



سیگنال های و سیستم ها

پروژه درس سیگنال ها و سیستم ها

استاد درس: دکتر اخوان بی تقصیر

سید حمید عظیمی دخت ۹۹۳۳۲۱۳

۵ بهمن ۱۴۰۳

مقدمه

این پروژه با هدف تحلیل و شناسایی نت های موسیقی با استفاده از فایل های صوتی و الگوریتم KNN طراحی و پیاده سازی شده است. پروژه شامل دو بخش اصلی است: در بخش اول، فرکانس غالب نت های موسیقی استخراج شده و قطعه "تولد مبارک" بر اساس ترتیب مشخصی ساخته شد. در بخش دوم، از الگوریتم KNN برای طبقه بندی فایل های صوتی تست و شناسایی نت های موسیقی استفاده شد. این گزارش به تفصیل مراحل انجام پروژه را بررسی کرده و نحوه پیاده سازی آن را توضیح می دهد.

بخش اول: پردازش فایل های صوتی و ساخت قطعه موسیقی

در ابتدای پروژه، فایل های صوتی مربوط به نت های موسیقی از جمله Mi Re, Do و ... برای اکتاو اول و دوم در فرمت مشخص جمع آوری شدند. هر فایل نمایانگر یک نت خاص بوده و حاوی صدای مشخصی است که باید تحلیل می شود. برای این منظور، ابتدا از کتابخانه Pydub برای خواندن فایل های صوتی استفاده شد. فایل ها ممکن است به صورت استریو ضبط شده باشند؛ بنابراین، در صورتی که فایل صوتی شامل دو کانال باشد، تنها یکی از کانال ها استخراج و برای تحلیل استفاده می شود.

پس از استخراج داده های صوتی، تحلیل فوریه (FFT) بر روی نمونه های صوتی انجام شد. این تحلیل امکان تبدیل سیگنال زمانی به سیگنال فرکانسی را فراهم می کند. از نتایج FFT، فرکانس های غالب استخراج شدند که این فرکانس ها نشان دهنده نت اصلی هر فایل هستند. با این روش، فرکانس مشخص هر نت مانند Do یا Re به صورت دقیق تعیین شد.

در گام بعدی، ترتیب مشخصی از نت ها بر اساس فایل صوتی "تولد مبارک" تعیین شد. برای ساخت این قطعه، فرکانس های استخراج شده با توجه به مدت زمان هر نت مرتب شدند. در نهایت، فایل صوتی نهایی با کنار هم قرار دادن فرکانس ها و مدت زمان های مشخص ساخته شد. نتیجه این بخش، اجرای موفق قطعه "تولد مبارک" بود که به طور کامل صحت یابی شد.

تولدت مبارک



شکل ۶

شکل ۱: قطعه آهنگ ساخته شده از نوت ها

فرکانس های غالب نت ها

در این بخش، فرکانس غالب برای هر نت موسیقی استخراج شده از فایل های صوتی آورده شده است. فرکانس های محاسبه شده به شرح زیر هستند:

