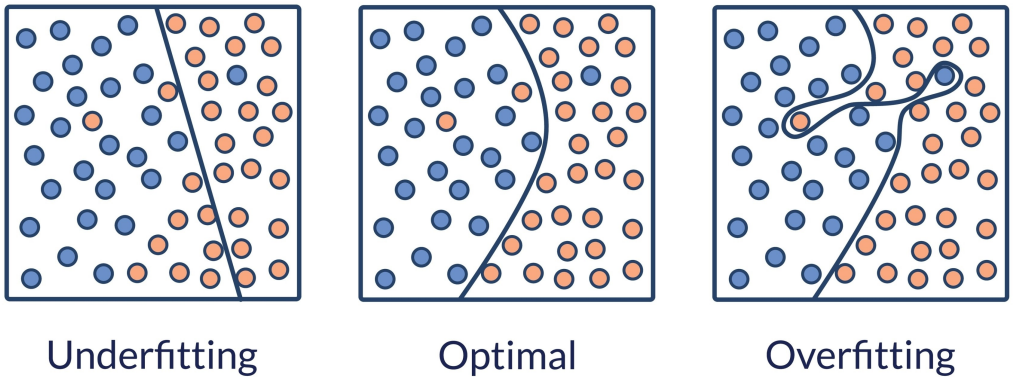
El Panorama del Machine Learning Parte 4

Prueba y Validación

- La única forma confiable de saber si un modelo generaliza bien es evaluándolo con datos no vistos.
- Evaluar solo con los datos de entrenamiento puede dar una falsa impresión de buen rendimiento.
- División Entrenamiento/Prueba
  - Conjunto de entrenamiento: usado para entrenar el modelo.
  - Conjunto de prueba: usado para evaluar el rendimiento.
- El rendimiento en el conjunto de prueba proporciona una estimación del error de generalización (también llamado error fuera de muestra).
- Si el error de entrenamiento es bajo pero el error de prueba es alto, el modelo probablemente está sobreajustando.
- Proporción de División Común: Una configuración típica es 80% entrenamiento / 20% prueba, aunque puede variar dependiendo del tamaño del conjunto de datos.

Ajuste de Hiperparámetros y Selección de Modelos

- El problema



  -
  - Confiar únicamente en el conjunto de prueba para el ajuste de hiperparámetros es peligroso.
  - Si ajustas repetidamente el modelo para minimizar el error del conjunto de prueba, lo estás adaptando a esos datos específicos.
  - Resultado: El modelo "sobreajusta" el conjunto de prueba, produciendo una tasa de error optimista pero un rendimiento pobre en datos verdaderamente nuevos (producción).

- Validación de retención:

  - Separa una porción del conjunto de entrenamiento para crear un conjunto de validación (o conjunto de desarrollo).
  - Proceso:
    - Entrena múltiples modelos candidatos (con diferentes hiperparámetros) en el conjunto de entrenamiento reducido.
    - Selecciona el mejor modelo basándose en el rendimiento en el conjunto de validación.

    - Re-entrena el mejor modelo en el conjunto de entrenamiento completo (entrenamiento reducido + validación).
    - Paso Final: Evalúa este modelo final en el conjunto de prueba para estimar el error de generalización.

