحاتی در مورد اجرای الگوریتم­ها و نتایج به­دست آمده خواهیم داد. فصل چهارم به توضیح معماری سامانه، فرایند توسعه نرم افزار و نحوه پیاده سازی سامانه پرداخته می­شود. فصل پنجم جمع­بندی و ارائه پیشنهاداتی برای بهبود سامانه است.

# فصل دوم مقدمات، تعاریف، الگوریتم­ها

در این فصل به تعاریف مفاهیم مهم مانند یادگیری ماشین و یادگیری عمیق می­پردازیم، الگوریتم­های استفاده شده را به طور کامل توضیح خواهیم داد و نحوه آماده سازی داده­ها را شرح می­دهیم.

## 1-2- یادگیری ماشین

در سطح بسیار بالا، یادگیری ماشین فرآیند آموزش یک سیستم رایانه­ای است که چگونه در صورت دریافت داده بتواند پیش­بینی­های دقیق انجام دهد]7[. این پیش­بینی­ها می­تواند پاسخ به این سوال باشد که آیا میوه­ی داخل تصویر موز است یا سیب؛ یا افرادی را که در جاده از مقابل اتومبیل خودران عبور می­کنند شناسایی کند یا اینکه آیا کلمه فیلم در یک عبارت به کار رفته یا خیر، یا تشخیص هرزنامه[[18]](#footnote-18) بودن ایمیل و یا حتی تولید زیرنویس برای فیلم­های یوتوب. تفاوت کلیدی آن با نرم­افزار­هایی که پیش از این وجود داشتند این است که یک توسعه دهنده انسانی کدی را ننوشته است که به سیستم دستور دهد چگونه تفاوت بین موز و سیب را تشخیص دهد. در عوض، یک مدل یادگیری ماشین آموخته شده است که چگونه می­تواند با آموزش روی مقدار زیادی داده، میوه­ها را به طور قابل اعتماد[[19]](#footnote-19) تشخیص دهد، در این حالت احتمالا تعداد زیادی از تصاویر دارای برچسب موز یا سیب هستند.

یادگیری ماشین به­ صورت کلی به به دو دسته با نظارت و بدون نظارت تقسیم می­شود.

### 2-1-1- یادگیری با نظارت

الگوریتم­های یادگیری تحت نظارت، یک مدل ریاضی از مجموعه­ای از داده­ها می­سازند که هم شامل ورودی­ها و هم خروجی­های مورد نظر است. داده­ها که به عنوان داده­های آموزشی شناخته می­شوند شامل مجموعه­ای از نمونه­های آموزشی به همراه برچسب می­باشد. در مدل ریاضی، هر مثال آموزشی توسط یک آرایه یا بردار نشان داده می­شود، که گاهی اوقات بردار ویژگی[[20]](#footnote-20) نامیده می­شود و داده­های آموزش توسط یک ماتریس نشان داده می­شود. از طریق بهینه­سازی تکراری یک تابع هدف، الگوریتم­های یادگیری تحت نظارت تابعی را یاد می­گیرند که می­تواند برای پیش­بینی خروجی مرتبط با ورودی­های جدید استفاده شود. یک تابع بهینه به الگوریتم اجازه می­دهد تا خروجی ورودی­هایی را که بخشی از داده­های آموزش نبودند ، به درستی تعیین کند. گفته می شود که الگوریتمی که دقت خروجی­ها یا پیش­بینی های خود را با گذشت زمان بهبود می­بخشد ، انجام آن کار را آموخته است.

انواع الگوریتم­های یادگیری تحت نظارت شامل یادگیری فعال[[21]](#footnote-21) ، دسته­بندی[[22]](#footnote-22)(کلاس­بندی یا طبقه­بندی نیز گفته می­شود) و رگرسیون[[23]](#footnote-23) است. الگوریتم­های طبقه­بندی هنگامی استفاده می شوند که خروجی­ها یک مجموعه محدودی از مقادیر باشند. هنگامی که ممکن است خروجی­ها دارای مقدار عددی در محدوده خاصی باشند، از الگوریتم­های رگرسیون استفاده می­شوند. به عنوان مثال، برای یک الگوریتم طبقه­بندی که ایمیل­ها را فیلتر می­کند، ورودی یک ایمیل است و خروجی نام پوشه­ای است که در آن ایمیل را بایگانی می­کند.

با این حال، آموزش این سیستم­ها معمولاً به داده­های زیاد با برچسب نیاز دارد، برخی از سیستم­ها برای تسلط بر یک کار به میلیون­ها داده­ برچسب­دار نیاز دارند.

### 2-1-2- یادگیری بدون نظارت

الگوریتم­های یادگیری بدون نظارت مجموعه­ای از داده­ها را شامل می­شوند که فقط شامل ورودی­ها هستند و ساختار آن­ها را مانند گروه­بندی یا خوشه­بندی[[24]](#footnote-24) نقاط داده می­یابند. بنابراین، الگوریتم­ها از داده­های آزمایشی که برچسب­گذاری، دسته­بندی یا طبقه­بندی نشده­اند، می­آموزند. به جای پاسخ به بازخورد­، الگوریتم­های یادگیری بدون نظارت، مشترکات[[25]](#footnote-25) موجود در داده­ها را شناسایی می­کنند و بر­اساس وجود یا عدم وجود چنین اشتراکاتی در هر بخش جدید از داده­ها واکنش نشان می­دهند. یک کاربرد اصلی یادگیری بدون نظارت در زمینه برآورد تراکم در آمار، مانند یافتن تابع چگالی احتمال است؛ اگرچه یادگیری بدون نظارت دامنه­های دیگری را مانند خلاصه کردن و توضیح ویژگی­های داده شامل می­شود.

تجزیه و تحلیل خوشه[[26]](#footnote-26)، انتساب مجموعه­ای از مشاهدات به زیرمجموعه­ها (خوشه ها) نامیده می­شود، به طوری که مشاهدات درون یک خوشه با توجه به یک یا چند معیار از پیش تعیین شده مشابه هستند، در حالی که مشاهدات گرفته شده از خوشه­های مختلف شباهت ندارند. تکنیک­های مختلف خوشه­بندی فرض­های مختلفی را در مورد ساختار داده­ها ارائه می­دهند، که اغلب توسط برخی از معیارهای تشابه تعریف شده و ارزیابی می­شوند.

## 2-2- آماده­سازی داده­ها

قبل از اینکه به الگوریتم­های یادگیری ماشین بپردازیم نیاز داریم تا داده­ها را نرمال کرده و تبدیل به داده­هایی کنیم که برای الگوریتم قابل فهم باشند. این کار در دو مرحله­ی پیش پردازش[[27]](#footnote-27) و استخراج ویژگی[[28]](#footnote-28) انجام می­شود.

در این پروژه داده­های ما متن فارسی است که از نظرات محصولات در سایت دیجیکالا استخراج و برچسب گذاری شده­اند. برچسب­ها به همراه کلاس احساسی در جدول2-1 آورده شده است. همچنین تعداد عبارات هرکلاس در مجموعه داده­ها در جدول2-2 آورده شده­است.

جدول 2‌-‌1 کلاس­های احساسی به همراه برچسب آن­ها.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| کلاس احساسی | خشمگین | عصبانی | خنثی | خوشحال | بسیار راضی |
| برچسب | 2- | 1- | 0 | 1+ | 2+ |

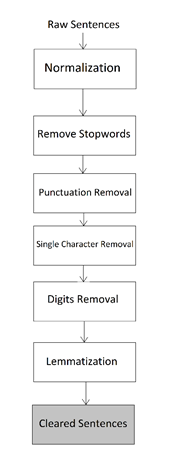
جدول 2‌-‌2 تعداد عبارت هر کلاس احساسی در مجموعه داده.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| برچسب | 2- | 1- | 0 | 1+ | 2+ |
| تعداد عبارت | 40 | 1697 | 3152 | 2185 | 1342 |

### 2-2-1- پیش پردازش

داده­هایی که برای آموزش و تست داریم از نظرات محصولات در سایت دیجیکالا گرفته شده­اند، این داده­ها که متن فارسی هستند حاوی اطلاعات بدون استفاده هستند که ممکن است عملکرد الگوریتم را دچار مشکل کنند، مثل علائم نگارشی و حروف اضافه و کلماتی که حاوی بار معنایی خاصی نیستند.

همانطور که در شکل 2-1 نشان داده شده است، 6 مرحله مختلف پیش پردازش استفاده شده­است. ابتدا هر عبارت توسط یک ابزار پیش پردازش کننده برای زبان فارسی، به نام Hazm نرمال سازی می­شود، در این مرحله فاصله­های خالی که اشتباه هستند درست خواهند شد. در چهار مرحله­ی بعدی کلمات توقف[[29]](#footnote-29)، علائم نگارشی، حروف تنها و اعداد حذف می­شوند. مرحله آخر، لم سازی[[30]](#footnote-30)، فرایندی است که اشکال مختلف کلمه را در یک گروه قرار می­دهد تا بتوان آن­ها را به عنوان یک کلمه واحد در نظر گرفت و تحلیل کرد. لم سازی شبیه ریشه­یابی است اما اصل و موضوع کلمه را به دست می­آورد، به­طوری که کلماتی با معنی مشابه را به یک کلمه پیوند می­دهد.



شکل 2-1: مراحل پیش­ پردازش

### 2-2-2- تعبیه کلمه عصبی

بعد از مرحله پیش پردازش، داده­ها به صورت متن نرمال شده و کلمات لم هستند؛ ولی هنوز هم داده­های ما برای ورودی الگوریتم مناسب نیستند، زیرا این الگوریتم­ها پایه ریاضی دارند. ما به جای بردار خلوت[[31]](#footnote-31) که در الگوریتم­های سنتی یادگیری ماشین استفاده می­شد از تعبیه کلمه عصبی[[32]](#footnote-32) در مدل­های یادگیری عمیق[[33]](#footnote-33) استفاده کردیم؛ در این روش بردار ویژگی از نگاشت هر کلمه به یک بردار با ابعاد پایین­تر[[34]](#footnote-34) به دست می­آید]8[.