Do octave ۱: •
519.45 Hz

Do octave ۲: •
1060.15 Hz

Fa octave ۱: •
705.62 Hz

La octave ۱: •
884.96 Hz

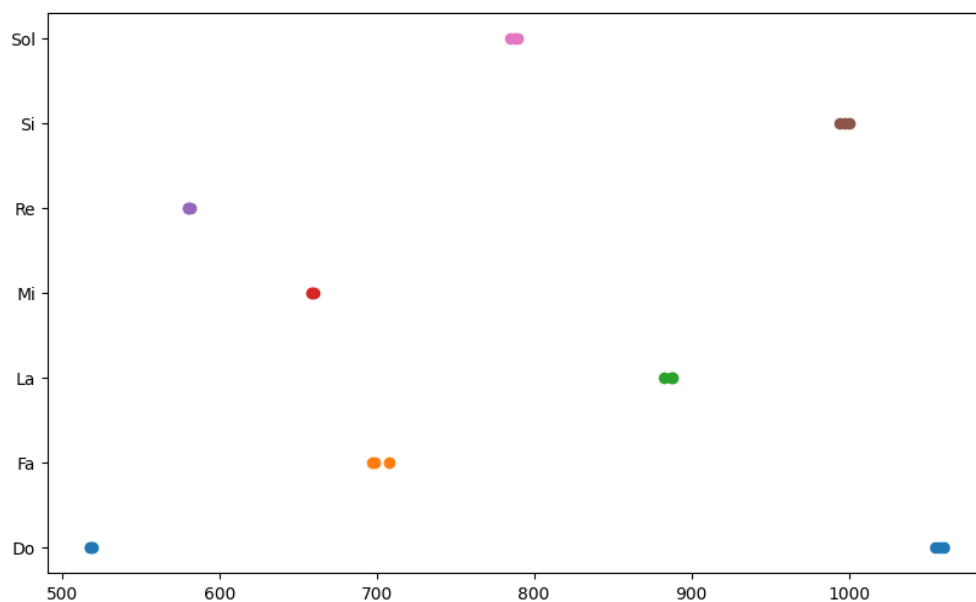
Mi octave ۱: •
660.12 Hz

Re octave ۱: •
579.87 Hz

Si octave ۱: •
996.89 Hz

Sol octave ۱: •
785.17 Hz

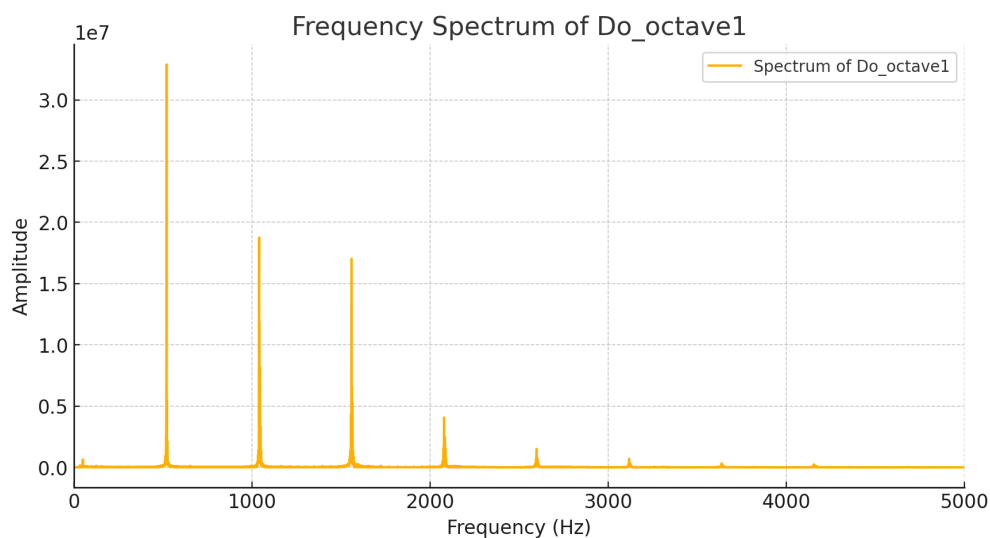
این مقادیر بیانگر فرکانس های غالب استخراج شده از طیف فرکانسی هر فایل صوتی هستند و به عنوان ویژگی کلیدی در شناسایی نت های موسیقی استفاده شده اند.