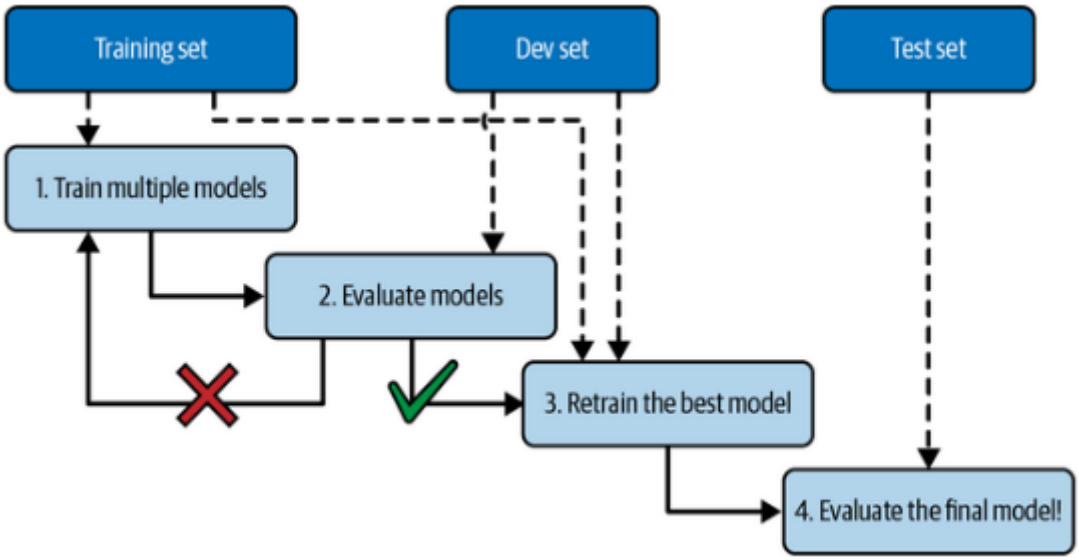


Figure 1-25. Model selection using holdout validation

- Compensaciones en el Tamaño del Conjunto de Validación:
    - Demasiado pequeño: Las evaluaciones se vuelven imprecisas, arriesgando la selección de un modelo subóptimo.
    - Demasiado grande: El conjunto de entrenamiento restante se vuelve demasiado pequeño. Dado que el modelo final se entrena en el conjunto completo, comparar candidatos entrenados en una fracción diminuta da una visión distorsionada de su potencial (como juzgar a un corredor de maratón basándose en un sprint).
- Validación Cruzada Repetida (La Solución):
    - Por qué usarla: Resuelve el dilema del tamaño del conjunto de validación.
    - Método: Usa muchos conjuntos de validación pequeños. Cada modelo se evalúa una vez por conjunto después de entrenar con el resto de los datos.
    - Beneficio: Promediar todas las evaluaciones proporciona una medida mucho más precisa del rendimiento.
    - Desventaja: El tiempo de entrenamiento se multiplica por el número de conjuntos de validación (computacionalmente costoso).

Desajuste de Datos

- El Contexto:
    - A menudo, tienes una gran cantidad de datos de entrenamiento de una fuente (ej., imágenes Web de alta calidad) pero tu aplicación objetivo usa datos de una fuente diferente (ej., imágenes Móviles borrosas).
    - Regla de Oro: Para medir el rendimiento del mundo real, tus conjuntos de Validación (Dev) y Prueba deben provenir exclusivamente de la distribución objetivo (imágenes Móviles).

- El Problema de Diagnóstico:
    - Si entrenas con imágenes Web y evalúas con imágenes Móviles (conjunto Dev), y el error es alto, estás ciego a la causa. No puedes saber si:
        1. El modelo está sobreajustado (no puede generalizar a ningún dato nuevo).
        2. El modelo sufre de Desajuste de Datos (no entiende el estilo específico de las imágenes Móviles).
- La Solución: El Conjunto "Train-Dev"
    - Crea un nuevo subconjunto llamado conjunto Train-Dev.
    - Fuente: Se extrae de los datos de Entrenamiento originales (imágenes Web).
    - Regla: El modelo no se entrena con él. Sirve como grupo de control para verificar la generalización dentro de la misma distribución de datos.
- El Flujo de Trabajo de Diagnóstico (Análisis de Brechas):
    - Comparar Error de Entrenamiento vs. Error de Train-Dev (Verificación de Varianza):
        - Escenario: El modelo funciona bien en los datos de Entrenamiento pero mal en los datos Train-Dev.
        - Observación: Ambos conjuntos provienen de la misma fuente (Web), pero el modelo falla en el no visto.
        - Conclusión: Sobreajuste (Alta Varianza). El modelo está memorizando, no aprendiendo.
        - Solución: Regularización, más datos, modelo más simple.
    - Comparar Error de Train-Dev vs. Error de Dev (Verificación de Desajuste):
        - Escenario: El modelo funciona bien en Train-Dev (Web) pero mal en Dev (Móvil).
        - Observación: El modelo generaliza bien dentro de la distribución Web, pero falla cuando la distribución cambia a Móvil.
        - Conclusión: Desajuste de Datos. El modelo no ha aprendido las características específicas del entorno de producción.
        - Solución: Hacer que los datos de entrenamiento se parezcan más a los datos de producción (Síntesis de Datos Artificial), ej., agregando ruido, desenfoque o simulando mala iluminación.