به طور خلاصه، هر جمله رمزگذاری[[35]](#footnote-35) می­شود ، کلمات بعدی برداری می­شوند. در نتیجه، هرچه بردارهای مشابه بیشتری در میان جملات قرار بگیرند، شباهت بیشتری خواهند داشت. شایان ذکر است که تعبیه کلمه عصبی نه تنها یک ردیاب مترادف است بلکه یک روش برای یافتن کلمات از یک خانواده (به عنوان مثال گربه، سگ) است. این بردارها به دو صورت ممکن به شرح زیر فرا می­گیرند.

#### 1-2-2-2- تعبیه کلمات عصبی آنلاین

این روش به مجموعه داده موجود متکی است و در فرایند یادگیری عصبی عملی خواهد شد. در واقع، بردارهای خروجی با هیچگونه عمل ریاضی روی ورودی محاسبه نمی­شوند. بنابراین، هر کلمه در جملات همانطور که ظاهر می­شود با یک عدد صحیح رمزگذاری می­شود. در این حالت، اگر تعداد کلمات مجموعه واژگان را نشان دهد و بُعد تعبیه بردارها را نشان دهد، پس از آموزش شبکه عصبی، انتظار داریم یک بردار تعبیه شده در اندازه به شرح زیر باشد. در همین حال، شکل 2-2 نحوه کار لایه تعبیه­شده آنلاین برای دو جمله را نشان می­دهد.



شکل 2-2: لایه تعبیه آنلاین]9[

#### 2-2-2-2- تعبیه کلمات از پیش آموزش داده شده

بر خلاف تعبیه عصبی آنلاین، این روش از بردار­های تعبیه شده از قبل آموزش ­داده­شده استفاده می­کند. چندین بردار عصبی از پیش آموزش داده­شده توسط دانشگاه­ها و همچنین شرکت­ها ساخته شده­است، در میان آن­ها انتخاب مناسب برای زبان فارسی کتابخانه FastText است که توسط فیسبوک ارائه شده­است­]10[. داده­های استفاده شده در FastText از ویکی پدیای فارسی جمع آوری شده­اند و هر کلمه به یک بردار 300 بعدی نگاشت می­شود.

## 3-2- الگوریتم بیز ساده

الگوریتم بیز ساده یک روش دسته­بندی بر اساس قضیه بیز[[36]](#footnote-36) با فرض استقلال در بین پیش­بینی کننده­ها است. به زبان ساده­تر، یک دسته­بند بیز ساده فرض می­کند که وجود یک ویژگی خاص در یک کلاس با وجود ویژگی دیگر ارتباطی ندارد]11[.

به عنوان مثال، میوه­ای اگر قرمز، گرد و قطر حدود 7 سانتی­متر باشد ، ممکن است یک سیب در نظر گرفته شود. حتی اگر این ویژگی­ها به یکدیگر یا وجود ویژگی­های دیگر بستگی داشته باشند، همه این خصوصیات به طور مستقل در احتمال سیب بودن این میوه نقش دارند و به همین دلیل این الگوریتم به «ساده» معروف است.

ساخت مدل بیز ساده آسان است و به ویژه برای مجموعه داده­های بسیار بزرگ مفید است. علاوه بر سادگی، بیز ساده شناخته شده است که حتی از روش­های طبقه بندی بسیار پیچیده نیز پیشی می­گیرد.

قضیه بیز با توجه به کلاس متغیر و بردار وابسته ویژگی تا ، رابطه زیر را بیان می­کند:

با استفاده از فرض استقلال مشروط ساده داریم:

برای تمامی ها، این رابطه به صورت زیر ساده می­شود:

از آنجایی که با توجه به ورودی ثابت است، می­توانیم از قانون دسته­بندی زیر استفاده کنیم:

و می­توانیم از تخمین Maximum A Posteriori (MAP) برای تخمین و استفاده کرد؛ اولین مورد فراوانی نسبی کلاس (دسته) در مجموعه آموزش است.

### 2-3-1- دسته بند ساده گاوسی[[37]](#footnote-37)

اگر مشاهدات و داده­ها از نوع پیوسته باشند، از مدل احتمالی با توزیع گاوسی یا نرمال برای متغیرهای مربوط به شواهد می‌توانید استفاده کنید. در این حالت هر دسته یا گروه دارای توزیع گاوسی است. به این ترتیب اگر دسته یا کلاس داشته باشیم می‌توانیم برای هر دسته میانگین و واریانس را محاسبه کرده و پارامترهای توزیع نرمال را برای آن‌ها برآورد کنیم. فرض کنید که میانگین و واریانس دسته ام یعنی باشد. همچنین را مشاهدات حاصل از متغیرهای تصادفی در نظر بگیرید. از آنجایی که توزیع در هر دسته گاوسی(نرمال) فرض شده است، خواهیم داشت:

### 2-3-2- دسته بند بیز ساده چندجمله­ای[[38]](#footnote-38)

بیز ساده چندجمله‌ای، به عنوان یک دسته‌بند متنی بسیار به کار می‌آید. در این حالت برحسب مدل احتمالی یا توزیع چند جمله­ای، برداری از ویژگی برای یک مشاهده به صورت با احتمالات در نظر گرفته می‌شود. مشخص است که در این حالت بردار بیانگر تعداد مشاهداتی است که ویژگی خاصی را دارا هستند. به این ترتیب تابع درست­نمایی در چنین مدلی به شکل زیر نوشته می‌شود:

اگر مدل بیز ساده را بر اساس لگاریتم تابع درست­نمایی بنویسیم، به­صورت یک دسته­بند خطی درخواهد آمد.

واضح است که در این حالت و است.

### 2-3-3- دسته ­بند بیز ساده برنولی[[39]](#footnote-39)

در این قسمت به بررسی توزیع برنولی و دسته‌بندی بیز خواهیم پرداخت. به شکلی این نوع از دسته‌بند بیز بیشترین کاربرد را در دسته‌بندی متن‌های کوتاه داشته، به همین دلیل محبوبیت بیشتری نیز دارد. در این مدل در حالت چند متغیره، فرض بر این است که موجود یا ناموجود بودن یک ویژگی در نظر گرفته شود. برای مثال با توجه به یک لغت­نامه مربوط به اصطلاحات ورزشی، متن دلخواهی مورد تجزیه و تحلیل قرار می‌گیرد و بررسی می‌شود که آیا کلمات مربوط به لغتنامه ورزشی در متن وجود دارند یا خیر. به این ترتیب مدل تابع درستنمایی متن براساس کلاس های مختلف به شکل زیر نوشته می‌شود.

مشخص است که منظور از احتمال تولید مشاهده از کلاس است. توجه شود که با توجه به استقلال مشاهدات، تابع درست­نمایی به صورت حاصل­ضرب نوشته شده است.

## 4-2- الگوریتم ماشین بردار پشتیبان

ماشین بردار پشتیبان (SVM)، یک الگوریتم نظارت‌شده یادگیری ماشین است که هم برای مسائل دسته‌بندی و هم مسائل رگرسیون قابل استفاده است؛ با این حال از آن بیشتر در مسائل دسته‌بندی استفاده می‌شود. در الگوریتمSVM ، هر نمونه داده را به عنوان یک نقطه در فضای n بعدی روی نمودار پراکندگی داده‌ها ترسیم کرده ( nتعداد ویژگی‌هایی است که یک نمونه داده دارد) و مقدار هر ویژگی مربوط به داده‌ها، یکی از مؤلفه‌های مختصات‌ نقطه روی نمودار را مشخص می‌کند. سپس، با ترسیم یک خط راست، داده‌های مختلف و متمایز از یکدیگر را دسته‌بندی می‌کند (مطابق شکل ۱)]12[.

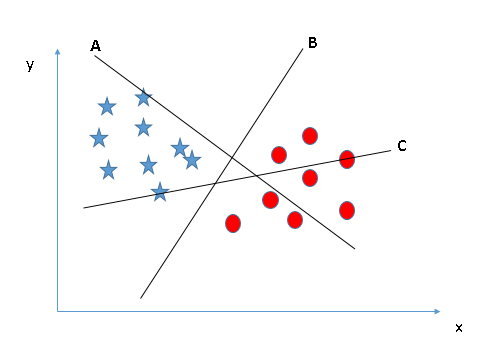


شکل 2-3: ترسیم داده‌ها در فضای n بعدی در الگوریتم ماشین بردار پشتیبان

به بیان ساده، بردارهای پشتیبان در واقع مختصات یک مشاهده منفرد هستند. ماشین بردار پشتیبان مرزی است که به بهترین شکل دسته‌های داده‌ها را از یکدیگر جدا می‌کند.

پیش از این، فرآیند جداسازی دو دسته با یک خط راست مورد بررسی قرار گرفت. اکنون، این پرسش مطرح می‌شود که «چگونه می‌توان این خط راست مناسب را تعیین کرد؟». پاسخ این سؤال در ادامه آورده شده است.

تشخیص خط راست صحیح (سناریو اول): در شکل ۲ سه خط راستA ، B و C وجود دارند. اکنون نیاز به تعیین خط راست صحیح برای دسته‌بندی ستاره‌های آبی و دایره‌های قرمز است.