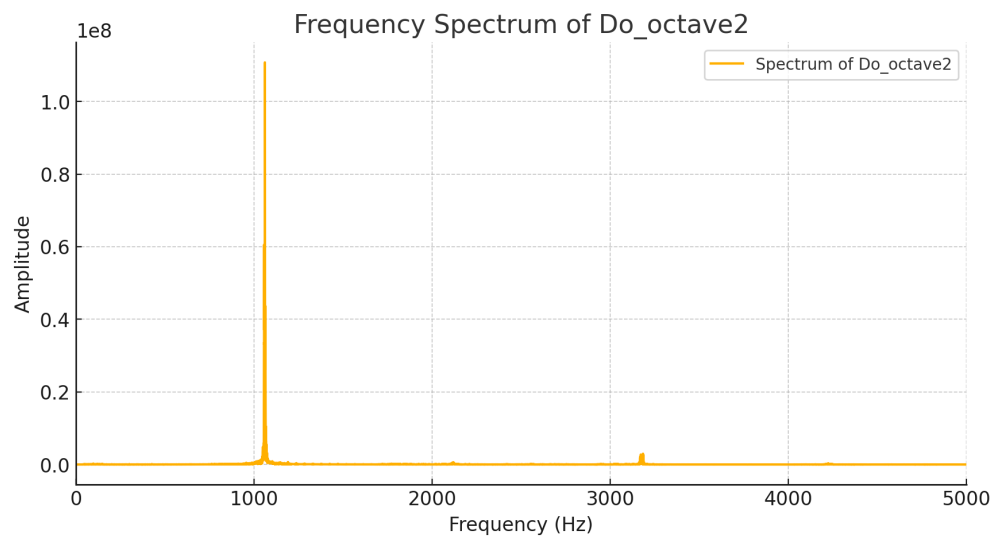
شکل ۲: فرکانس غالب هر نوت

ساخت آهنگ تولدت مبارک

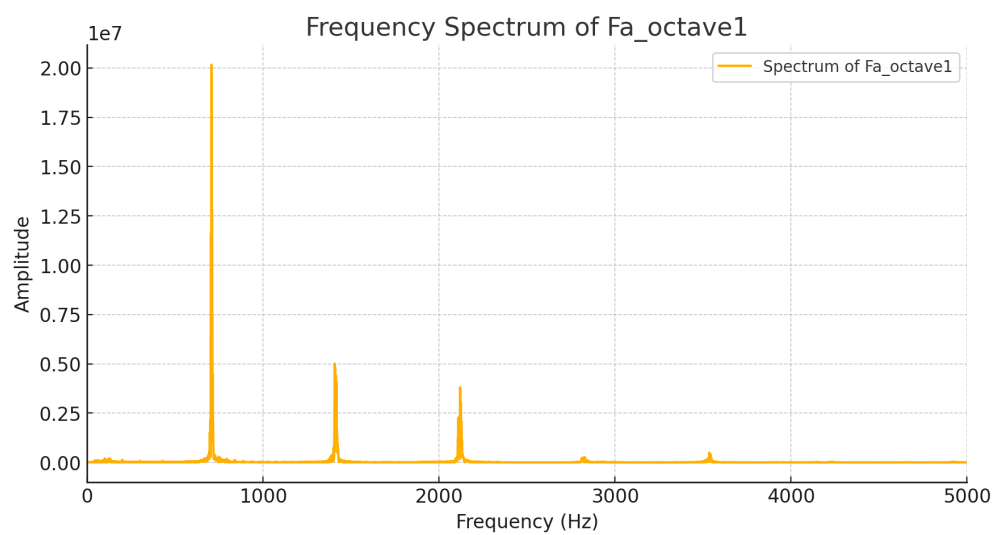
برای شناسایی کَشش زمانی هر نت، ابتدا مدت زمان فایل‌های صوتی محاسبه شد. سپس بیشترین و کمترین مقادیر کَشش زمانی بررسی شدند تا محدوده زمانی نت‌ها مشخص شود. از آنجایی که قطعه “تولدت مبارک” تنها شامل سه کَشش زمانی مختلف (مانند نت سیاه، چنگ و سیاه نقطه‌دار) بود، تصمیم گرفته شد کَشش‌های زمانی به سه گروه دسته‌بندی شوند. برای دسته‌بندی کَشش‌های زمانی به سه گروه، از الگوریتم k -نزدیک‌ترین همسایه‌ها (KNN) با تعداد خوشه‌های $k = 3$ استفاده شد. این الگوریتم توانست کَشش زمانی هر نت را به یکی از گروه‌های مشخص (مانند سیاه، چنگ و نقطه‌دار) تخصیص دهد. در نهایت، اطلاعات به دست آمده از کَشش زمانی و نت‌ها برای ساخت قطعه موسیقی استفاده شد.



