Machine learning to deliver blood more reliably

The Iron Man(drone) of Rwanda

Parikshit Deshpande, Abhishek Akkur
parikshd@stanford.edu, abhakk01@stanford.edu

STANFORD UNIVERSITY

Objective

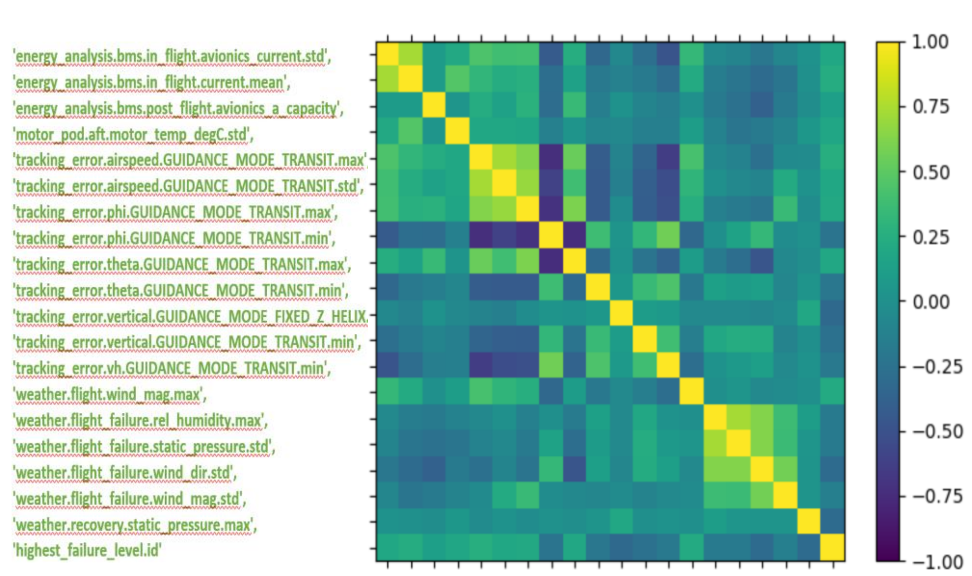
- Use Machine learning for predictive maintenance of drones for Zipline Inc.
- Predict the probability of a successful flight based on historical telemetry data.
- Categorize the failure into mission failure or flight failure.
- Based on prediction identify the parts which can cause the flight to fail.
- Use the model as a prototype to analysis flights in production.

Telemetry Data

- During the flight, telemetry data like energy analysis (current drawn from battery, etc.), flight tracking information and weather information is collected.
- After each flight, this data along with true labels is analyzed and stored in AWS S3.
- We have ~ 3000 real flight data captured, and around ~20 flights are added every day.
- Each flight captures 1000~1200 features varying on mission status.

Feature Reduction

- After cleansing the data (strings, nan, constant columns) we were left with ~ 700 features.
- We used the correlation matrix to remove highly correlated features and uncorrelated features with output label.
- Correlation matrix and final features **reduced to 18 features.**



Train and Test data

- **Labels**: Label highest_failure_level is categorized as
- 1-Success, 2-Mission Failure (flight returned without delivery) and 4- Flight Failure (flight deploys parachute)
- **Data size**: ~3000 flights as train set. ~250 flights as test set.

Models

- **Locally weighted linear regression:**

$$\sum_i w^{(i)}(y^{(i)} - \theta^T x^{(i)})^2.$$

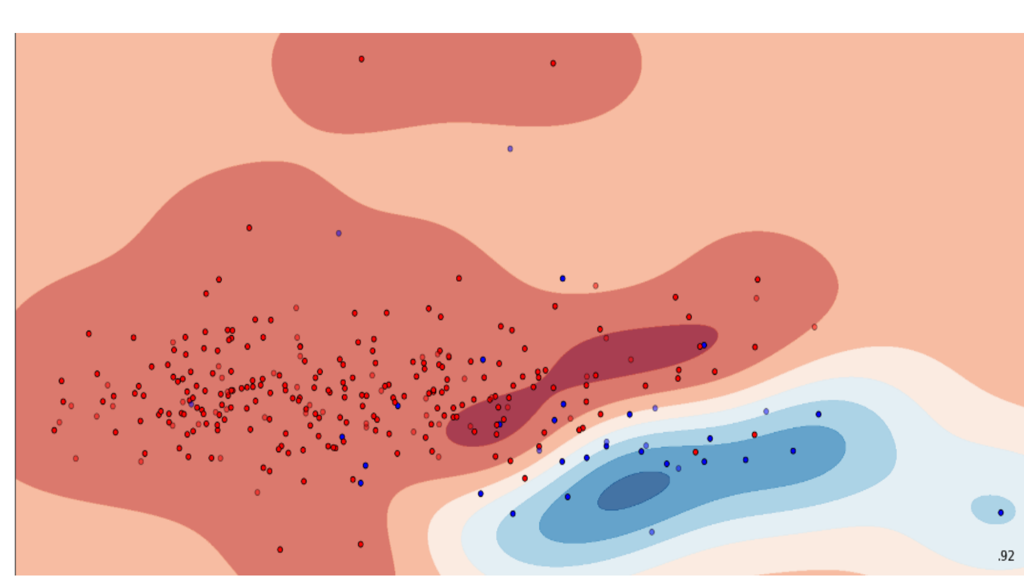
- **Logistic regression:**
  - Normal Equation $\theta = (X^T X)^{-1} X^T \vec{y}.$
  - Gradient Descent (L1, L2 Regularization)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)})).$$

- **Support Vector Machine**

$$f(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i y_i \Phi(\mathbf{s}_i)\cdot\Phi(\mathbf{x}) + b = \sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b$$