شکل 2-4: تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۱)

یک قانون کلیدی وجود دارد که برای تعیین خط راست صحیح باید آن را همواره به یاد داشت: «خط راستی که دو دسته را به طور بهتری از یکدیگر جدا می‌کند، خطی است که باید انتخاب شود.». در این سناریو، خط «B» به شکل بسیار عالی هر دو دسته را از یکدیگر جدا می‌کند.

شناسایی خط راست صحیح (سناریو دوم): در شکل 2-5 هر سه خطA ، B و C دسته ستاره‌های آبی را به خوبی از دایره‌های قرمز جدا می‌کنند. پس چگونه می‌توان خط راست صحیح را انتخاب کرد؟



شکل 2-5: تشخیص خط راست صحیح با الگوریتم ماشین بردار پشتیبان (سناریو ۲)

با توجه به شکل 2-5، محاسبه فاصله نزدیک‌ترین نقطه داده (که از هر دسته‌ای می‌تواند باشد) از خط راست می‌تواند به انتخاب خط راست صحیح کمک کند. به این فاصله حاشیه گفته می‌شود.



شکل 2-6: تشخیص خط راست صحیح با محاسبه حاشیه در الگوریتم ماشین بردار پشتیبان (سناریو ۲)

در شکل 2-6 می‌توان مشاهده کرد که فاصله خط راست C در مقایسه با هر دو خط A و B از نزدیک‌ترین نقاط داده‌ای موجود در هر کلاس، بیشتر است. بنابراین، خط C را به عنوان خط راست صحیح برمی‌گزینیم. دلیل واضح دیگر برای انتخاب این خط استحکام بیشتر آن است. اگر خط راست حاشیه کمی داشته باشد، احتمال طبقه‌بندی نشدن برخی داده‌ها[[40]](#footnote-40) وجود دارد.

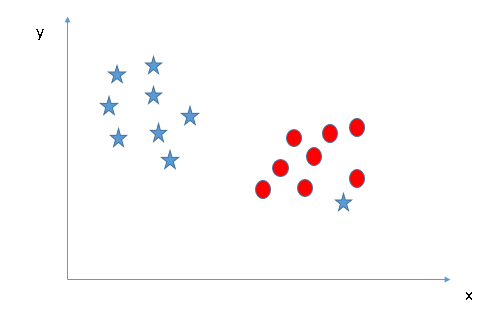
تشخیص خط راست صحیح (سناریو سوم): در شکل 2-7 برای تعیین دسته دایره‌های قرمز و ستاره‌های آبی از قوانین تشریح شده در بخش پیش قبلی برای تعیین خط راست صحیح استفاده می‌شود.



شکل 2-7: تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان (سناریو 3)

برخی ممکن است خط B را به دلیل حاشیه بیشتری که در مقایسه با A از نزدیک‌ترین نقطه داده موجود دارد انتخاب کنند. اما، نکته مهم آن است که در اصول تعیین خط راست در الگوریتم ماشین بردار پشتیبان، خط راستی که دسته‌ها را به درستی تقسیم کند (صحت) بر خطی که حاشیه بیشتری دارد، دارای اولویت است. بنابراین در این مثال، خط راست B یک خطای طبقه‌بندی دارد و خط A همه داده‌ها را به درستی طبقه‌بندی کرده است. بنابراین خط راست A‌ صحیح است.

آیا می‌توان داده‌های دارای دورافتادگی را دسته‌بندی کرد؟ (سناریو چهارم): در شکل 2-8، امکان طبقه‌بندی دو دسته با یک خط راست وجود ندارد، زیرا یکی از ستاره‌های آبی به صورت یک دورافتادگی در قلمرو دیگر دسته یعنی دایره‌های قرمز است.



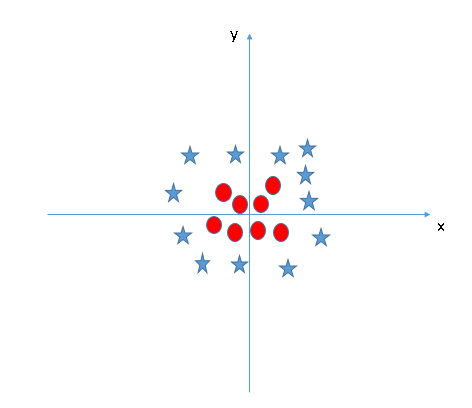
شکل 2-8: تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان هنگام وجود دورافتادگی (سناریو ۴)

همان‌گونه که پیش از این بیان شد، وجود یک ستاره آبی در قلمرو دسته دیگر به عنوان یک دورافتادگی برای دسته ستاره آبی محسوب می‌شود. یکی از ویژگی‌های ماشین بردار پشتیبان آن است که دورافتادگی‌ها را نادیده گرفته و تنها خط راستی را که بیشترین حاشیه را با نقاط داده دسته‌ها دارد انتخاب می‌کند.



شکل 2-9: تشخیص خط راست صحیح در الگوریتم ماشین بردار پشتیبان هنگام وجود دورافتادگی (سناریو ۴)

حل مسائل نیازمند خط غیر راست جهت جداسازی دسته‌ها (سناریو پنجم): در شکل 2-10، نمی‌توان یک خط راست بین دو کلاس داشت، بنابراین این سؤال مطرح می‌شود که ماشین بردار پشتیبان چگونه می‌تواند این دو دسته را طبقه‌بندی کند. تاکنون و در مثال‌های پیشین تنها خط راست مورد بررسی قرار گرفت.



شکل 2-10: حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵)

ماشین بردار پشتیبان می‌تواند این مسئله را به سادگی حل کند. این مسئله با افزودن یک ویژگی جدید قابل حل است. این ویژگی جدید تبدیل است که باید بر روی داده‌ها اعمال شود. اکنون می‌توان داده‌ها را روی محور x و z ترسیم کرد.



شکل 2-11: حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵)

در نمودار شکل 2-11، نقاط داده با شرایط زیر در نظر گرفته شده‌اند:

* همه مقادیر برای z همواره مثبت خواهند بود، زیرا z مجموعه مربعات x و y است.
* در نمودار اصلی، دایره‌های قرمز به محورهای x و y اصلی نزدیک‌ترند و این امر موجب می‌شود مقادیر z کاهش پیدا کند و دایره‌های قرمز در نمودار جدید به محور x نزدیک‌تر هستند و ستاره‌های آبی نسبت به دایره‌های قرمز فاصله بیشتری از محور x ها دارند.

در الگوریتم ماشین بردار پشتیبان، داشتن یک خط راست بین این دو کلاس آسان است. اما، سؤال دیگری که در این مرحله مطرح می‌شود آن است که آیا لازم است این ویژگی به صورت دستی به خط راست اضافه شود؟ پاسخ منفی است، ماشین بردار پشتیبان از روشی که به آن ترفند هسته[[41]](#footnote-41) (کرنل) گفته می‌شود، استفاده می‌کند. در این روش در واقع توابعی وجود دارند که فضای ورودی بُعد پایین را دریافت کرده و آن را به فضای بُعد بالاتر تبدیل می‌کنند. این تبدیل، یک مسئله غیر قابل جداسازی را به مسئله قابل جداسازی مبدل می‌کند. به این توابع، تابع‌های هسته (کرنل) گفته می‌شود.

توابع کرنل بیشتر در مسائل جداسازی غیرخطی مفید هستند. این توابع برخی از داده‌های فوق‌العاده پیچیده را تبدیل می‌کنند و سپس فرآیندی را می‌یابند که با استفاده از آن بتوانند این داده‌ها را بر اساس برچسب‌هایی که کاربر تعریف کرده، جداسازی ‌کنند. هنگامی که به خط جداساز در فضای ورودی اصلی نگاه می‌کنیم، این خط شبیه به یک دایره است (شکل 2-12).



شکل 2-12: حل مسائل نیازمند خط غیر راست در الگوریتم ماشین بردار پشتیبان (سناریو ۵)

ماشین بردار پشتیبان اساساً یک جداکننده دودویی است. در بخش قبلی پایه‌های تئوری ماشین‌های بردار پشتیبان برای دسته‌بندی دو کلاس تشریح شد. یک تشخیص الگوی چند کلاسی می‌تواند به وسیله­ی­ ترکیب ماشین‌های بردار پشیبان دو کلاسی حاصل شود. به‌طور معمول دو دید برای این هدف وجود دارد. یکی از آن‌ها استراتژی «یک در مقابل همه» برای دسته‌بندی هر جفت کلاس و کلاس‌های باقی‌مانده‌است. دیگر استراتژی «یک در مقابل یک» برای دسته‌بندی هر جفت است. در شرایطی که دسته‌بندی اول به دسته‌بندی مبهم منجر می‌شود. برای مسائل چند کلاسی، رهیافت کلی کاهش مساله چند کلاسی به چندین مسئله دودویی است. هریک از مسائل با یک جداکننده دودویی حل می‌شود. سپس خروجی جداکننده‌های دودویی SVM با هم ترکیب شده و به این ترتیب مسئله چند کلاس حل می‌شود.

## 5-2- یادگیری عمیق

یادگیری عمیق که زیر شاخه از یادگیری ماشین و بر مبنای مجموعه‌ای از الگوریتم‌ها است که در تلاش هستند مفاهیم انتزاعی سطح بالا در داده­ها را مدل نمایند که این فرایند را با استفاده از یک گراف عمیق که دارای چندین لایه پردازشی متشکل از چندین لایه تبدیلات خطی و غیر خطی هستند، مدل می‌کنند. به بیان دیگر پایه­ی آن بر یادگیری نمایش دانش[[42]](#footnote-42) و ویژگی‌ها در لایه‌های مدل است]13[.