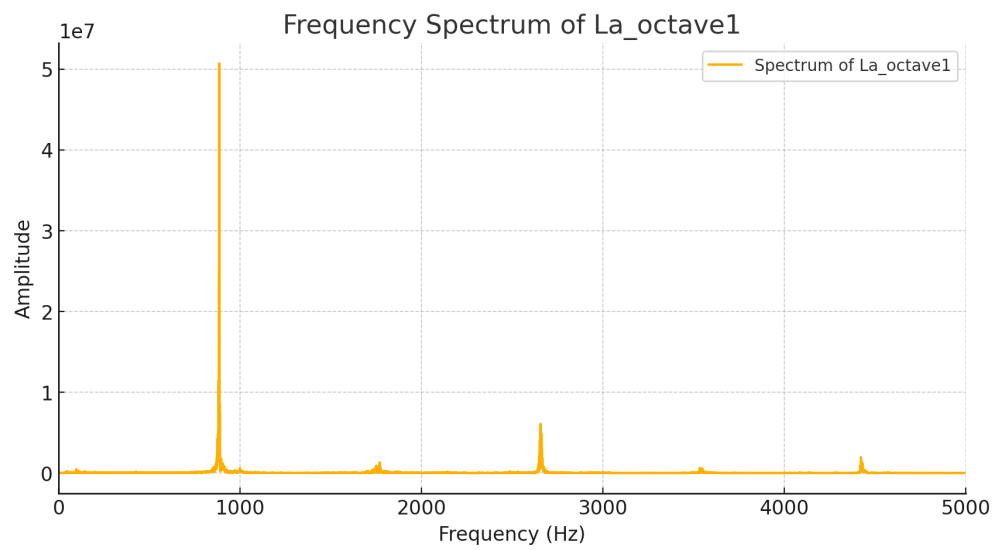
شکل ۳: طیف فرکانسی نوت دو اکتاو اول



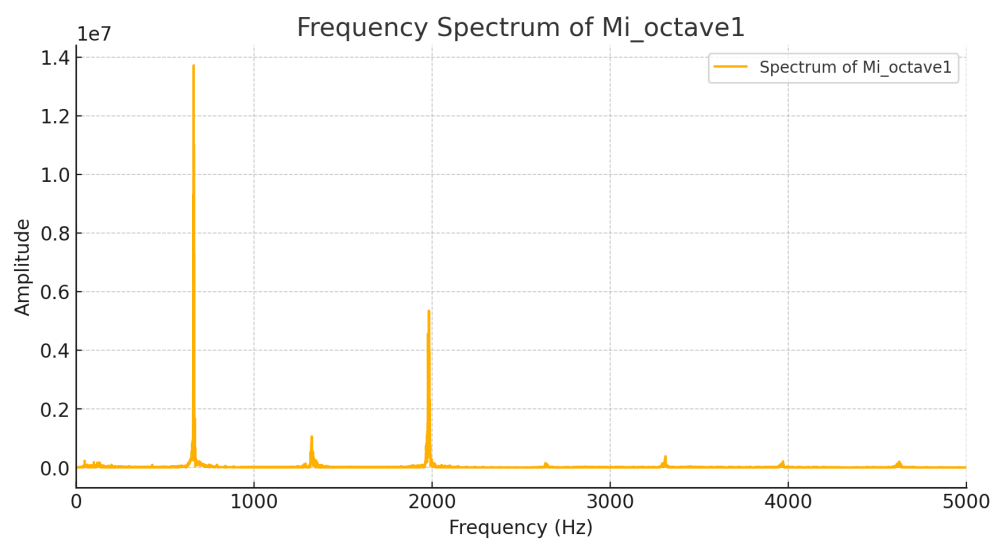
شکل ۴: طیف فرکانسی نوت دو اکتاو دوم



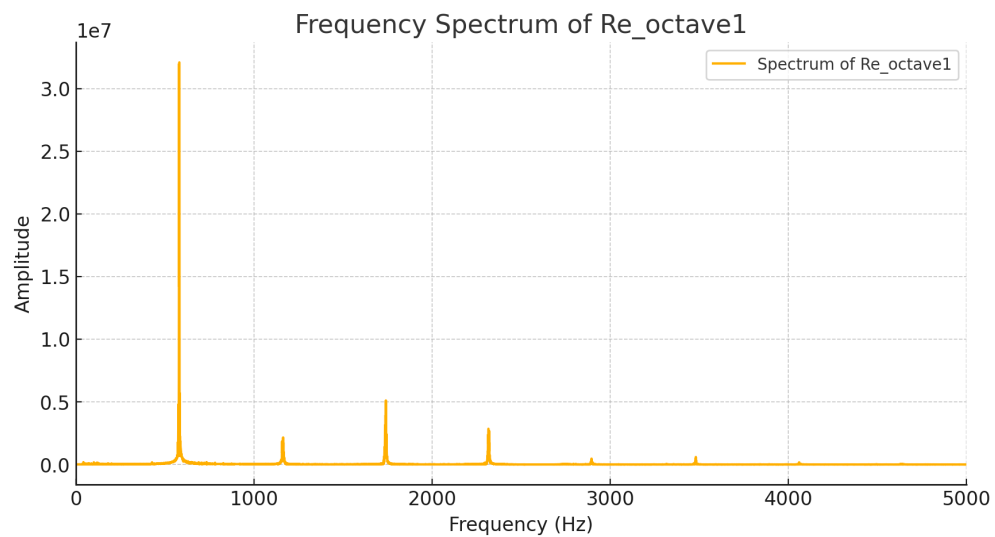
شکل ۵: طیف فرکانسی نوت فا



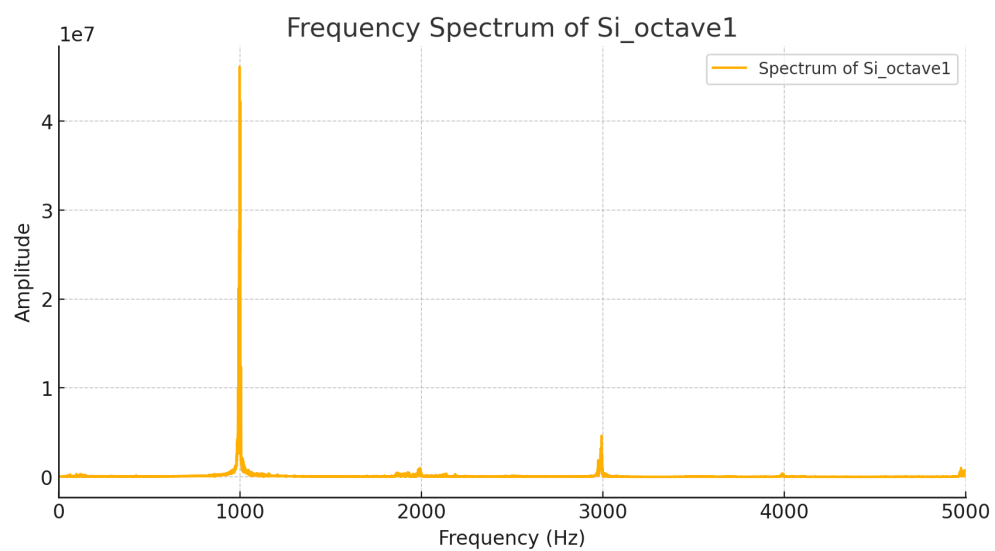
شکل ۶: طیف فرکانسی نوت لا



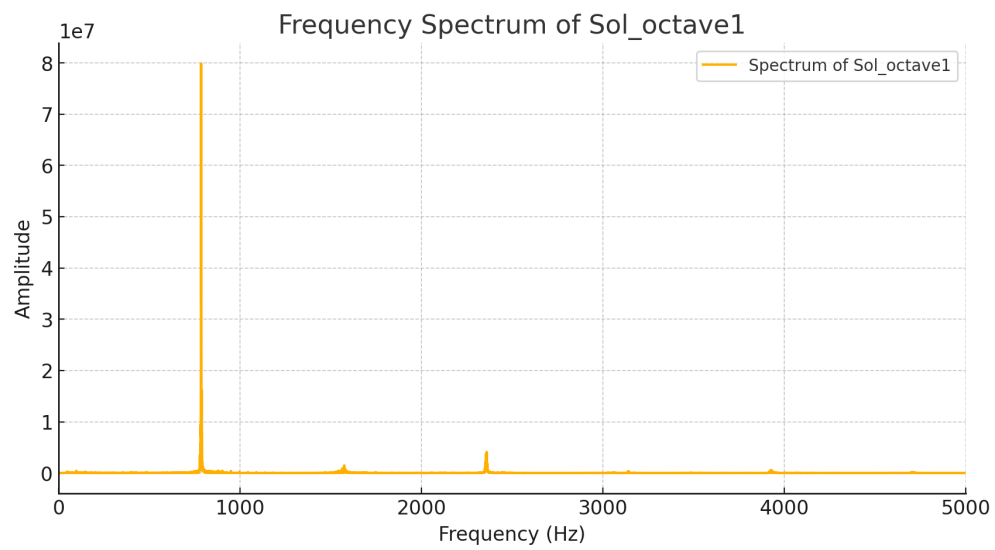
شکل ۷: طیف فرکانسی نوت می



شکل ۸: طیف فرکانسی نوت ر



شکل ۹: طیف فرکانسی نوت سی



شکل ۱۰: طیف فرکانسی نوت سل

بخش دوم: شناسایی نت‌ها با استفاده از الگوریتم KNN

در این بخش از پروژه، هدف شناسایی نت‌های مربوط به فایل‌های صوتی تست بود که فرکانس آنها مشخص نشده بود. برای این منظور، از الگوریتم K-Nearest Neighbors (KNN) که یکی از روش‌های ساده و موثر یادگیری تحت نظارت است، استفاده شد. ابتدا، داده‌های صوتی شامل فایل‌های مشخص برچسب‌دار (داده‌های آموزشی) و فایل‌های تست (بدون برچسب) تقسیم شدند. داده‌های آموزشی شامل فایل‌هایی بود که فرکانس غالب آنها از بخش اول پروژه استخراج شده و به هر کدام برچسب نت مرتبط داده شده بود. فایل‌های تست نیز شامل صوت‌هایی بودند که فرکانس آنها باید تعیین و به کلاس‌های مناسب (نت‌های موسیقی) تخصیص داده می‌شد. الگوریتم KNN برای شناسایی