**Deployment**

Kita telah belajar bagaimana mengembangkan dan melatih sebuah model pada Google Colaboratory. Tentunya kita ingin agar model yang telah kita latih dapat terintegrasi dengan perangkat lunak lain. Misalnya kita ingin agar model kita dapat dipakai pada sebuah ponsel untuk memotret lalu mendeteksi penyakit pada tanaman cabai. Atau kita ingin membuat sebuah situs untuk mendeteksi jenis hewan pada sebuah gambar dan masalah lain yang lebih kompleks.

Proyek machine learning umumnya terdiri dari tiga fase berbeda, yaitu training, *saving model*, dan deployment. Pada tahap pertama, kita melakukan proses pelatihan atau training model dengan data yang kita miliki. Selain itu, kita juga menguji model dengan data uji dan melatihnya kembali sampai kita merasa puas dengan performanya. Kita telah belajar tentang tahapan ini di modul-modul sebelumnya.

Tahap selanjutnya, kita menyimpan model menjadi file yang bisa digunakan di server produksi. Tahap terakhir, kita menerapkan model yang telah disimpan tadi ke server produksi dan siap membuat prediksi pada data baru. Tahap terakhir ini disebut penerapan model pada tahap produksi atau deployment.

Kita bisa melakukan proses deployment pada tiga platform yaitu.

1. Mobile dan IoT (internet of things).  
   Di sini kita menjalankan proses deployment dengan menggunakan tf.lite (TensorFlow lite) pada perangkat mobile dan *embedded device*seperti Android, iOS, Edge TPU, dan Raspberry Pi. Pelajari caranya [di sini](https://www.tensorflow.org/lite).
2. Cloud menggunakan TensorFlow serving.  
   Service ini memudahkan penerapan algoritma dan eksperimen yang baru, sekaligus mempertahankan arsitektur server dan API yang sama. Jika Anda tertarik untuk mempelajarinya lebih lanjut, silakan buka [tautan](https://www.tensorflow.org/tfx/guide/serving) ini dan[tutorial](https://medium.com/free-code-camp/how-to-deploy-tensorflow-models-to-production-using-tf-serving-4b4b78d41700)berikut.
3. Browser dan Node.js dengan menggunakan TensorFlow.js.  
   TensorFlow.js merupakan library untuk machine learning pada JavaScript. Dengan library ini kita bisa membangun model ML dengan javascript, sekaligus menggunakannya langsung pada browser atau Node.js. Buka [tautan](https://www.tensorflow.org/js) berikut untuk mempelajari caranya.

Jika Anda tertarik untuk belajar lebih dalam dan berlatih *deploy* model machine learning, lanjutkan proses belajar Anda hingga kelas[Belajar Pengembangan Machine Learning](https://www.dicoding.com/academies/185) ya karena detailnya teruraikan di sana. Pada kelas ini kita akan berlatih mengembangkan model hingga ke tahap produksi.

Menarik, kan? Tetap semangat belajarnya, yah!

**Feedback**

Konsep feedback telah menjadi komponen penting dalam pemodelan sistem secara umum dan khususnya dalam siklus pengembangan perangkat lunak. *Feedback loops* adalah proses di mana perubahan atau output dari salah satu bagian sistem dikirimkan kembali ke dalam sistem sebagai input sehingga mempengaruhi tindakan atau output sistem selanjutnya. Jay Wright Forrester, seorang *computer engineer*dan *system scientist* menyatakan bahwainteraksi struktur fisik, arus informasi, dan proses keputusan menciptakan jaringan feedback loop yang menghasilkan dinamika sistem [24].

Ketika model Anda telah di-*deploy* di tahap produksi, sangat penting untuk selalu memonitor kinerja model Anda. Memonitor kinerja atau performa model dapat dilakukan dengan teknik yang sama saat kita mengembangkannya. Pada model klasifikasi, hal yang dimonitor adalah akurasinya terhadap data-data baru yang ditemui. Sedangkan pada model regresi, tingkat erornya yang dimonitor.

Kita juga bisa mendapatkan feedback dari sisi pengguna. Contohnya model kita dipakai pada sebuah aplikasi peminjaman uang untuk menentukan apakah seseorang dapat diberikan pinjaman atau tidak. Dan ternyata, ada beberapa keluhan dari pengguna yang mengatakan pengajuan pinjaman mereka ditolak padahal mereka orang yang tergolong kredibel. Hal seperti inilah yang menunjukkan kenapa memonitor dan mengumpulkan feedback sangat penting setelah sebuah model diterapkan di tahap produksi.

Saat dijalankan dengan benar, *feedback loops* dapat membantu kita membuat model menjadi lebih baik (*feedback positif*). Tetapi, feedback loops juga bisa berujung pada konsekuensi negatif yang tidak diinginkan seperti bias, atau pengukuran performa model yang tidak akurat (*feedback negatif*).

Berikut adalah contoh kasus feedback positif dan negatif.

1. Feedback Positif  
   Pada tahun 2017 [Netflix](https://www.theatlantic.com/entertainment/archive/2017/03/netflix-believes-in-the-power-of-thumbs/520242/) mengubah sistem *rating-*nya dari penilaian berbasis bintang (*star rating*) menjadi sistem yang lebih sederhana: *thumbs up*dan *thumbs down.*Hal ini dilakukan berdasarkan feedback bahwa star rating dianggap kurang merepresentasikan pengalaman pengguna dalam menonton film di Netflix. Perubahan menjadi thumbs rating yang lebih mudah dan sederhana ini mendorong lebih banyak pengguna untuk memberikan rating sehingga akan berpengaruh pada akurasi sistem rekomendasi.
2. Feedback negatif  
   Anda sedang membuat model machine learning untuk mengidentifikasi area berisiko tinggi yang memerlukan lebih banyak pengawasan oleh pihak kepolisian. Model Anda menemukan fakta bahwa daerah A memiliki banyak tindakan kejahatan selama 3 bulan terakhir. Dari hasil ini kemudian diputuskan untuk menambahkan lebih banyak polisi di daerah ini. Dengan ekstra personil, penangkapan tersangka tindak kejahatan juga semakin bertambah. Hal ini memvalidasi prediksi model bahwa daerah ini memiliki level kriminalitas yang di atas rata-rata. Kemudian anggota keamanan pun diterjunkan di daerah ini. Demikian seterusnya dan siklusnya terus berlanjut.  
   Kasus ini adalah contoh bias pada model machine learning. Saat data dikirim kembali ke sistem, prediksi bahwa daerah A memiliki tingkat kejahatan tinggi (sehingga membutuhkan lebih banyak polisi) menjadi semakin kuat. Feedback ini menciptakan bias yang membuat hasil kumulatifnya bisa menyebabkan masalah di kemudian hari. Oleh karena itu, sebelum mengambil keputusan, kita perlu memahami alasan mengapa dan bagaimana pola tersebut ada.

Sampai di sini Anda telah mendalami feedback pada model machine learning. Anda juga telah paham mengapa kita perlu mengumpulkan feedback untuk terus memonitor performa model. Jangan lupa mengecek ulang apakah feedback tersebut termasuk feedback positif (yang bisa memperkuat model) atau negatif (menghasilkan bias) sebelum kita memasukkannya ke dalam model.

Memang tidak mungkin untuk sama sekali menghilangkan bias pada model machine learning. Kita perlu menilai dan mengidentifikasi potensi masalah yang terjadi pada sistem agar bisa mengatasi atau menghindarinya sedini mungkin. Perhatikan pola feedback dan berhati-hatilah dalam mengambil keputusan.