یک نمونه آموزشی (برای نمونه: تصویر یک گربه) می‌تواند به صورت‌های گوناگون بسان یک بردار ریاضی پر شده از مقدار به ازای هر پیکسل و در دید کلی‌تر به شکل یک مجموعه از زیرشکل‌های کوچک‌تر (نظیر اعضای صورت گربه) مدل‌سازی شود. برخی از این روش‌های مدل‌سازی سبب ساده شدن فرایند یادگیری ماشین (برای نمونه: تشخیص تصویر گربه) می‌شوند. در یادگیری عمیق امید به جایگزینی استخراج این ویژگی‌های تصویر به دست بشر (مانند اعضای گربه) با روش‌های کامل خودکار تحت نظارت و نیمه نظارتی وجود دارد]14[.

انگیزه نخستین در به وجود آمدن این ساختار یادگیری از راه بررسی ساختار عصبی در مغز انسان الهام گرفته شده‌است که در آن یاخته‌های عصبی با فرستادن پیام به یکدیگر درک را امکان‌پذیر می‌کنند. بسته به فرض‌های گوناگون در مورد نحوه اتصال این یاخته‌های عصبی، مدل‌ها و ساختارهای مختلفی در این حوزه پیشنهاد و بررسی شده‌اند، هرچند که این مدل‌ها به صورت طبیعی در مغز انسان وجود ندارد و مغز انسان پیچیدگی‌های بیشتری را دارا است. این مدل‌ها نظیر شبکه عصبی عمیق، شبکه عصبی پیچیده، شبکه باور عمیق پیشرفت‌های خوبی را در حوزه‌های پردازش زبان‌های طبیعی، پردازش تصویر ایجاد کرده‌اند.

در حقیقت عبارت یادگیری عمیق، بررسی روش‌های تازه برای شبکه عصبی مصنوعی است.

امروزه از یادگیری عمیق در طرح‌های سرمایه‌گذاری برای افزایش میزان بازده استفاده می‌کنند.

یادگیری عمیق، رده‌ای از الگوریتم‌های یادگیری ماشین است که از چندین لایه برای استخراج ویژگی‌های سطح بالا از ورودی خام استفاده می‌کنند. به بیانی دیگر، رده‌ای از تکنیک‌های یادگیری ماشین که از چندین لایه‌ی پردازش اطلاعات و به‌ویژه اطلاعات غیرخطی بهره می‌برد تا عملیات تبدیل یا استخراج ویژگی نظارت‌شده یا نظارت‌نشده را عموماً با هدف تحلیل یا بازشناخت الگو، کلاس‌بندی، خوشه‌بندی انجام دهد.

برای مثال، در پردازش تصویر، لایه‌های پست‌تر می‌توانند لبه‌ها را تشخیص دهند، در حالی که لایه‌های عالی‌تر ممکن است ویژگی‌های پرمعناتر برای انسان، همچون حروف یا چهره‌ها، را تشخیص دهند.

یادگیری عمیق زیرشاخه‌ای از یادگیری ماشین است که از لایه‌های متعدد تبدیلات خطی به منظور پردازش سیگنال‌های حسی مانند صدا و تصویر استفاده می‌کند. ماشین در این روش هر مفهوم پیچیده را به مفاهیم ساده‌تری تقسیم می‌کند، و با ادامه این روند به مفاهیم پایه‌ای می‌رسد که قادر به تصمیم‌گیری برای آن‌ها است و بدین ترتیب نیازی به نظارت کامل انسان برای مشخص کردن اطلاعات لازم ماشین در هر لحظه نیست. موضوعی که در یادگیری عمیق اهمیت زیادی دارد، نحوه ارائه اطلاعات است. ارائه دادن اطلاعات به ماشین باید به نحوی باشد که ماشین در کمترین زمان اطلاعات کلیدی را که می‌تواند با استناد به آن‌ها تصمیم بگیرد را دریافت کند. هنگام طراحی الگوریتم‌های یادگیری عمیق می‌بایست به عوامل دگرگونی[[43]](#footnote-43) که اطلاعات مشاهده شده را توضیح می‌دهند توجه کنیم، این عوامل معمولاً عوامل قابل‌مشاهده‌ای نیستند بلکه عواملی هستند که بر روی دسته قابل مشاهده تأثیرگذار بوده یا زاده ساختارهای ذهنی انسان برای ساده‌تر کردن مسائل هستند. برای مثال در هنگام پردازش گفتار عوامل دگرگونی می‌توانند لهجه گوینده، سن یا جنسیت او باشند. در هنگام پردازش تصویر یک ماشین، میزان درخشش خورشید یک عامل دگرگونی است. یکی از مشکلات هوش مصنوعی تأثیر زیاد عوامل دگرگونی بر روی اطلاعات دریافتی است. برای مثال بسیاری از پیکسل‌های دریافتی از یک ماشین قرمز در شب ممکن است سیاه دیده بشوند. برای حل این مشکلات بعضاً به درک بالای اطلاعات (در حدود انسان) نیازمندیم و در واقع گاهی یافتن نحوه مناسب نمایش اطلاعات به اندازه خود مسئله سخت و زمان‌بر است.

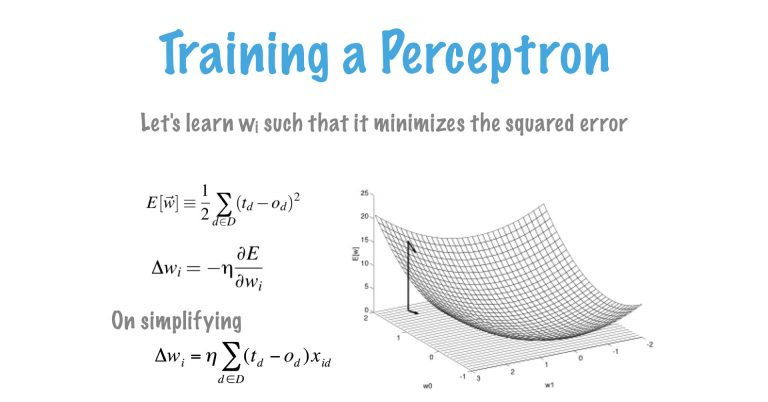
## 6-2- شبکه عصبی پیچشی

شبکه عصبی پیچشی (ConvNet) یک الگوریتم یادگیری عمیق است که تصویر ورودی را دریافت می‌کند و به هر یک از اشیا/جنبه‌های موجود در تصویر میزان اهمیت (وزن‌های قابل یادگیری و بایاس[[44]](#footnote-44)) تخصیص می‌دهد و قادر به متمایزسازی آن‌ها از یکدیگر است. در الگوریتم ConvNet در مقایسه با دیگر الگوریتم‌های دسته‌بندی به پیش‌پردازش کمتری نیاز است. در حالیکه فیلترهای روش‌های اولیه به صورت دستی مهندسی شده‌اند، شبکه عصبی پیچشی (ConvNets)، با آموزش دیدن به اندازه کافی، توانایی فراگیری این فیلترها/مشخصات را کسب می‌کند.

معماری ConvNet مشابه با الگوی اتصال نورون‌ها[[45]](#footnote-45) در مغز انسان است و از سازمان‌دهی قشر بصری[[46]](#footnote-46) در مغز الهام گرفته شده است. هر نورون به محرک‌ها تنها در منطقه محدودی از میدان بصری که تحت عنوان میدان تاثیر[[47]](#footnote-47) شناخته شده است پاسخ می‌دهد. یک مجموعه از این میدان‌ها برای پوشش دادن کل ناحیه بصری با یکدیگر هم‌پوشانی دارند]15[.

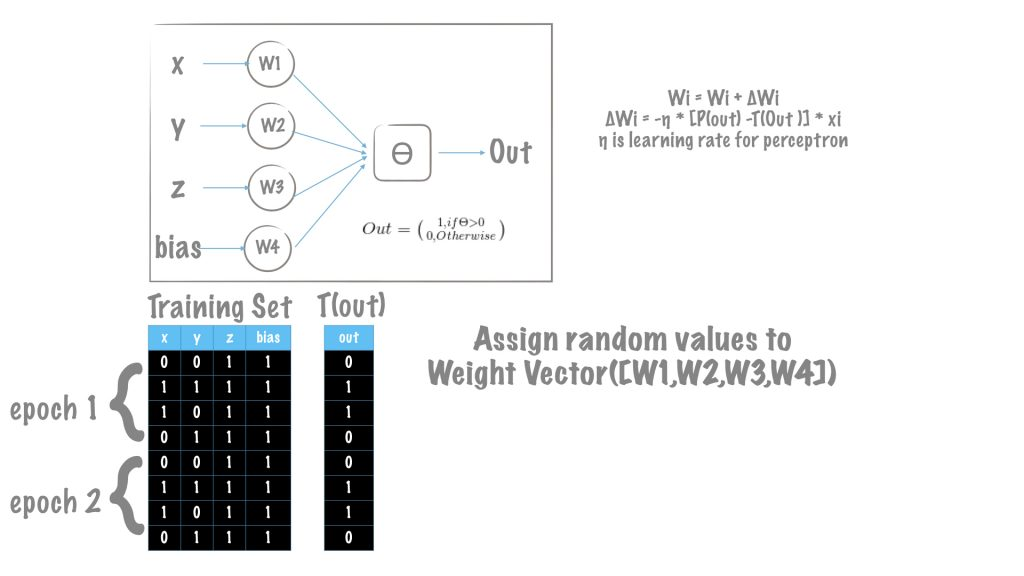
پرسپترون یکی از انواع پایه و ابتدایی شبکه‌های عصبی است که قابلیت یادگیری دارد. برای توسعه پرسپترون از رفتار بیولوژیکی نورون‌ها ایده گرفته می‌شود. یک پرسپترون، همانند یک نورون، یک سیگنال ورودی دریافت می‌کند. آن سیگنال را پردازش می‌کند و یک پاسخ را شبیه‌سازی می‌کند. یک پرسپترون می‌تواند برای حل مسئله جدایی‌پذیر تشخیص خطی آموزش داده شود.