هر فایل تست، فرکانس آن را با فرکانس‌های فایل‌های آموزشی مقایسه می‌کند. با استفاده از معیار فاصله اقلیدسی، نزدیک‌ترین فرکانس‌ها به فایل تست پیدا شده و بر اساس رای‌گیری اکثریت، برچسب کلاس مشخص می‌شود. در این پروژه، مقدار k ، که نشان‌دهنده تعداد همسایگان نزدیک است، برابر با ۴ انتخاب شد. این مقدار با توجه به تعادل بین دقت و پیچیدگی محاسباتی تعیین شده است. از آنجایی که فایل‌های تست فاقد برچسب بودند، امکان ارزیابی مستقیم دقت مدل وجود نداشت. اما خروجی الگوریتم KNN بررسی شد و برچسب‌های پیش‌بینی‌شده به طور منطقی و متناسب با فرکانس‌های آموزشی مرتبط ارزیابی شدند. این روش امکان شناسایی دقیق نت‌های فایل‌های تست را فراهم ساخت.

انتخاب مقدار K برای الگوریتم KNN

برای تعیین مقدار بهینه K در الگوریتم k -نزدیک‌ترین همسایه‌ها، (KNN) دقت اعتبارسنجی برای مقادیر مختلف K محاسبه شد. نتایج به صورت زیر است:

- $K = 1$: دقت اعتبارسنجی 1.0 یا 100%
- $K = 2$: دقت اعتبارسنجی 1.0 یا 100%
- $K = 3$: دقت اعتبارسنجی 1.0 یا 100%
- $K = 4$: دقت اعتبارسنجی 1.0 یا 100%
- $K = 5$: دقت اعتبارسنجی 1.0 یا 100%
- $K = 6$: دقت اعتبارسنجی 0.5 یا 50%
- $K = 7$: دقت اعتبارسنجی 0.5 یا 50%
- $K = 8$: دقت اعتبارسنجی 0.5 یا 50%

