Entrega 2

Recapitulación

En la entrega anterior identificamos el tipo de problema a enfrentar, las metodologías y herramientas a usarse en el desarrollo del proyecto, además de las preguntas de interés y métricas que serán usadas para evaluar y analizar nuestros modelos, de tal forma que logremos identificar o establecer relaciones entre los datos recolectados (posiciones de las articulaciones) y nuestra variable objetivo (tipo de movimiento).

Obtención de nuevos datos

Como se planeó en la anterior entrega, se obtuvieron más vídeos de los cinco movimientos que se van a intentar predecir. Se hicieron tres tomas de cada movimiento con diferentes personas de múltiples alturas y contexturas, para así incluir una mayor variabilidad en los modelos, sin embargo, también se consideró normalizar los datos para que estás características no sean determinantes y se pueda generalizar mejor el modelo.

Normalización

Para generalizar nuestro modelo hemos decidido normalizar los datos haciendo uso del StandardScaler que provee la librería scikit learn, empleando la siguiente fórmula: $z = (x - u)/s$ donde x es nuestra muestra actual, u y s son la media y desviación estándar del conjunto de entrenamiento. De esta forma nuestros modelos encontrarán características cuya distribución se asemejan a una normal, lo que en teoría ofrecerá mejores resultados y prevendrá el overfitting sobre individuos, es decir que solo funcione con los individuos que hicieron de modelos durante la recolección de datos.

Entrenamiento de modelos

Como modelos de clasificación se eligieron SVM, Random Forest y XGBoost, cada uno recibió una partición 70-30 para entrenamiento y testing y la configuración inicial de cada uno fue la siguiente:

- **SVM**
  - kernel='rbf'
  - C=1
  - gamma='scale'
- **Random Forest**
  - n_estimators=100
  - max_depth=10
- **XGBoost**
  - eval_metric='mlogloss'
  - n_estimators=100
  - max_depth=6
  - scale_pos_weight=1
  - learning_rate=0.1

Ajuste de hiper parámetros

Para el ajuste hiper parámetros se desarrolló una matriz de parámetros para cada modelo y un total de 3 pliegues y nuestra variable de puntaje fue la accuracy. Para cada modelo se decidió trabajar con los siguientes campos:

- **SVM**
  - **C:** Es un parámetro de penalización para elegir entre maximizar el margen y minimizar el error de clasificación, con un C mayor se reduce el margen y con uno menor incrementa la posibilidad de un error de clasificación.
  - **kernel:** Define la función que transforma los datos en un espacio de mayor dimensión para que SVM logre encontrar el hiperplano que separa los datos.
  - **gamma:** Controla la influencia de los puntos de entrenamiento en el modelo según su distancia.
- **Random Forest**
  - **n_estimators:** Define el número de árboles que se crearán.
  - **max_depth:** Establece la profundidad de cada árbol.
  - **min_samples_split:** Indica el número mínimo de muestras para dividir un nodo
  - **min_samples_leaf:** Indica el número mínimo de muestras que debe tener cada hoja
  - **bootstrap:** Define si se usarán o no muestras de bootstrap para construir los árboles
- **XGBoost**
  - **n_estimators:** Define el número de árboles que se crearán.
  - **max_depth:** Establece la profundidad de cada árbol.
  - **scale_pos_weight:** Ajusta penalizaciones entre clases cuando nuestros datos no se encuentran balanceados.
  - **learning_rate:** Controla la tasa de aprendizaje definiendo el tamaño de cada paso por iteración del modelo.

Los mejores hiper parámetros son:

- **SVM:** {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}
- **Random Forest:** {'bootstrap': True, 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
- **XGBoost:** {'learning_rate': 0.15, 'max_depth': 9, 'n_estimators': 200, 'scale_pos_weight': 1}

Resultados obtenidos

Inicialmente se obtuvieron las siguientes métricas para los modelos:

- SVM: Accuracy del 90.8%
- Random Forest: Accuracy del 97.4%
- XGBoost: Accuracy del 98.9%

Después de aplicar Grid Search para encontrar los mejores hiper parámetros se obtuvieron los siguientes resultados:

- SVM: Accuracy del 98.5%
- Random Forest: Accuracy del 97.9%
- XGBoost: Accuracy del 99.1%

Tememos que estamos enfrentando un problema de overfitting cuya solución va más allá del grid search y que se ve afectada aún más por parámetros como C, n_estimators, max_depth, learning_rate, min_samples_split y min_samples_leaf, los cuales alteran la forma en que los modelos clasifican y aprenden de las clases para lograr categorizarlas. Ofreciendo altos puntajes de accuracy provocados por un sobre ajuste de los datos. Esperamos solucionar el overfitting con la obtención de nuevos datos que hayan sido recuperados por compañeros ya que las diferentes formas de tomar muestras de datos deberían ayudar con la generalización del modelo en múltiples entornos y con un mayor número de personas.

Plan de despliegue

Para el uso del modelo en un contexto real será necesario contar con una cámara enfocando un corredor de al menos 4 metros con una silla al final de este. La cámara debe grabar a 30 fps la secuencia de movimientos del paciente: pararse de la silla, caminar hacia la cámara, dar la vuelta, caminar hacia la silla y sentarse. Al terminar se deberá enviar el video a la aplicación que implementa el modelo y este usará el insumo para determinar los movimientos realizados en cada momento y potencialmente realizar diagnóstico de enfermedades.

Impacto de la solución

Esperamos que el modelo desarrollado pueda ser usado en el contexto médico para la detección preliminar de condiciones físicas relacionadas con el movimiento de cada articulación del cuerpo humano como lo son el parkinson o la rigidez muscular. Esto podría traer ventajas al lograr una examinación sin la necesidad de un especialista en la salud presente, además que podría entregar resultados ideales en un menor tiempo. De esta forma habrá menos presión sobre los profesionales, optimizando la forma en se podrá gestionar tiempo, equipos y espacios. Sin embargo somos conscientes que dado el contexto en que se espera aplicar es de gran importancia ser precavidos buscando abarcar la mayor cantidad de positivos reales cuando se trata de condiciones médicas, esperando ofrecer un alto puntaje de recall. Además de los posibles usos que se alejen del propósito del modelo como lo puede ser la vigilancia sin consentimiento a través del seguimiento postural de la herramienta, vulnerando la privacidad de los afectados. Se hace entonces necesario dejar expresas las reglas y condiciones vinculadas al uso del modelo o incluso la protección del código fuente para prevenir estos escenarios.