**گزارش پروژه سيگنال**

**کتابخانه‌های مورد استفاده**

* Os برای کار با فایل‌ها و پوشه‌ها.
* Numpy برای عملیات عددی.
* scipy.io.wavfile برای خواندن و نوشتن فایل‌های صوتی.
* scipy.fft برای تبدیل فوریه و معکوس فوریه.
* scipy.signal.resample برای تغییر نرخ نمونه‌برداری (تغییر سرعت صوت).
* matplotlib.pyplot برای رسم نمودارها.

**توابع تعریف‌شده**

1. **read\_voie** خواندن فایل صوتی و محاسبه تبدیل فوریه و فرکانس‌های مربوطه.
2. **change\_voice\_speed** تغییر سرعت صوت با استفاده از تابع resample.
3. **low\_pass\_filter** فیلتر پایین‌گذر برای حذف فرکانس‌های بالا (نویز).
4. **reverse\_voie** معکوس کردن داده‌های صوتی.
5. **mix\_voices** ترکیب چندین داده صوتی با طول‌های مختلف.
6. **write\_voice** نوشتن داده‌های صوتی به یک فایل..
7. **ensure\_dir** ایجاد پوشه‌ها در صورت عدم وجود.
8. **plot\_frequency\_amplitude** رسم نمودار فرکانس/دامنه و ذخیره آن.
9. **plot\_spectrogrm**  رسم spectrogram و ذخیره آن.

**فرآیند اصلی**

1. **خواندن فایل صوتی**: فایل صوتی potc.wav خوانده می‌شود.
2. **رسم نمودارهای اولیه**:
   * نمودار فرکانس/دامنه
   * نمودار spectrogram
3. **حذف نویز**:
   * حذف نویز از داده‌های صوتی با استفاده از آستانه نویز و ذخیره داده‌های پاک‌سازی شده.
   * رسم نمودارهای فرکانس/دامنه و spectrogram برای داده‌های پاک‌سازی شده.
4. **تغییر سرعت صوت**:
   * تغییر سرعت صوت به دو برابر و نصف سرعت اصلی و ذخیره داده‌ها.
   * رسم نمودارهای فرکانس/دامنه و spectrogram برای داده‌های تغییر سرعت داده شده.
5. **معکوس کردن صوت**:
   * معکوس کردن داده‌های صوتی و ذخیره آن.
   * رسم نمودارهای فرکانس/دامنه و spectrogram برای داده‌های معکوس شده.
6. **ترکیب صوت‌ها**:
   * ترکیب داده‌های پاک‌سازی شده، داده‌های با سرعت دو برابر، داده‌های با سرعت نصف، و داده‌های معکوس شده.
   * ذخیره داده‌های ترکیبی.
   * رسم نمودارهای فرکانس/دامنه و spectrogram برای داده‌های ترکیبی.

**توضيحات بيشتر و دقيق تر در فابل report.ipynb**