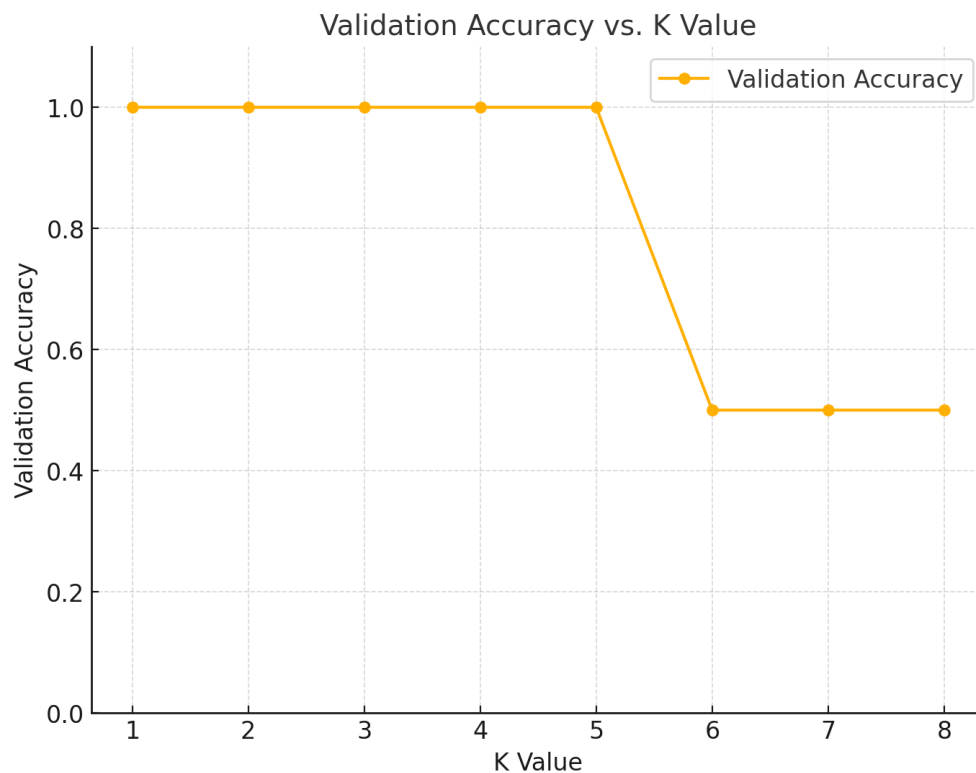
همان‌طور که در نمودار مربوطه مشاهده می‌شود، دقت اعتبارسنجی در مقادیر $K = 1$ تا $K = 5$ برابر 1.0 است، که نشان‌دهنده عملکرد عالی الگوریتم در این مقادیر است. با این حال، از $K = 6$ به بعد، دقت به 0.5 کاهش می‌یابد. این کاهش به دلیل تأثیر منفی همسایگان دورتر و نادرست بر پیش‌بینی‌ها است.

نتیجه‌گیری

با توجه به این نتایج، مقدار بهینه K بین 1 تا 5 قرار دارد. انتخاب مقدار $K = 4$ می‌تواند تعادلی مناسب بین دقت بالا و پیچیدگی محاسباتی فراهم کند.

پیش‌بینی برچسب‌های داده‌های تست با الگوریتم KNN

در این بخش، نتایج پیش‌بینی الگوریتم k -نزدیک‌ترین همسایه‌ها (KNN) برای داده‌های تست ارائه شده است. هر فایل صوتی تست شامل یک نت موسیقی است که الگوریتم بر اساس ویژگی‌های صوتی، برچسب مربوطه را برای آن پیش‌بینی کرده است. نتایج پیش‌بینی به صورت زیر است:



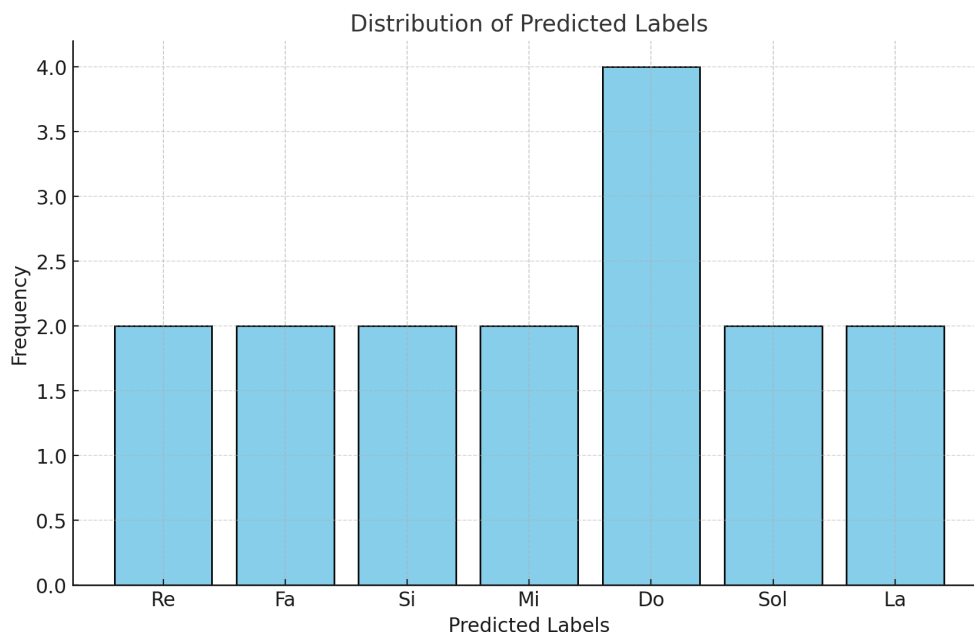
شکل ۱۱: نمودار دقت بر اساس مقدار k

پیش‌بینی‌های الگوریتم

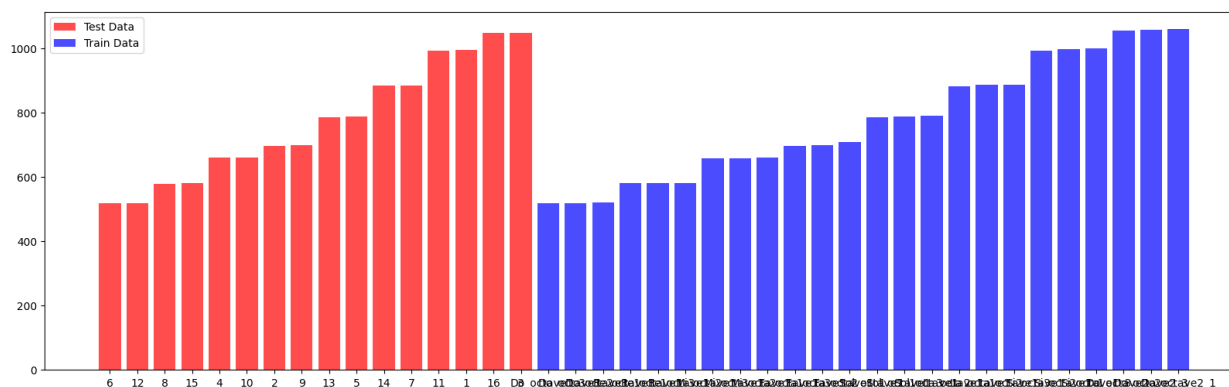
- فایل ۸.m۴a: پیش‌بینی شده به عنوان نت. Re.
- فایل ۹.m۴a: پیش‌بینی شده به عنوان نت. Fa.
- فایل ۱۱.m۴a: پیش‌بینی شده به عنوان نت. Si.
- فایل ۱۰.m۴a: پیش‌بینی شده به عنوان نت. Mi.
- فایل ۱۲.m۴a: پیش‌بینی شده به عنوان نت. Do.
- فایل ۱۳.m۴a: پیش‌بینی شده به عنوان نت. Sol.
- فایل ۱۶.m۴a: پیش‌بینی شده به عنوان نت. Do.
- فایل ۱۴.m۴a: پیش‌بینی شده به عنوان نت. La.
- فایل ۱۵.m۴a: پیش‌بینی شده به عنوان نت. Re.
- فایل ۱.m۴a: پیش‌بینی شده به عنوان نت. Si.
- فایل ۲.m۴a: پیش‌بینی شده به عنوان نت. Fa.
- فایل ۳.m۴a: پیش‌بینی شده به عنوان نت. Do.
- فایل ۷.m۴a: پیش‌بینی شده به عنوان نت. La.
- فایل ۶.m۴a: پیش‌بینی شده به عنوان نت. Do.
- فایل ۴.m۴a: پیش‌بینی شده به عنوان نت. Mi.
- فایل ۵.m۴a: پیش‌بینی شده به عنوان نت. Sol.

تحلیل پیش بینی ها

همان‌طور که مشاهده می‌شود، الگوریتم توانسته است برای هر فایل صوتی تست یک برچسب پیش‌بینی کند. برچسب "Do" بیشترین تعداد پیش‌بینی (۴ بار) را داشته است، در حالی که سایر نت‌ها تعداد پیش‌بینی کمتری دارند (۲ بار). این نتایج نشان می‌دهد که فایل‌های تست دارای توزیع نسبتاً متعادل از نت‌ها هستند و الگوریتم توانسته است به خوبی ویژگی‌های هر فایل را تشخیص دهد.



شکل ۱۲: نمودار تعدا پیش بینی های انجام شده برای داده های تست



شکل ۱۳: نمودار فرکانس های غالب برای داده های تست و آموزش

توضیح الگوریتم KNN

الگوریتم k -نزدیک‌ترین همسایه‌ها (K-Nearest Neighbors) یکی از الگوریتم‌های ساده و موثر در یادگیری ماشین است که برای مسائل طبقه‌بندی و رگرسیون استفاده می‌شود. این الگوریتم بر اساس اصل نزدیکی داده‌ها در فضای ویژگی‌ها عمل می‌کند و فرض می‌کند که نمونه‌های مشابه، کلاس‌های مشابهی دارند.

نحوه کارکرد الگوریتم KNN

۱. آماده‌سازی داده‌ها: الگوریتم KNN به داده‌های برچسب‌دار (داده‌های آموزشی) نیاز دارد. هر نمونه از داده‌های آموزشی دارای یک بردار ویژگی است که ویژگی‌های نمونه (مانند فرکانس غالب در این پروژه) را توصیف می‌کند و یک برچسب که کلاس نمونه (مانند نام نت موسیقی) را مشخص می‌کند.