الگوریتم یادگیری ماشین، الگوریتمی است که قادر به یادگیری از داده‌های موجود است. اما منظور ما از یادگیری چیست؟ میچل یادگیری را اینطور تعریف کرده است: «یک برنامه کامپیوتری در صورتی از تجربه E با توجه به کلاس‌های وظایف T و مقیاس عملکرد P یاد می‌گیرد که کارایی آن در وظیفه T برحسب مقیاس عملکرد P با استفاده از تجربه E بهبود یابد»]16[.



شکل 2-13: آموزش یک پرسپترون

در شکل 2-14، از توابع ریاضی ذکرشده در شکل 2-13 جهت حصول نتایج موردنظر ما استفاده‌شده است.

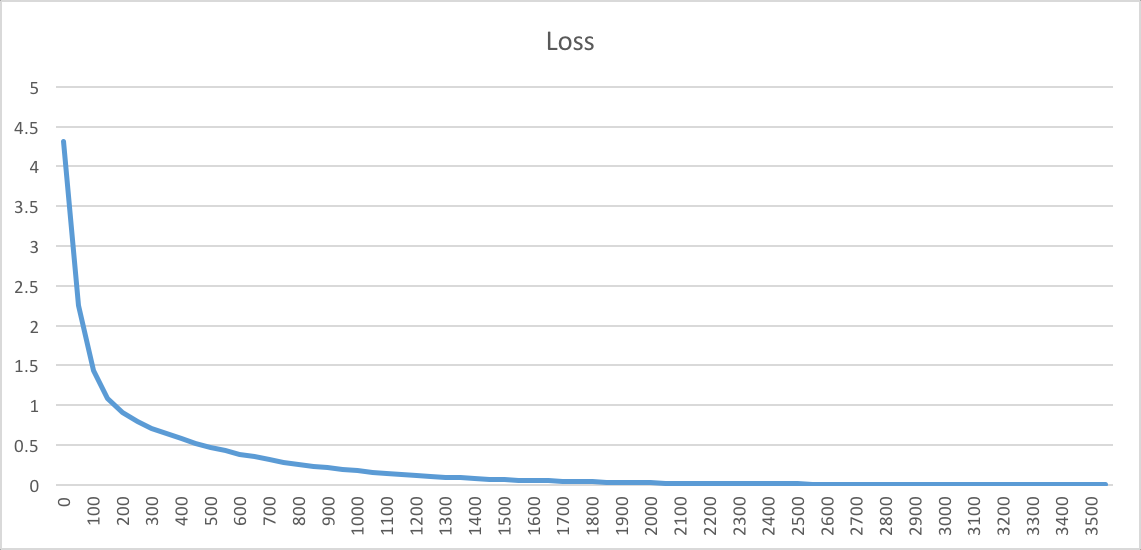


شکل 2-14: آموزش پرسپترون

یک تکرار[[48]](#footnote-48)، یک گذر کامل از کل مجموعه آموزشی در هنگام آموزش مدل یادگیری ماشین است. در یک تکرار، تمام نمونه‌های آموزشی یک‌بار به مدل شما ارائه می‌شوند. بنابراین، تعداد کل تکرارها در آموزش یک مدل معادل تعداد دوره‌ها در کل مجموعه داده آموزشی است.

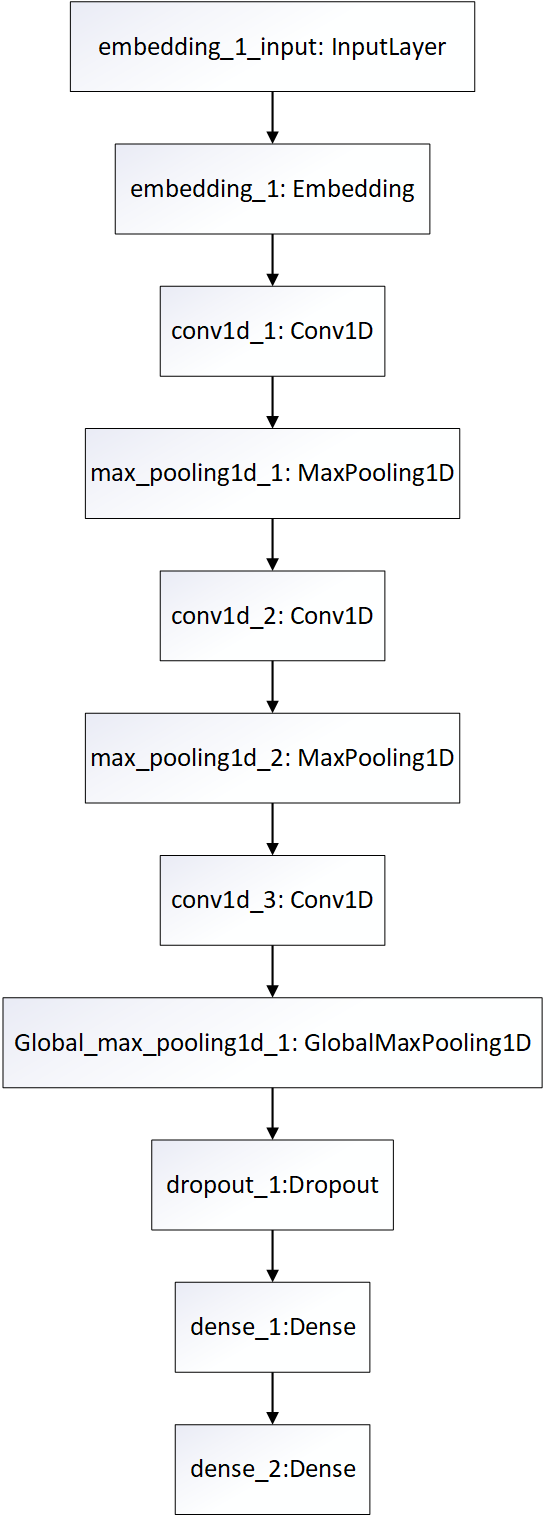
همانطور که در بخش الگوریتم آموزش توضیح داده شد، هدف ما افزایش عملکرد کارهای T است. در هر تکرار، ما پرسپترون را برای پیش‌بینی خروجی دلخواه در یک‌ جهت و به‌صورت پیشرو[[49]](#footnote-49) حرکت می‌دهیم. هنگامی‌که خروجی‌ها پیش‌بینی ‌شده (محاسبه ‌شده) با مقادیر مطلوب سازگار نباشد، وزن‌های پرسپترون را تغییر می‌دهیم. تغییر مقادیر وزن‌ها (w) به معنی افزایش یا کاهش وزن‌ها می‌باشد. این تغییر همان عملی است که در الگوریتم آموزش برای کمینه کردن خطای کلی تعریف‌شده است.

هنگام بهینه‌سازی وزن‌ها در آموزش هر مدل، همیشه خطا وجود دارد. نتیجه تابع خطای آموزش معمولی باید مشابه شکل 2-15 باشد.

 شکل 2-15: آموزش پرسپترون

### 2-6-1- معماری شبکه عصبی

معماری مورد استفاده ما برای شبکه عصبی در شکل 2-16 آورده شده است. این معماری معمولاً در محدوده پردازش تصویر استفاده می شود. با این حال یک مطالعه توسط کیم]17[ نشان داده است که می تواند برای طبقه بندی متن نیز کارآمد باشد. بنابراین ، مطابق شکل 2-16، مدل پیشنهادی شامل 9 لایه اضافی دیگر از جمله CNN به شرح زیر است:



شکل 2-16: معماری شبکه عصبی پیچشی

* InputLayer: با توجه به اینکه متون از کلمات تعبیه شده ساخته شده¬اند، در این لایه تعداد سلول های عصبی برابر است با حداکثر طول جملات رمز شده توسط کلمه. در مجموعه داده ما، طولانی ترین متن شامل 257 کلمه است، بنابراین این مدل 257 نورون در لایه اول خود دارد.
* Embedding: این لایه، لایه تعبیه است که در آن هر کلمه در بردار چند بعدی تعبیه شده است. مهم است که توجه داشته باشید اندازه این لایه در مدل های آموزش داده شده توسط تعبیه FastText قبلاً در ابعاد 300 ثابت شده است. با این حال ، مدل های پیشنهادی با استفاده از تعبیه کلمه Keras دارای خروجی 2000 نورون است.
* Conv 1D: که یک هسته کانولوشن ایجاد می کند که با ورودی لایه در یک بعد واحد متصل می شود و یک تنش[[50]](#footnote-50) خروجی تولید می کند. در این لایه ، اندازه هسته برابر با 4 و فیلترها 64 هستند.
* Max pooling 1D: که با شکل جهانی آن متفاوت است. با جزئیات بیشتر ، طول استخر می­گیرد ، اما حداکثر تجمع جهانی این کار را نمی کند. با توجه به اینکه گام یک است و اندازه استخر برابر با 2 ، در این صورت حداکثر مقدار هر اسلاید با طول 2 انتخاب می شود.
* Conv 1D: که شبیه لایه کانولوشن قبلی است. با این حال ، اندازه هسته آن 8 است ، به این معنی که 8 عنصر متوالی در نظر گرفته می شوند و در نهایت آنها یک مقدار واحد تولید می کنند.
* Max pooling 1D: که قبلا بدون تفاوت پارامترها توضیح داده شده است.
* Conv 1D: که همان دیگران است. اما ، اندازه هسته آن به 16 تغییر کرده است.
* Global max pooling 1D: که از سنسور دو بعدی اندازه ورودی و کانالهای ورودی استفاده می کند و حداکثر تمام مقادیر مربوط به هر یک از کانالهای ورودی را محاسبه می کند. توجه به این نکته مهم است که گام برابر است با یک.
* Dropout: که در زمان آموزش 10 درصد گره ها را رها می کند. شایان ذکر است که هرچه میزان ترک تحصیل بیشتر باشد ، مدل تعمیم یافته تری نیز ساخته می شود. بنابراین ، درصد براساس یک معامله انتخاب شده است.
* Dense: که از Sigmoid به عنوان یک تابع فعال سازی استفاده می کند. اگر خروجی مدل قبل از استفاده از تابع فعال سازی باشد، آنگاه به صورت زیر به دست می­آید:
* Dense: که یا یک طبقه بندی باینری یا یک طبقه بندی از احساسات 2- تا 2+ را برمی گرداند.

