S

Machine Learning Engineer Nanodegree

Capstone Proposal

Mahmoud Helal

May 1st, 2019

Domain Background

Mobile application market considered one of the fastest growing markets in the last few years, according to (Gandhewar N and el.) [1] in 2010 Mobile devices approximately have usage 3.5 times more than PCs, that not only leads most of the companies with web-based systems like Facebook and Twitter, and companies that develop desktop applications such as Acrobat, Office, Photoshop to provide their services on mobile devices but also recently make companies and startups boost their business by only target smartphones platforms such as telegram, WhatsApp, Uber and TrueCaller or developing mobile games like PUBG, Clash of Clans and Candy Crush Saga..

Problem Statement

Most new applications and startups tend to fail; 90% of startups fail and the number one for failure is the lack of market need of their product [2] , so any startup or company intend to develop a new mobile app needs to make certain their App fulfill a market need; A model can predict application success by classifying the estimated number of downloads will occur based on previous similar applications published on the App store with similar features can help most of startups and mobile development companies to take business decisions to proceed or alter their plans from earlier stages, save money and reduce risks and increase customer satisfaction.

Datasets and Inputs

The dataset that will used is "Google Play Store Apps" [3] from Kaggle, the dataset structured in CSV format in one file named “*googleplaystore*” with 10.8k rows and 13 columns each row indicates real mobile application in play store and columns represent features as follows : Application name, Category the app belongs to, Overall user rating, Number of user reviews, Size of the app, Number of user downloads/installs, Type (Paid or Free), Price, Rating, Age group the app is targeted at (Children, Mature 21+, Adult), Genres, Last Updated, Current Version, Min required Android version.

This dataset will split into training. Validation and testing sets with number of downloads feature as label/target; Number of downloads label will be reduced into three or four main classes i.e. (+1K to 100K, +100K to 1M, …) and so one in order to facilitate kernel algorithm, also will try to split data based on time rather than random shuffle to make newer apps in testing dataset.

Solution Statement

Build a kernel that predict application estimated number of downloads can be reached considering this as a measurement for application success based on application category (Family, Game, …) and type (Free, paid) and other features based on given dataset, the kernel will be trained using supervised learning algorithms and classifiers such as (Logistic Regression, Naive Bayes, decision trees, SVM, …)

Benchmark Model

First will try to build basic kernel with simple classifier (Logistic Regression) the output performance will be used as datum for further classifiers and enhancements applied to kernel to compare how far our kernel progress.

Evaluation Metrics

The main evaluation method that will be used to measure kernel performance is f beta score with beta = 1; F1 score is considered a good evaluation metric for unbalanced datasets, the optimal value at 1 for perfect classifier and worst value at 0 for random classifier, the F1 score equation as follows:

$$F1\ score\ = 2 * \frac{precision\ *\ recall}{precision + recall}$$

Where $percision = \frac{true\ positive}{true\ positive + false\ positive}$

And $recall = \frac{true\ positive}{true\ positive + false\ negative}$

Project Design

1. **Pre-processing and Data Exploration:** in this stage we will try to explore and visualize data and remove noise (NANs, outliers, …) which should help us understand dataset and increase classification performance.

2. **Base kernel:** building Linear regression classifier as a benchmark model and get baseline performance values.

3. **Classifiers Comparison:** Explore more classifiers (Naive Bayes, decision trees, SVM, …) and compare results with base kernel result then choose best three classifiers to enhancement stage
4. **Enhancements:** may try different tuning and enhancements for example. try to use PCA for feature reduction or try cross-validation (K-Fold) to ensure kernel reliability.
5. **Conclusion:** summarizing project, justifying results and declare best classifier with highest performance.

References

[1] N. Gandhewar and R. Sheikh, "Google Android: An Emerging Software Platform For Mobile Devices," International Journal on Computer Science and Engineering (IJCSE), vol. 1, no. 1, pp. 12-17, 2010.

[2] E. Griffith, "Why startups fail, according to their founders," Fortune, 14 September 2014. [Online]. Available: http://fortune.com/2014/09/25/why-startups-fail-according-to-their-founders/.

[3] L. Gupta, "Google Play Store Apps," [Online]. Available: https://www.kaggle.com/lava18/google-play-store-apps.

[4] Statista, "Number of available applications in the Google Play Store from December 2009 to December 2018," [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[5] C. S. Brown, "Report: Google Play revenue grew 27% in 2018, besting iOS store growth," 17 January 2019. [Online]. Available: https://www.androidauthority.com/google-play-store-revenue-2018-944201/.

[6] Statcounter, "Mobile Operating System Market Share Worldwide - March 2019," [Online]. Available: http://gs.statcounter.com/os-market-share/mobile/worldwide.