۲. محاسبه فاصله: برای پیش‌بینی کلاس یک داده جدید (داده تست)، الگوریتم فاصله بین داده تست و تمامی داده‌های آموزشی را محاسبه می‌کند. یکی از رایج‌ترین معیارها برای اندازه‌گیری فاصله، فاصله اقلیدسی است که به صورت زیر محاسبه می‌شود:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

در اینجا، x_i و y_i مقادیر ویژگی‌های نمونه تست و نمونه آموزشی در بُعد i هستند.

۳. انتخاب نزدیک‌ترین همسایه‌ها: پس از محاسبه فاصله‌ها، k نمونه آموزشی که کمترین فاصله را با داده تست دارند به عنوان نزدیک‌ترین همسایه‌ها انتخاب می‌شوند. مقدار k از پیش تعیین شده است و نقش مهمی در عملکرد الگوریتم دارد:

- اگر k خیلی کوچک باشد، الگوریتم ممکن است به نویز حساس شود.
- اگر k خیلی بزرگ باشد، الگوریتم ممکن است دقت خود را از دست بدهد زیرا اطلاعات زیادی از همسایگان دورتر در تصمیم‌گیری دخیل می‌شود.

۴. رای‌گیری یا میانگین‌گیری:

- در مسائل طبقه‌بندی: الگوریتم بر اساس رای‌گیری اکثریت کلاس‌های k همسایه، کلاس داده تست را تعیین می‌کند. به عبارت دیگر، کلاسی که بیشترین تعداد همسایگان را دارد به عنوان پیش‌بینی انتخاب می‌شود.
- در مسائل رگرسیون: الگوریتم میانگین مقدار ویژگی هدف (خروجی) k همسایه را به عنوان خروجی پیش‌بینی می‌کند.

۵. خروجی نهایی: کلاس یا مقدار پیش‌بینی‌شده به داده تست اختصاص داده می‌شود.

مزایا و معایب الگوریتم KNN

مزایا:

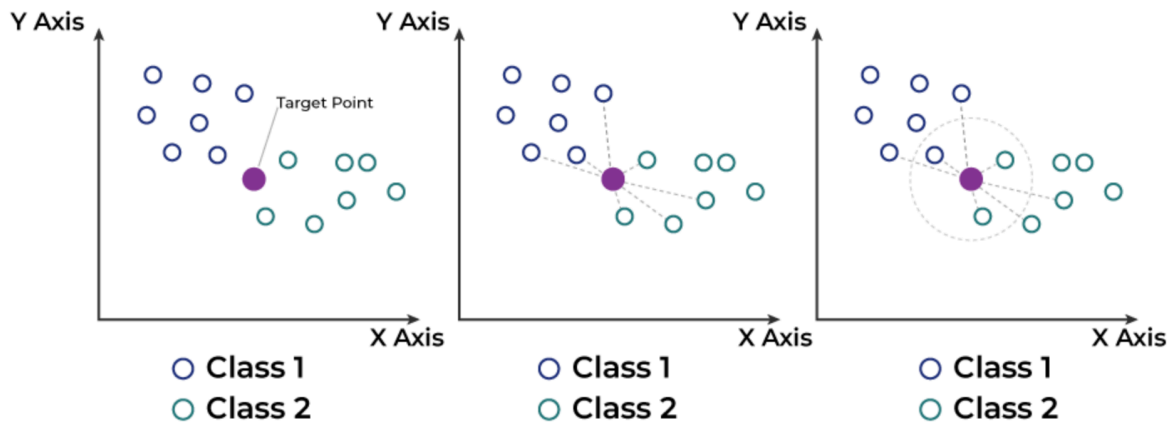
- ساده و قابل فهم.
- نیاز به هیچ فرآیند یادگیری پیچیده‌ای ندارد و مستقیماً از داده‌ها استفاده می‌کند.
- مناسب برای داده‌های چندکلاسه.

معایب:

- حساس به نویز، به خصوص اگر داده‌های آموزشی تمیز نباشند.
- پیچیدگی محاسباتی بالا برای داده‌های بزرگ، زیرا محاسبه فاصله برای تمام نمونه‌ها زمان‌بر است.
- عملکرد وابسته به انتخاب مناسب k و مقیاس داده‌ها.

نکات مهم در استفاده از KNN

- انتخاب مقدار k : مقدار k معمولاً با آزمون و خطا تعیین می‌شود. در این پروژه، k برابر با ۴ انتخاب شد که تعادلی مناسب بین دقت و پیچیدگی محاسباتی فراهم می‌کند.
- نرمال‌سازی ویژگی‌ها: چون فاصله‌ها به مقیاس ویژگی‌ها وابسته هستند، بهتر است داده‌ها نرمال‌سازی شوند تا همه ویژگی‌ها تأثیر یکسانی در محاسبه فاصله داشته باشند.
- اندازه داده‌های آموزشی: تعداد داده‌های آموزشی باید به اندازه کافی بزرگ باشد تا الگوریتم بتواند الگوهای دقیق‌تری را شناسایی کند.



شکل ۱۴: نحوه کار الگوریتم KNN