# فصل سوم اجرا الگوریتم ها و ساخت مدل

در سامانه طراحی شده از سه الگوریتم مختلف استفاده کردیم و پنج نسبت مختلف داده آموزش و تست را امتحان کردیم. به ازای هر الگوریتم و هر نسبت داده های آموزش و تست، 10 مدل مختلف ساخته شده است، که در هر کدام از این مدل ها داده ها به صورت تصادفی بین مجموعه های آموزش و تست تقسیم شده­اند (عدد بیان شده نسبت داده های تست به کل داده ها بر حسب درصد را بیان می­کند). از بین هر 10 مدل( تعداد کل مدل ها 150 است)، مدلی که بیشترین دقت را داشته انتخاب شده و در مرحله بعدی که در فصل بعد توضیح داده خواهد شد استفاده شده است. همچنین در جدول 3-1 میزان دقت هر مدل با توجه نسبت داده های تست به کل داده آورده شده است.

لازم به ذکر است که با توجه مساله ما که چند کلاسی است از بین انواع دسته­بند هایی که بر اساس الگوریتم بیز ساده هستند دسته­بند بیز ساده چند­جمله­ای استفاده شده است. در مدل های مبتنی بر ماشین بردار پشتیبان از دسته­بند بردار پشتیبان خطی استفاده شده که همانطور که در فصل قبل توضیح داده شد برای مسائل چند کلاسی از یک در مقابل بقیه بهره می­جوید.

جدول 3‌-‌1 دقت دسته­بند ها بر اساس نسبت داده­های تست.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| الگوریتم به کار رفته | نسبت داده های تست به کل داده ها بر حسب درصد | | | | |
| 10 | 20 | 25 | 30 | 40 |
| NB | 54.17 | 54.78 | 54.09 | 53.39 | 53.69 |
| SVM | 68.19 | 66.03 | 65.58 | 65.21 | 64.91 |
| CNN | 61.86 | 61.25 | 60.36 | 59.06 | 59.79 |

## 1-3- نتیجه گیری

با توجه به جدول 3-1 کاملا مشهود است که هرچه تعداد داده­ها بیشتر باشد مدل آموزش داده شده از دقت بالاتری برخوردار است. بالاترین میزان دقت را در مدل مبتنی بر ماشین بردار پشتیبان و درحالتی که بیشترین تعداد داده آموزشی را داریم اتفاق می­افتد.

معمولا روش­های یادگیری عمیق عملکرد بهتری از روش­های یادگیری ماشین دارند، در اینجا نیز شبکه عصبی پیچشی از بیز ساده بهتر عمل کرده است، اما نسبت به ماشین بردار پشتیبان عملکرد قابل قبولی نداشته است. علت تعداد داده­های آموزشی است، روش های مبتنی بر یادگیری عمیق نسبت به یادگیری ماشین به تعداد بسیار بیشتری داده آموزشی نیاز دارند. در این پروژه تعداد کل داده های ما 7416 است که برای یادگیری ماشین مناسب است ولی مدل های یادگیری عمیق با این تعداد عملکرد خوبی ندارند.

# فصل چهارم پیاده سازی سامانه

در این فصل راجع به بخش های مختلف سامانه و معماری آن توضیح خواهیم داد.

## 1-4- انتخاب ابزار و زبان

اولین چالشی که با آن مواجه بودیم، انتخاب زبان و ابزار بود. مواردی که برای ما مهم بود سرعت اجرا و راحتی پیاده سازی بود. با توجه به کتابخانه های ارائه شده برای زبان پایتون[[51]](#footnote-51) و سرعت اجرای بالای آن، این زبان بهترین انتخاب به نظر می­رسد. مدل های ساخته شده در این قسمت، در بخش سرور که بعدا توضیح داده خواهد شد استفاده می­شود.

## 2-4- نمودار مورد کاربردی

در سیستم طراحی شده کاربر می­تواند دو کار انجام دهد:

* ارسال نظر
* مشاهده گروه احساسی نظر

سیستم نیز دو کار انجام می­دهد:

* اجرای تحلیل احساسی روی نظرات کاربران
* محاسبه گروه احساسی نظرات ارسال شده

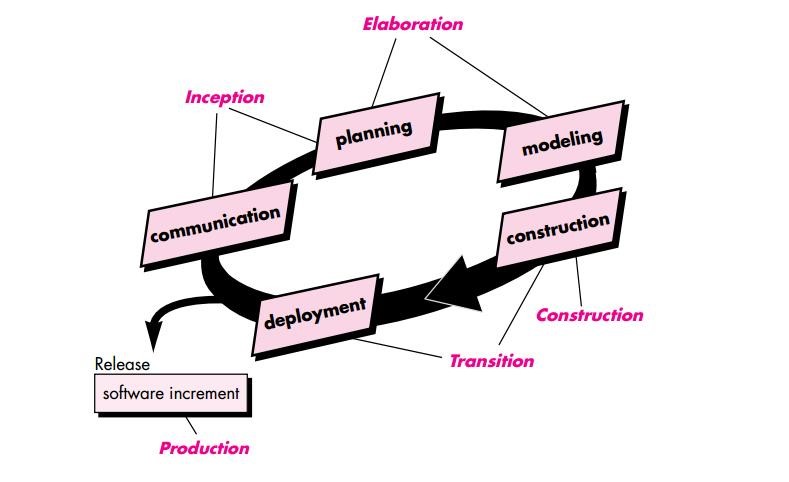
در شکل 4-1 که نمودار کاربرد آورده شده تمام این موارد قابل مشاهده است.



شکل 4-1: نمودار کاربرد

## 3-4- فرایند توسعه نرم افزار

با توجه به این كه ماهیت پروژه به گونه­ای است كه ابتدا نیاز است تا نمونه اولیه­ای ساخته شود و طي فرآیندی چرخشي الگوریتم بهبود یابد، از متودولوژی Unified Process (شکل 4-2) برای فرآیند توسعه نرم افزار استفاده خواهیم كرد.



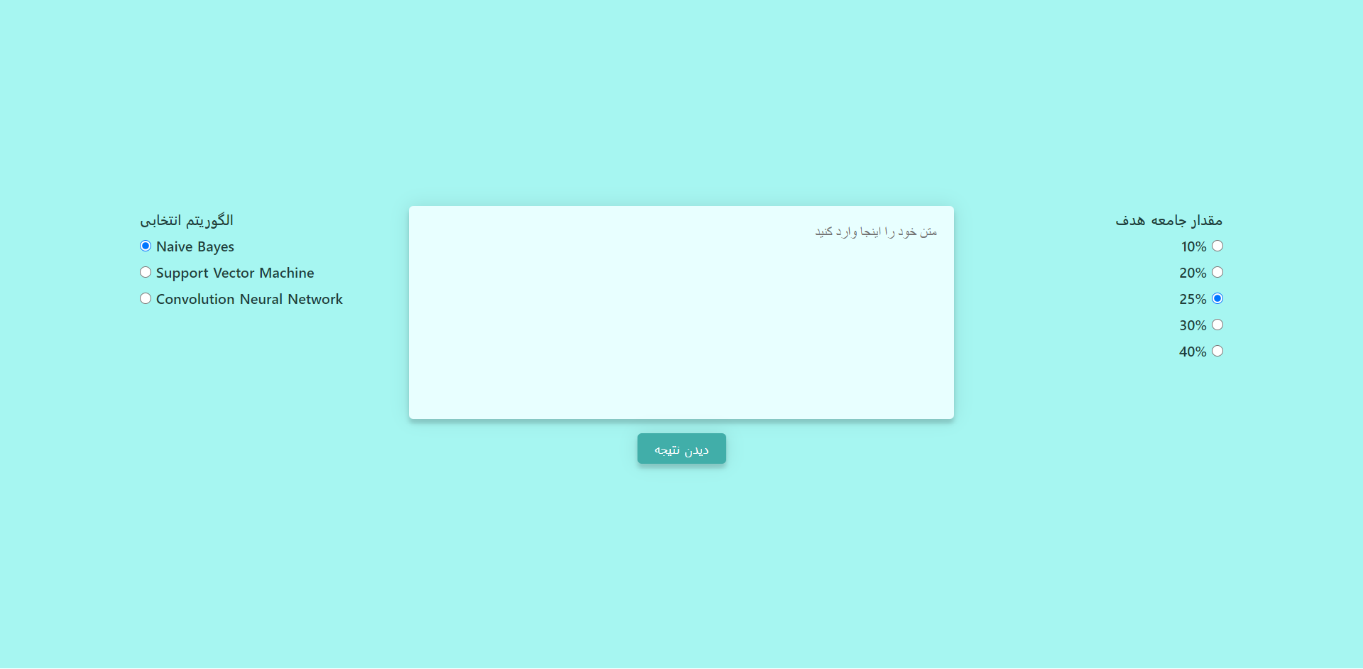
شکل 4-2: Unified Process ]18[

## 4-4- اجزا سامانه

در این بخش به توضیح اجزا سامانه که برنامه سمت سرور و برنامه سمت کاربر است می­پردازیم.