  - With Linear kernel $k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
  - With RBF kernel $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)$
- **Decision Trees & Random Forests**
- **Principal Component Analysis (PCA):**
  - To visualize the data and select appropriate model we performed PCA on the features.
  - First 2 principal components were found using the top 2 eigenvectors.
  - SVM RBF with 2 principal components plotted the contour with labels:

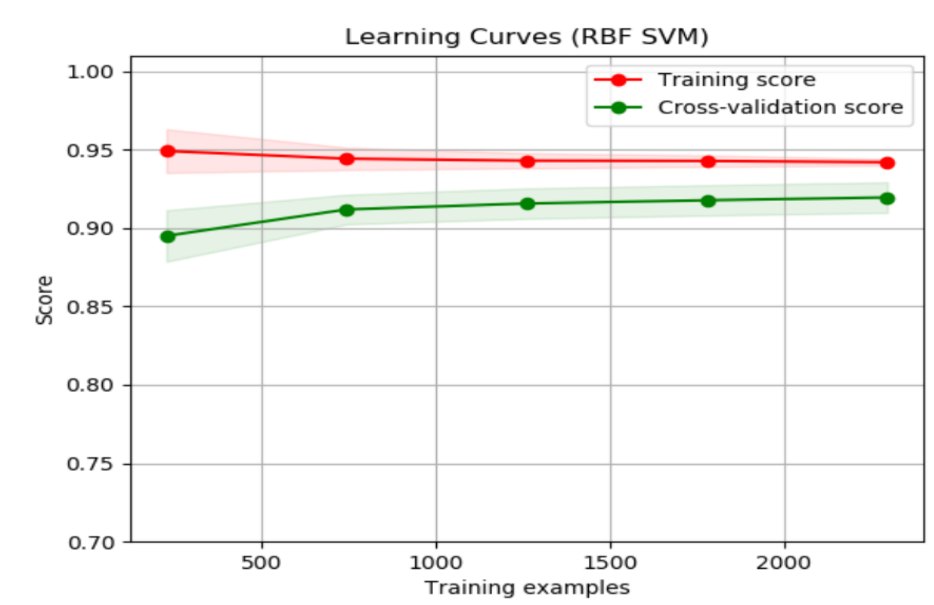


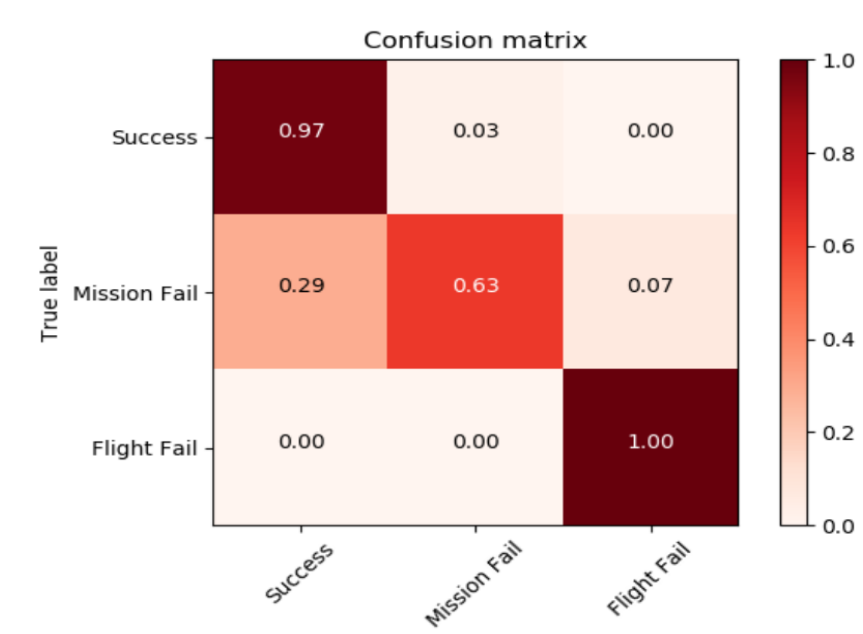
Results and Discussion

We achieved following results accuracy for the models that we used

Model	Train result(%)	Test Result (%)
LWR	-	71.69
Normal Eq.	81.89	89.55
Grad. Descent(L1)	79.42	87.39
Grad Descent (L2)	79.42	87.39
Decision Trees	91.68	90.12
Random Forests	86.09	86.45
SVM Linear Kernel	86.09	86.45
SVM RBF Kernel	93.04	92.18

Discussion: Based on the above results and the PCA plot, for our non-linear data, SVM-RBF Kernel & Decision Trees gave good results SVM-RBF achieved best result.





Review & Future Work

Result Analysis

- Model predicts flight failures with 100% accuracy, and 92.18% overall accuracy.
- Flights classified as mission failure/ flight failure but reported as success, might require maintenance.

If we had more time

- We could go with regression approach to predict how much time is left before the next failure. (RUL – Remaining useful time)
- We can also run unsupervised anomaly detection on the telemetry signals reported.

Deliverables

- **Serialized Model:** Our final model will be serialized and added to the codebase of Zipline.
- For a new flight, our model will be run to get the probabilities of success, mission failure and flight failure.
- Based on our predictions and thresholds set by Zipline, they can anticipate maintenance work on the parts used in the flight.

Acknowledgements

We thank Emma Schott and Matt Fay from Zipline for providing the data and meeting with us regularly to answer our questions about the data.

References & Tools

- Sklearn (https://scikit-learn.org)
- Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification"
- Widodo, Achmad, and Bo-Suk Yang. "Support vector machine in machine condition monitoring and fault diagnosis.
- Susto, Gian Antonio, et al. "Machine learning for predictive maintenance: A multiple classifier approach."