### 4-4-1- برنامه سمت کاربر

برنامه سمت کاربر با استفاده از چارچوب[[52]](#footnote-52) Vue.js که بر پایه جاوا اسکریپت برای ساخت طراحی کاربری می­باشد پیاده سازی شده­است. ارسال درخواست­ها به برنامه سمت سرور از طریق کتابخانه axios انجام می­شود. به کمک axios می­توان تمام درخواست های REST API را با توجه به تنظیمات هر کدام با داده­های مختلف ارسال کنیم. در شکل 5 تصویر این برنامه را مشاهده می­کنید.



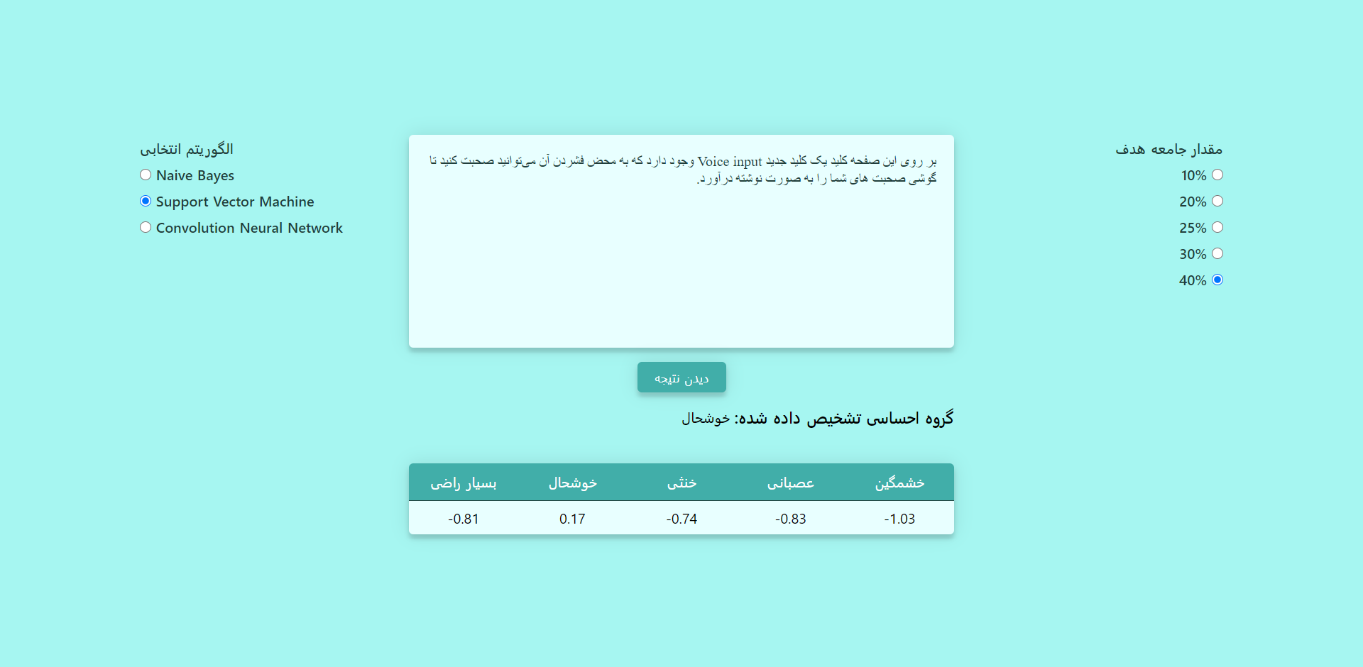
شکل 4-3: برنامه سمت کاربر

همانطور که در شکل 4-3 می­بینید در سمت چپ یک منو برای انتخاب الگوریتم وجود دارد که مشخص می­کند تحلیل احساسی بر اساس کدام الگوریتم باشد، و در سمت راست نیز می­توان نسبت داده های تست به کل داده ها را مشخص کرد. در وسط تصویر نیز همانطور که مشخص است محل قرارگیری متن، تعبیه شده است. برای مثال بعد از وارد کردن متن، اگر از منو سمت چپ Naïve Bayes و از منو سمت راست 25% انتخاب شود؛ سپس دکمه دیدن نتایج زده شود، به کمک axios درخواستی به سرور ارسال می­شود که حاوی متن وارد شده، درصد و الگوریتم است و جوابی که دریافت می­شود در شکل 4-4 آمده است.

 شکل 4-4: برنامه سمت کاربر-پاسخ الگوریتم بیز ساده

همانطور که در شکل 4-4 مشاهده می­کنید سامانه یک جدول به عنوان خروجی نشان می­دهد، زمانی که الگوریتم بیز ساده انتخاب شود اعداد مربوط به هر کلاس احساسی بیانگر میزان تعلق ورودی ( همان متنی که وارد کردیم) به آن کلاس است، این اعداد بر حسب درصد بوده و مجموع 5 کلاس برابر با 100 خواهد شد. واضح است که کلاس تشخیص داده شده توسط الگوریتم، کلاس با بیشترین مقدار است، در این مثال کلاس خوشحال با 45.84 درصد بیشترین مقدار بین کلاس­های احساسی را دارد و اتفاقا در این مورد الگوریتم بز ساده کلاس احساسی را درست تشخیص داده است.

اگر از منوی سمت چپ الگوریتم Support Vector Machine را انتخاب کنیم و از منو راست 40 درصد را و دکمه دیدن نتیجه را بزنیم جدول متفاوتی دریافت میکنیم که در شکل 4-5 آورده شده است. اعداد مربوط به هر کلاس اینبار می­توانند مثبت یا منفی باشند. همانطور که در فصل قبل توضیح داده شد الگوریتم ماشین بردار پشتیبان دسته­ها را با خط جدا می­کند، عدد های بیان شده در جدول شکل 4-5 بیانگر فاصله داده( همان متنی که وارد کرده ایم) تا این خط ها را بیان می­کنند، هرچه این عدد بزرگتر باشد ( اعداد را به صورت حقیقی فرض کنید و مقایسه کنید، یعنی منفی و مثبت بودن معنی دارد و در مقایسه لحاظ می­شود) میزان تعلق داده به کلاس مربوطه بیشتر است. کلاس تشخیص داده شده توسط الگوریتم بزرگترین عدد از بین این اعداد است. در این مثال خاص تنها کلاس خوشحال عددی مثبت دارد و بقیه منفی هستند، جواب الگوریتم به پرسش ما نیز کلاس خوشحال است.



شکل 4-5: برنامه سمت کاربر-پاسخ الگوریتم ماشین بردار پشتیبان

و اما در آخر اگر از منو سمت چپ الگوریتم Convolution Neural Network را انتخاب کنیم و از منو سمت راست مقدار 10 درصد علامت بزنیم و دکمه دیدن نتیجه را فشار دهیم، خروجی را در شکل 4-6 می­توانید مشاهده کنید. جدول دریافت شده از سامانه مشابه جدول شکل 4-4 و الگوریتم بیز ساده است و اعداد هر کلاس نشان دهنده میزان تعلق به آن کلاس هستند که برحسب درصد بیان شده­اند و بدیهی است که جمع تمامی آن­ها 100 خواهد بود. برای این متن ورودی خاص مدل آموزش داده شده مبتنی بر شبکه عصبی پیچشی میزان تعلق 98.48 درصدی به کلاس احساسی خوشحال را نشان می­دهد که درست است.



شکل 4-6: برنامه سمت کاربر-پاسخ الگوریتم شبکه عصبی پیچشی

### 4-4-2- برنامه سمت سرور

برنامه سمت سرور از طریق چارچوب Django و Django REST که به زبان پایتون می­باشد، پیاده سازی شده است. همانطور که در قسمت قبل توضیح داده شد درخواستی از سمت کاربر به سرور فرستاده می­شود که حاوی الگوریتم انتخابی و نسبت داده تست به کل داده ها است (درصد داده­های تست)، در سمت سرور به ازای هر الگوریتم و هر درصد داده­های تست یک مدل آموزش داده شده وجود دارد که از فایل خوانده می­شود و متن گرفته شده از سمت کاربر به عنوان ورودی به این مدل داده می­شود و خروجی که در بخش قبلی مفصل توضیح داده شد از آن گرفته می­شود و به سمت کاربر ارسال می­شود.

این مدل­ها به کمک زبان پایتون و کتابخانه های sklearn و keras ساخته شده­اند و در فصل 2 به طور کامل شرح داده شدند.

# فصل پنجم جمع‌بندي و پیشنهادات

در پايان گزارش‌هاي علمي و فني لازم است كه جمع‌بندي يا نتيجه‌گيري نهايي ارائه شود. در اين موارد مي‌توان آخرين فصل پایان نامه كه پیش از مراجع قرار مي‌گيرد را به اين امر اختصاص داد.



## 1-5- جمع­بندی

هدف این پروژه، تحلیل متن فارسی، که می­تواند نظر یک کاربر برای یک محصول در یک سایت فروشگاهی باشد، است. برای این منظور از دو الگوریتم مبتنی بر یادگیری ماشین بیز ساده و ماشین بردار پشتیبان و همچنین الگوریتم مبتنی بر یادگیری عمیق شبکه عصبی پیچشی استفاده کردیم. هر سه الگوریتم به همراه انواعشان در فصل دوم به طور کامل توضیح داده شدند. همچنین با استفاده از چند نسبت مختلف داده های تست به کل داده ها، اثر تعداد داده های تست بر دقت الگوریتم را آزمایش کردیم. با استفاده از زبان پایتون این سه الگوریتم را پیاده سازی کردیم و از آن­ها مدل ساختیم ؛ سپس مدل ساخته شده را با مجموعه داده های آموزشی که دارای برچسب بودند آموزش دادیم و در فایل ذخیره کردیم.

برنامه سمت کاربر به کمک چارچوب Vue.js که مبتنی بر زبان جاوا اسکریپت است پیاده سازی شده است. در این قسمت کاربر می­تواند الگوریتم مورد نظر و درصد داده های تست را انتخاب کند تا متنی که وارد می­کند بر اساس آن ها دسته­بندی شود و کلاس احساسی آن مشخص شود. کلاس های احساسی شامل خشمگین، عصبانی، خنثی، خوشحال و بسیار راضی است.

بخش سرور با چارچوب Django نوشته شده است و به زبان پایتون است. سمت سرور درخواست های بخش کاربر را گرفته و پاسخ می­دهد. این درخواست شامل یک متن فارسی، الگوریتم مورد استفاده و درصد داده های تست است، به کمک این اطلاعات مدل خواسته شده از فایل خوانده می­شود و متن فارسی به عنوان ورودی به مدل داده می­شود تا با توجه به آموزش هایی که دیده است یک حدس برای کلاس احساسی این متن بزند.

## 2-5- پیشنهاد

یکی از مشکلات استفاده از الگوریتم های دسته­بندی مبتنی بر یادگیری ماشین و یادگیری عمیق نیاز به تعداد زیادی داده آموزشی برچسب گذاری شده و کیفیت آن ها است. تهیه این داده ها نسبتا راحت است ولی برچسب­گذاری آن­ها نیازمند زمان و هزینه زیادی است. برای بهبود این مشکل می­توان از روش های افزایش داده استفاده کرد. در مجموعه داده ای که ما برای این پروژه داریم تعداد داده های هر کلاس متفاوت است، مثلا برای کلاس خوشحال 2185 داده و برای کلاس خشمگین تنها 40 داده موجود است. برای عملکرد بهتر می­توان به کلاس هایی که داده های کمی دارند داده اضافه کرد و از کلاس هایی که تعداد زیادی داده دارند داده هایی را حذف کرد. اینکار به تعادل مجموعه داده کمک خواهد کرد. همچنین می­توان به کمک مترجم گوگل در هر عبارت یک یا چند کلمه را به صورت تصادفی به انگلیسی ترجمه کرد، سپس ترجمه دیگر فارسی آن را که هم معنی کلمه مورد نظر است به جای آن داخل جمله قرار داد و اینگونه می­توان تعداد داده ها را دو برابر کرد.

پیشنهاد دیگری که برای بهبود دقت تحلیل احساسات می­توان مطرح کرد تحلیل و پردازش گروهی کلمات است، زیرا در مواقعی چند کلمه در کنار هم معنی متفاوتی از تک تک آن ها خواهند داشت؛ می­توان عبارت های پرکاربرد و ضرب المثل های فارسی را نیز اضافه کرد.

در صورت جمع آوری تعداد بیشتری داده می­توان معماری های دیگری از شکبه عصبی پیچشی را نیز آزمایش کرد. همچنین از روش های دیگر مبتنی بر یادگیری عمیق مانند شبکه عصبی بازگشتی یا انواع روش های مبتنی بر حافظه کوتاه مدت طولانی استفاده کرد.

**منابع و مراجع**

1. Bates, Joseph. "The role of emotion in believable agents." *Communications of the ACM* 37.7 (1994): 122-125.
2. Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.
3. Yu, Liang-Chih, et al. "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news." *Knowledge-Based Systems* 41 (2013): 89-97.
4. Kim, Hannah, and Young-Seob Jeong. "Sentiment classification using convolutional neural networks." *Applied Sciences* 9.11 (2019): 2347.
5. Dashtipour, Kia, et al. "Exploiting deep learning for persian sentiment analysis." *International Conference on Brain Inspired Cognitive Systems*. Springer, Cham, 2018.
6. Joulin, Armand, et al. "Fasttext. zip: Compressing text classification models." *arXiv preprint arXiv:1612.03651* (2016).
7. Heath, Nick. “What Is Machine Learning? Everything You Need to Know.” *ZDNet*, 14 Sept. 2018, www.zdnet.com/article/what-is-machine-learning-everything-you-need-to-know.
8. Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." *Proceedings of the 48th annual meeting of the association for computational linguistics*. 2010.
9. Hosseini, Pedram, et al. "SentiPers: A sentiment analysis corpus for Persian." *arXiv preprint arXiv:1801.07737* (2018).
10. Bojanowski, Piotr, et al. "Enriching word vectors with subword information." *Transactions of the Association for Computational Linguistics* 5 (2017): 135-146.
11. Ray, Sunil. “6 Easy Steps to Learn Naive Bayes Algorithm with Codes in Python and R.” *Analytics Vidhya*, 1 Apr. 2020, www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained.
12. ---. “Understanding Support Vector Machine(SVM) Algorithm from Examples (along with Code).” *Analytics Vidhya*, 15 Apr. 2020, www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code.
13. Bengio, Yoshua. *Learning Deep Architectures for AI (Foundations and Trends(r) in Machine Learning)*. Now Publishers Inc, 2009.
14. Song, Hyun Ah, and Soo-Young Lee. "Hierarchical representation using NMF." International conference on neural information processing. Springer, Berlin, Heidelberg, 2013.
15. Kushwaha, Amit. “Understanding CNN (Part-1).” Codementor, 25 July 2017, [www.codementor.io/@kushwahamit2016/understanding-cnn-part-1-ac369ogsm](http://www.codementor.io/@kushwahamit2016/understanding-cnn-part-1-ac369ogsm).
16. Mitchell, Tom. Machine Learning by Tom M. Mitchell (1997-03-01). McGraw-Hill Education; 1 edition (1997-03-01), 2020.
17. Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).
18. Pressman, Roger, and Bruce Maxim. " Software Engineering: A Practitioner's Approach 8th Edition." fig.4.7 p. 57.

# پيوست‌ها

# Abstract

The goal of this project is to analyze emotions on Persian text. This has been done with the help of simple multi-sentence Bayesian classifiers, linear support vector classifiers and convolutional neural network. In this project, the accuracy of each algorithm is investigated based on the number of training data. Client side and web-based application is designed and built with the help of Vue.js, which has the ability to take a Persian text, send information to the server side that implemented with the help of Django, and receive and display the emotional group of the text.

**Key Words:** sentiment analysis, machine learning, deep learning, natural language process



Amirkabir University of Technology  
(Tehran Polytechnic)

Department of Computer Engineering

MSc Thesis

Sentiment analysis on Persian text

By

Mohammadreza Shamshirgarha

Supervisor

Dr. Mostafa Haghir Chehreghani

October & 2020

1. 1 Sentiment analysis [↑](#footnote-ref-1)
2. 2 Pattern [↑](#footnote-ref-2)
3. 3 Opinion mining [↑](#footnote-ref-3)
4. 4 Multi-class [↑](#footnote-ref-4)
5. Machine learning [↑](#footnote-ref-5)
6. Supervised [↑](#footnote-ref-6)
7. Unsupervised [↑](#footnote-ref-7)
8. Naïve Bayes [↑](#footnote-ref-8)
9. Support vector machine [↑](#footnote-ref-9)
10. Convolutional neural network [↑](#footnote-ref-10)
11. Embedding matrix [↑](#footnote-ref-11)
12. Convolution [↑](#footnote-ref-12)
13. Max-pooling [↑](#footnote-ref-13)
14. Stride [↑](#footnote-ref-14)
15. Fully-connected [↑](#footnote-ref-15)
16. Deep autoencoder [↑](#footnote-ref-16)
17. Multilayer perceptron [↑](#footnote-ref-17)
18. Spam [↑](#footnote-ref-18)
19. Reliably [↑](#footnote-ref-19)
20. Feature vector [↑](#footnote-ref-20)
21. Active [↑](#footnote-ref-21)
22. Classification [↑](#footnote-ref-22)
23. Regression [↑](#footnote-ref-23)
24. Clustering [↑](#footnote-ref-24)
25. Commonalities [↑](#footnote-ref-25)
26. Cluster analysis [↑](#footnote-ref-26)
27. Preprocess [↑](#footnote-ref-27)
28. Feature extraction [↑](#footnote-ref-28)
29. Stopwords [↑](#footnote-ref-29)
30. Lemmatization [↑](#footnote-ref-30)
31. Sparse vectorization [↑](#footnote-ref-31)
32. Neural word embedding [↑](#footnote-ref-32)
33. Deep learning [↑](#footnote-ref-33)
34. Low-dimension [↑](#footnote-ref-34)
35. Tokenize [↑](#footnote-ref-35)
36. Bayes Theorem [↑](#footnote-ref-36)
37. Gaussian Naive Bayes [↑](#footnote-ref-37)
38. Multinomial Naive Bayes [↑](#footnote-ref-38)
39. Bernoulli Naive Bayes [↑](#footnote-ref-39)
40. Miss-classification [↑](#footnote-ref-40)
41. Kernel [↑](#footnote-ref-41)
42. Knowledge representation [↑](#footnote-ref-42)
43. Factors of variation [↑](#footnote-ref-43)
44. Bias [↑](#footnote-ref-44)
45. Neurons [↑](#footnote-ref-45)
46. Visual Cortex [↑](#footnote-ref-46)
47. Receptive Field [↑](#footnote-ref-47)
48. epoch [↑](#footnote-ref-48)
49. feed-forward [↑](#footnote-ref-49)
50. Tensor [↑](#footnote-ref-50)
51. [↑](#footnote-ref-51)
52. Framework [↑](#footnote-ref-52)